FUJITSU

FUJITSU Software
Interstage Application Server

Reference Manual (API Edition)

Windows/Solaris/Linux

B1WS-1095-03ENZ0(00)
April 2014

Preface

Purpose of this Document

This manual explains the application interfaces provided by Interstage Application Server.

Note

Throughout this manual Interstage Application Server is referred to as Interstage.

Intended Readers

This manual is intended for users of Interstage Application Server and developers of distributed applications with Interstage Application Server.

It is assumed that readers of this manual have a basic knowledge of the following:

- C
- C++
- Java
- COBOL
- The Internet
- Object-oriented technology
- Distributed object technology (CORBA)
- Relational databases
- Basic knowledge of the OS used

Structure of This Document

The structure of this manual is as follows:

Chapter 1 C Interface

This chapter explains the C application interface.

Chapter 2 C++ Interface

This chapter explains the C++ application interface.

Chapter 3 Java Interface

This chapter explains the Java application interface.

Chapter 4 COBOL Interface

This chapter explains the COBOL application interface.

Conventions

Representation of Platform-specific Information

In the manuals of this product, there are parts containing content that relates to all products that run on the supported platform. In this case, an icon indicating the product platform has been added to these parts if the content varies according to the product. For this reason, refer only to the information that applies to your situation.

	Indicates that this product (32-bit) is running on Windows.
	Indicates that this product (64-bit) is running on Windows.
	Indicates that this product (32/64-bit) is running on Windows.

Solaris32	Indicates that this product (32-bit) is running on Solaris.
Solaris64	Indicates that this product (64-bit) is running on Solaris.
Solaris32/64	Indicates that this product (32/64-bit) is running on Solaris.
Linux32	Indicates that this product (32-bit) is running on Linux.
Linux64	Indicates that this product (64-bit) is running on Linux.
Linux32/64	Indicates that this product (32/64-bit) is running on Linux.

Abbreviations

Read occurrences of the following Components as their corresponding Service.

Service	Component
CORBA Service	ObjectDirector
Component Transaction Service	TransactionDirector

Export Controls

Exportation/release of this document may require necessary procedures in accordance with the regulations of the Foreign Exchange and Foreign Trade Control Law of Japan and/or US export control laws.

Trademarks

Trademarks of other companies are used in this documentation only to identify particular products or systems.

Product Trademarks/Registered Trademarks
Microsoft, Active Directory, ActiveX, Excel, Internet Explorer, MS-DOS, MSDN, Visual Basic, Visual C++, Visual Studio, Windows, Windows NT, Windows Server, Win32 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Other company and product names in this documentation are trademarks or registered trademarks of their respective owners.

Copyrights

April 2014 Third Edition

November 2012 First Edition

Contents

Chapter 1 C Interface

This chapter describes the C interface.

1.1 ORB Interface

This section explains the ORB (Object Request Broker) interface providing the basic interface for initializing the ORB. The ORB function mediates communications between clients and servers.

1.1.1 CORBA_ORB_init()

Name

CORBA_ORB_init

Synopsis

```
#include <orb.h>
CORBA_ORB  CORBA_ORB_init(
     int  *arg_c,
     char  **arg_v,
     CORBA_ORBid  orb_identifier,
     CORBA_Environment  *env);
```

Description

This function initializes the ORB, and returns a pseudo object reference of ORB. Issuing this function enables the use of the other functions described below. The object reference returned is used in parameters when other ORB interfaces are called.

The functions mentioned in the following sections can be used by invoking this function.

Parameters

arg_c

A pointer to a copy of the argc to be passed to the main function.

arg_v

A pointer to a copy of the argv to be passed to the main function.

orb_identifier

A name that identifies the ORB. Specify FJ_OM_ORBid.

env

A structure that may contain exception information.

Return Values

For normal termination, the ORB object reference is set.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in *_major* in the *env* structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

Notes

- This function should only be called once by a process.

- For shared server and unshared server, after setting the ORB control parameter, set the starting parameter of each of the following server applications in *arg_v*. The definition file is specified at the -ax option of OD_impl_inst and the starting parameter is defined at param of that definition file.

  1st parameter: "-ORB_FJ_PROC_ID"

  2nd parameter: Process number

  3rd parameter: "-ORB_FJ_HOME_DIR"

  4th parameter: Value of $OD_HOME

- You can use HTTP tunneling on a client application by passing an HTTP tunneling parameter as an argument to CORBA_ORB_init(). For an application that passes main arguments to CORBA_ORB_init() as they are, specify an HTTP tunneling parameter as the startup parameter. For details of HTTP tunneling, refer to 'How to start HTTP tunneling' in the Security System Guide.

- If CORBA_ORB_resolve_initial_references() is used to fetch the individual client initial service object reference, the values for argc and argv must be set as described in "Fetching Naming Service Initial Reference" in the Distributed Application Development Guide (CORBA Service Edition).

Example

- Example for HTTP tunneling.

```
char *argv[] = {
    "",    /* The first argument is ignored */
    "-ORB_FJ_HTTP", "yes",
    "-ORB_FJ_HTTPGW", "http://host.com/od-httpgw",
    NULL };
int argc = 5;

CORBA_Environment  env;
CORBA_ORB  orb;

orb = CORBA_ORB_init( &argc, argv, FJ_OM_ORBid, &env );
```

- Example for initial service.

```
char *argv[] = {
    "",    /* The first argument is ignored */
    "-ORB_FJ_HTTP", "yes",
    "-ORB_FJ_HTTPGW", "http://host.com/od-httpgw",
    NULL };
int argc = 5;

CORBA_Environment  env;
CORBA_ORB  orb;

orb = CORBA_ORB_init( &argc, argv, FJ_OM_ORBid, &env );
```

1.1.2 CORBA_ORB_BOA_init()

Name

CORBA_ORB_BOA_init

Synopsis

```
#include <orb.h>
CORBA_BOA  CORBA_ORB_BOA_init(
     CORBA_ORB  orb,
     int  *arg_c,
     char  **arg_v,
```

```
    CORBA_BOA_OAid  boa_identifier,
    CORBA_Environment  *env );
```

Description

This function initializes BOA (Basic Object Adapter) that controls server applications, and returns a pseudo object reference of BOA.

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

arg_c

A pointer to a copy of the argc to be passed to the main function.

arg_v

A pointer to a copy of the argv to be passed to the main function.

boa_identifier

Specify CORBA_BOA_OAid.

env

A structure that may contain exception information.

Return Values

For normal termination, the BOA object reference is returned, and CORBA_NO_EXCEPTION is set to _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set to _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.1.3 CORBA_ORB_resolve_initial_references()

Name

CORBA_ORB_resolve_initial_references

Synopsis

```
#include <orb.h>
CORBA_Object  CORBA_ORB_resolve_initial_references(
     CORBA_ORB  orb,
     CORBA_ORB_ObjectId  identifier,
     CORBA_Environment *env );
```

Description

This function returns the object reference of the object specified by identifier.

The object reference is fetched as described in "Fetching Naming Service Initial Reference" in the Distributed Application Development Guide (CORBA Service Edition).

Identifier can specify the following objects:

CORBA_ORB_ObjectId_LightInterfaceRepository

Interface Repository (Static skeleton interface)

CORBA_ORB_ObjectId_InterfaceRepository

Interface Repository (Dynamic skeleton interface)

CORBA_ORB_ObjectId_ImplementationRepository

Implementation Repository

CORBA_ORB_ObjectId_NameService

Name service

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

identifier

The initial reference identifier.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference is returned.

If the specified object does not exist, NULL is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

Remarks

By customizing the arguments to CORBA_ORB_init(), you can acquire the object reference to the client-specific initial service. For details, refer to 'Acquiring the initial reference to the naming service' in the Distributed Application Development Guide (CORBA Service Edition).

1.1.4 CORBA_ORB_resolve_initial_references_remote()

Name

CORBA_ORB_resolve_initial_references_remote

Synopsis

```
#include <orb.h>
CORBA_Object   CORBA_ORB_resolve_initial_references_remote(
     CORBA_ORB  orb,
     CORBA_ORB_ObjectId  identifier,
     CORBA_ORB_RemoteModifier *m,
     CORBA_Environment *env );
typedef CORBA_string  CORBA_Identifier
typedef CORBA_string InitAgentDesignator;
typedef sequence<InitAgentDesignator> RemoteModifier;
```

Description

This function returns the object reference of the object specified by the identifier. The object reference is retrieved by searching initial_services of the host defined in m in URL format

identifier can specify the following objects:

CORBA_ORB_ObjectId_InterfaceRepository: Interface Repository (Dynamic skeleton interface)

CORBA_ORB_ObjectId_NameService: Naming service

Multiple URLs can be specified at m. If so, they are searched in order of specification and the search is discontinued as soon as the object reference is found.

The URL specification format is shown below:

iiop://<address>[:<port>]

Either the host name, DNS name, or the IP address can be specified as <address>. The address cannot be omitted.

For <port>, specify the ORB port number at the connection destination.

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

identifier

The initial reference identifier.

m

A list of connection URLs.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference is returned.

If the specified object does not exist, NULL is returned.

For abnormal termination, exception information is set in the env structure. If CORBA_SYSTEM_EXCEPTION is issued, the following detailed information is set in _id of the env structure:

IDL:CORBA/ORB/InvalidName:1.0

Object specified by identifier not found

Example

```
CORBA_string  URLs[2];
CORBA_ORB_RemoteModifier m;

m._length = m._maximum = 2;
m._buffer = URLs;

URLs[1] = "iiop://remotehost01:8002";
URLs[0] = "iiop://remotehost02:8002";

cos_naming = CORBA_ORB_resolve_initial_references_remote(
                   orb,
                   CORBA_ORB_ObjectId_NameService,
                   &m,
                   &env);
```

1.1.5 CORBA_ORB_list_initial_services()

Name

CORBA_ORB_list_initial_services

Synopsis

```
#include <orb.h>
CORBA_ORB_ObjectIdList*  CORBA_ORB_list_initial_services(
     CORBA_ORB  orb,
     CORBA_Environment  *env );
```

Description

This function returns a list of the ObjectIds of all usable objects.

The ObjectIds are stored in _buffer of the CORBA_ORB_ObjectIdList structure.

```
typedef struct {
     CORBA_unsigned_long _maximum;
     CORBA_unsigned_long _length;
     CORBA_ORB_ObjectId *_buffer;
} CORBA_sequence_string;
typedef CORBA_sequence_string CORBA_ORB_ObjectIdList;
```

These objects are automatically generated by an OD installation service when OD is setup, and they immediately become usable.

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

env

A structure that may contain exception information.

Return Values

For normal termination, the ObjectId list is set.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.1.6 CORBA_ORB_object_to_string()

Name

CORBA_ORB_object_to_string

Synopsis

```
#include <orb.h>
CORBA_string  CORBA_ORB_object_to_string(
     CORBA_ORB  orb,
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function converts the object reference of the object specified by obj into a string.

Since this function acquires area to store converted character strings, use CORBA_free () to release the area as soon as it is no longer needed.

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

obj

The object reference to be converted to a string.

env

A structure that may contain exception information.

Return Values

For normal termination, string is set.

For abnormal termination, the NULL is returned, and CORBA_SYSTEM_EXCEPTION is set in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.1.7 CORBA_ORB_string_to_object()

Name

CORBA_ORB_string_to_object

Synopsis

```
#include <orb.h>
CORBA_Object  CORBA_ORB_string_to_object(
     CORBA_ORB  orb,
     CORBA_string  str,
     CORBA_Environment  *env );
```

Description

This function converts the string specified by str into an object reference. All character array formats are converted according to "URL Schema" in the Distributed Application Development Guide (CORBA Service Edition).

Since this function acquires area to store object references, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

str

The string to be converted to an object reference.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference is set.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major of the env structure, and detailed information is set in _id and _minor. The meaning of _id is explained below. Refer to CORBA Service Minor Codes in Exception Information Minor Codes to be Reported from the CORBA Service for information on _minor.

IDL:CORBA/BAD_PARAM:1.0

The specified URL schema has an error in its format.

Remarks

For str, you can specify not only a character string beginning with "IOR:" that can be acquired by CORBA_ORB_object_to_string(), but also a character string beginning with "cobaloc" or "corbaname". For details, refer to 'URL schemas' in the Distributed Application Development Guide (CORBA Service Edition).

Example

- Example for URL Schema.

```
int main(int argc, char **argv)
{
    CORBA_ORB  orb;
    CORBA_Object  obj_ns, obj_sv, obj_sv2;
    CORBA_Environment  env;
    int current_argc = argc;
    CosNaming_Name  name;
    CosNaming_NameComponent  name_component;

    orb = CORBA_ORB_init( &current_argc, argv, FJ_OM_ORBid, &env );

    name._length = name._maximum = 1;
    name._buffer = &name_component;
    name_component.id = "test1::intf1";
    name_component.kind = "";

    obj_ns = CORBA_ORB_string_to_object(
        orb,
        "corbaloc::nshost:8002/NameService",
        &env );

    obj_sv = CosNaming_NamingContext_resolve( obj_ns, &name, &env );

    obj_sv2 = CORBA_ORB_string_to_object(
        orb,
        "corbaname::nshost:8002/NameService#test1::intf1",
        &env );

    test_intf1_add( obj_sv, 1, 2 );
    test_intf1_add( obj_sv2, 1, 2 );

    return 0;
}
```

1.1.8 CORBA_ORB_create_list()

Name

CORBA_ORB_create_list

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_ORB_create_list(
     CORBA_ORB  orb,
     CORBA_long  count,
     CORBA_NVList  *new_list,
     CORBA_Environment  *env );
typedef  CORBA_Object  CORBA_NVList;
```

Description

This function generates a list (CORBA_NVList) of items whose number is specified by count, and sets the list in new_list. There are two methods of adding items to the generated list:

To add a parameter to the list object created here, use CORBA_NVList_add_item() on the NVList interface. To release the list object created here, use CORBA_NVList_free() on the NVList interface.

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

count

The number of items in the list object (CORBA_NVList) that will contain parameters.

new_list

A pointer to the area that will contain the list object (CORBA_NVList).

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.1.9 CORBA_ORB_create_operation_list()

Name

CORBA_ORB_create_operation_list

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_ORB_create_operation_list(
     CORBA_ORB  orb,
     CORBA_OperationDef  oper,
     CORBA_NVList  *new_list,
     CORBA_Environment  *env );
typedef  CORBA_Object  CORBA_OperationDef;
```

Description

This function returns a list (CORBA_NVList) of the arguments required for the operation specified in oper, in the parameter new_list. The list is returned in the same order as the operation definition.

Since ORB does not have the specified operation information, it is requested from the Interface Repository.

To release the list object created here, use CORBA_NVList_free() on the NVList interface.

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

oper

The OperationDef object to be used to create list objects.

new_list

A pointer to the area that will contain the list object.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.1.10 CORBA_ORB_get_default_context()

Name

CORBA_ORB_get_default_context

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_ORB_get_default_context(
     CORBA_ORB  orb,
     CORBA_Context  *ctx,
     CORBA_Environment  *env );
typedef  CORBA_Object  CORBA_Context;
```

Description

This function sets the object reference of a default Context object in ctx. Use CORBA_Context_delete() to release the Context object created here.

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

ctx

A pointer to the area that will contain the Context object.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.2 Object Interface

This section describes the object interface functions that manage and control the object reference.

1.2.1 CORBA_Object_is_nil()

Name

CORBA_Object_is_nil

Synopsis

```
#include <orb.h>
CORBA_boolean  CORBA_Object_is_nil(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function checks whether the specified object reference obj indicates a valid object.

Parameters

obj

The object reference to be inspected.

env

A structure that may contain exception information.

Return Values

When a valid object is indicated, CORBA_TRUE is returned.

When an invalid object is indicated, CORBA_FALSE is returned.

1.2.2 CORBA_Object_duplicate()

Name

CORBA_Object_duplicate

Synopsis

```
#include <orb.h>
CORBA_Object  CORBA_Object_duplicate(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function copies the object reference, and returns the object reference of the object copy.

Since this function acquires area to store object references, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

obj

The object reference to be inspected.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference is returned, and CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.2.3 CORBA_Object_release()

Name

CORBA_Object_release

Synopsis

```
#include <orb.h>
void  CORBA_Object_release(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function releases the object reference of the object obj. The object reference can be a copy generated by the CORBA_object_duplicate() function, or an original object reference given by OD (in a user process).

This function does not delete the original object references (in OD) managed by BOA. Use the BOA interface CORBA_object_dispose() function to delete an original object reference.

Parameters

obj

The object reference to be inspected.

env

A structure that may contain exception information.

Return Values

None

1.2.4 CORBA_Object_create_request()

Name

CORBA_Object_create_request

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_Object_create_request(
     CORBA_Object  obj,
     CORBA_Context  ctx,
     CORBA_Identifer  operation,
     CORBA_NVList  arg_list,
     CORBA_NamedValue  result,
     CORBA_Request  request,
     CORBA_Flags  req_flags,
     CORBA_Environment *env );
typedef  CORBA_Object  CORBA_Context;
typedef  CORBA_string  CORBA_Identifer;
typedef  CORBA_Object  CORBA_NVList;
typedef  struct  CORBA_NamedValue {
     CORBA_Identifer  name;
     CORBA_any  argument;
     CORBA_long  len;
     CORBA_Flags  arg_modes;
} CORBA_NamedValue;
typedef  CORBA_Object  CORBA_Request;
```

Description

This function creates an initialized Request object based on the specified arguments.

Parameters

obj

The object reference to the server application.

ctx

Specify the Context object returned by CORBA_Context_create_child() or CORBA_OBJECT_NIL.

operation

The name of the method that will call the server application.

arg_list

The NVList object returned by CORBA_ORB_create_list() or CORBA_ORB_create_operation_list().

result

Specify the NamedValue object returned by CORBA_NVlist_add_item() in the result.

For void returned values, specify TC_null. For the NamedValue object TypeCode (any type), specify the TypeCode from the return value of the method called by CORBA_Request_invoke(), CORBA_Request_send().

request

A object reference to request. The OD client must specify this object reference when calling the Request interface:

CORBA_Request_invoke() or CORBA_Request_send() functions.

req_flags

The following flag can be specified:

CORBA_OUT_LIST_MEMORY:

The list object (NVList) specified at arg_list is connected to the request object (arg_list must be specified). If the request is destroyed by CORBA_Request_delete(), the list object is also destroyed.

The argument list must be specified. If CORBA_OUT_LIST_MEMORY is not specified, each segment of output argument memory can be used until it is released by the programmer. Use the CORBA_free() function to release each memory segment.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.2.5 CORBA_Object_get_implementation()

Name

CORBA_Object_get_implementation

Synopsis

```
#include <orb.h>
     CORBA_ImplementationDef  CORBA_Object_get_implementation(
     CORBA_Object  obj,
     CORBA_Environment  *env );
typedef  CORBA_Object  CORBA_ImplementationDef;
```

Description

This function returns the object reference of the Implementation Repository for the object specified in obj.

If this function is executed on an application linked with Odwin.dll, it cannot find the implementation repository ID. This will result in a user exception with the _id value "IDL:FJ/NameDoesntExist:1.0".

Parameters

obj

Implementation information for the object reference specified here can be acquired.

env

A structure that may contain exception information.

Return Values

Returns the object reference of the Implementation Repository.

1.2.6 CORBA_Object_get_interface()

Name

CORBA_Object_get_interface

Synopsis

```
#include <orb.h>
CORBA_InterfaceDef  CORBA_Object_get_interface(
     CORBA_Object  obj,
     CORBA_Environment  *env );
typedef  CORBA_Object  CORBA_InterfaceDef;
```

Description

This function returns the object reference of the Interface Repository for the object specified in obj.

Parameters

obj

Implementation information for the object reference specified here can be acquired.

env

A structure that may contain exception information.

Return Values

Returns the object reference of the Interface Repository.

Remarks

To acquire interface information, the interface repository service is accessed within this API. Therefore, the interface information must be registered with the interface repository service.

1.3 BOA Interface

This section describes the BOA interface functions that control server applications.

1.3.1 CORBA_BOA_create()

Name

CORBA_BOA_create

Synopsis

```
#include <orb.h>
CORBA_Object CORBA_BOA_create(
     CORBA_BOA  boa,
     CORBA_ReferenceData  *ref,
     CORBA_InterfaceDef  intf,
     CORBA_ImplementationDef  impl,
     CORBA_Environment  *env );
typedef  CORBA_sequence_octet  CORBA_ReferenceData;
typedef  struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     CORBA_octet  *_buffer;
} CORBA_sequence_octet;
typedef  CORBA_Object  CORBA_InterfaceDef;
typedef  CORBA_Object  CORBA_ImplementationDef;
```

Description

This function generates and returns an object reference:

Code type information is appended to the object reference to be generated, when the code information is set by the OD_impl_inst or OD_set_env commands.

When creating an object reference in the server application, the code type is set to the same value as it would be if the -L option was omitted from the OD_or_adm command.

To dispose of the object reference created here, use CORBA_BOA_dispose(). Alternatively, to release the object reference memory, use CORBA_Object_release().

Parameters

boa

The BOA object reference returned by CORBA_ORB_BOA_init().

ref

Object identification information to be set by the OD server when the object is generated. Identification information is set in _buffer in the CORBA_sequence_octet structure. This value remains unchanged until the object is destroyed.

intf

The object reference of the Interface Repository stored by the interface and installed by the object.

impl

The object reference of the Implementation Repository stored by the object installation information.

ref

A structure that may contain exception information.

Return Values

For normal termination, the object reference is returned, and CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.3.2 CORBA_BOA_dispose()

Name

CORBA_BOA_dispose

Synopsis

```
#include <orb.h>
void  CORBA_BOA_dispose(
     CORBA_BOA  boa,
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function destroys the object reference specified in obj.

Parameters

boa

The object reference returned by CORBA_ORB_BOA_init().

obj

The object reference to be disposed.

env

A structure that may contain exception information.

Return Values

None.

Remarks

To release the object reference, use CORBA_Object_release().

1.3.3 CORBA_BOA_get_id()

Name

CORBA_BOA_get_id

Synopsis

```
#include <orb.h>
CORBA_ReferenceData  *CORBA_BOA_get_id(
     CORBA_BOA  boa,
     CORBA_Object  obj,
     CORBA_Environment *env );
typedef CORBA_sequence_octet  CORBA_ReferenceData;
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     CORBA_octet  *_buffer;
} CORBA_sequence_octet;
```

Description

This function returns identification information for the object reference specified in obj. This information is the value specified by the CORBA_BOA_create() function, and does not change until the object is destroyed.

Parameters

boa

The object reference returned by CORBA_ORB_BOA_init().

obj

The identification information for the object reference specified here will be returned.

env

A structure that may contain exception information.

Return Values

For normal termination, identification information (octet type) is returned, and CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.3.4 CORBA_BOA_set_exception()

Name

CORBA_BOA_set_exception

Synopsis

```
#include <orb.h>
void  CORBA_BOA_set_exception(
     CORBA_BOA  boa,
     CORBA_unsigned_long  major,
     CORBA_string  user_id,
     void  *param,
     CORBA_Environment  *env );
```

Description

This function sets exception information. The server application can be terminated by calling this function before control is returned.

At this time, the exception information set here will be returned to the client application.

Parameters

boa

The object reference returned by CORBA_ORB_BOA_init().

major

The following values can be specified for _major:

CORBA_SYSTEM_EXCEPTION /* Standard exception */

CORBA_USER_EXCEPTION /* User exception */

user_id

An exception identifier.

env

The env structure passed to the operation function.

Return Values

None.

Note

Be careful when using the last argument env. While env for other APIs will contain exception information if the API abnormally terminates, env for this API must be set to a pointer to the CORBA_Environment structure passed to the server operation function.

1.3.5 CORBA_BOA_impl_is_ready()

Name

CORBA_BOA_impl_is_ready

Synopsis

```
#include <orb.h>
void  CORBA_BOA_impl_is_ready(
     CORBA_BOA  boa,
     CORBA_ImplementationDef  impl,
     CORBA_Environment  *env );
typedef  CORBA_Object  CORBA_ImplementationDef;
```

Description

This function notifies ORB that the shared or persistent server specified in impl is ready to receive requests.

For thread applications, when the mode is set to SYNC_END while registering the object in the implementation repository, the function does not return until the CORBA_BOA_deactivate_impl function is invoked.

Parameters

boa

The object reference returned by CORBA_ORB_BOA_init().

impl

Implementation information.

env

A structure that may contain exception information.

Return Values

None.

1.3.6 CORBA_BOA_deactivate_impl()

Name

CORBA_BOA_deactivate_impl

Synopsis

```
#include <orb.h>
void  CORBA_BOA_deactivate_impl(
     CORBA_BOA  boa,
     CORBA_ImplementationDef  impl,
     CORBA_Environment  *env );
```

Description

This function notifies ORB that the process for the shared or persistent server specified in impl is stopped.

Parameters

boa

The object reference returned by CORBA_ORB_BOA_init().

impl

Implementation information.

env

A structure that may contain exception information.

Return Values

None.

Notes

Do not invoke this function to stop CORBA WorkUnit processes. To stop CORBA WorkUnit processes, either execute the *isstopwu* command, or stop the CORBA WorkUnit process from the Interstage Management Console.

1.3.7 CORBA_BOA_obj_is_ready()

Name

CORBA_BOA_obj_is_ready

Synopsis

```
#include <orb.h>
void  CORBA_BOA_obj_is_ready(
     CORBA_BOA  boa,
     CORBA_Object  obj,
     CORBA_ImplementationDef  impl,
     CORBA_Environment  *env );
typedef  CORBA_Object  CORBA_ImplementationDef;
```

Description

This function notifies ORB that the object specified in obj on the unshared server (specified in impl) is ready to receive requests.

Parameters

boa

The object reference returned by CORBA_ORB_BOA_init().

obj

The object that is now ready for request receipt.

impl

Implementation information.

env

A structure that may contain exception information.

Return Values

None.

1.3.8 CORBA_BOA_deactivate_obj()

Name

CORBA_BOA_deactivate_obj

Synopsis

```
#include <orb.h>
void  CORBA_BOA_deactivate_obj(
     CORBA_BOA  boa,
     CORBA_Object  obj,
     CORBA_Environment *env );
typedef  CORBA_Object  CORBA_ImplementationDef;
```

Description

This function notifies the ORB that the object (in the shared or unshared server) specified in obj is deactivated.

Parameters

boa

The object reference returned by CORBA_ORB_BOA_init().

obj

The object to be stopped.

env

A structure that may contain exception information.

Return Values

None.

1.3.9 CORBA_BOA_set_impl_dsi()

Name

CORBA_BOA_set_impl_dsi

Synopsis

```
#include <orb.h>
void  CORBA_BOA_set_impl_dsi(
     CORBA_BOA  boa,
     CORBA_Environment  *env,
     CORBA_DynamicImplementationRoutine  dsi);
```

Description

This function adds the DSI processing function specified in dsi.

ORB calls the DSI function specified in dsi, if the server application method is called.

Parameters

boa

The object reference returned by CORBA_ORB_BOA_init().

env

A structure that may contain exception information.

dsi

A pointer to the DSI processing function.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.4 NVList Interface

This section describes the NVList interface functions.

1.4.1 CORBA_NVList_add_item()

Name

CORBA_NVList_add_item

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_NVList_add_item(
     CORBA_Object  list,
     CORBA_Identifier  item_name,
     CORBA_TypeCode  item_type,
     void  *value,
     CORBA_long  value_len,
     CORBA_Flags  item_flags,
     CORBA_Environment  *env );
```

Description

This function adds parameter information to the list object specified in list.

The parameter is added to the end of a previously added parameter.

Parameters

list

The list object that will contain the server application parameter information.

item_name

The parameter name.

item_type

The TypeCode object for the parameter.

value

A pointer to the parameter memory area.

value_len

The length of the parameter memory area.

item_flags

One of the values shown below can be specified in item_flags:

CORBA_ARG_IN	Related values are input arguments only

CORBA_ARG_OUT	Related values are output arguments only
CORBA_ARG_INOUT	Related values are input-output arguments
CORBA_IN_COPY_VALUE	The value is copied, and the copy is used
DEPENDENT_LIST	When a master list is released, sub lists are also released

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.4.2 CORBA_NVList_free()

Name

CORBA_NVList_free

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_NVList_free(
     CORBA_Object  list,
     CORBA_Environment  *env );
```

Description

This function releases the list structure of the list object specified in list, and the memory associated with it.

Parameters

list

The list object.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.4.3 CORBA_NVList_free_memory()

Name

CORBA_NVList_free_memory

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_NVList_free_memory(
     CORBA_Object  list,
     CORBA_Environment  *env );
```

Description

This function releases the memory associated with the list structure of the list object specified in list. It does not release the list structure itself.

Parameters

list

The list object.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.4.4 CORBA_NVList_get_count()

Name

CORBA_NVList_get_count

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_NVList_get_count(
     CORBA_Object  list,
     CORBA_long  *count,
     CORBA_Environment  *env );
```

Description

This function sets the total number of items assigned to the list object (specified in list) in count, then returns the value of count.

Parameters

list

The list object.

count

The total number of parameters is assigned.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned, and the total number of parameters is set in count.

For abnormal termination, CORBA_FAILED is returned.

1.5 Context Interface

This section describes the Context interface functions that control Context objects.

1.5.1 CORBA_Context_create_child()

Name

CORBA_Context_create_child

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_Context_create_child(
     CORBA_Object  ctx,
     CORBA_Identifier  ctx_name,
     CORBA_Context  *child_ctx,
     CORBA_Environment  *env );
typedef  CORBA_string  CORBA_Identifier;
typedef  CORBA_Object  CORBA_Context;
```

Description

This function generates the context objects related to objects specified at ctx. Use CORBA_Context_delete() to release the Context object created here.

Parameters

ctx

The object reference to be associated.

ctx_name

The name of the Context object to be created.

child_ctx

A pointer to the area that will contain the Context object.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned, and child_ctx is set in the Context object reference.

For abnormal termination, CORBA_FAILED is returned.

1.5.2 CORBA_Context_set_one_value()

Name

CORBA_Context_set_one_value

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_Context_set_one_value(
     CORBA_Object  ctx,
     CORBA_Identifier  prop_name,
     CORBA_string  value,
     CORBA_Environment  *env );
typedef  CORBA_string  CORBA_Identifier;
```

Description

This function enables the client or server to set one or more attribute values when either specifies an object reference obtained by CORBA_ORB_get_default_context() or CORBA_Context_create_child() in ctx.

Parameters

ctx

The object reference to the Context object.

prop_name

The name of the attribute value to be set.

value

The value of the attribute to be set.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.5.3 CORBA_Context_set_values()

Name

CORBA_Context_set_values

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_Context_set_values(
     CORBA_Object  ctx,
     CORBA_NVList  value,
     CORBA_Environment  *env );
typedef  CORBA_Object  CORBA_NVList;
```

Description

This function enables the client or server to set one or more attribute values when either specifies an object reference obtained by CORBA_ORB_get_default_context() or CORBA_Context_create_child().

A Context object only supports strings, therefore, when the CORBA_Context_set_values() function is used, each field in the specified NVList object must be set as follows:

Set CORBA_ARG_IN in the flag

Specify TC_string in the item_type

Parameters

ctx

The object reference to the Context object.

value

The NVList object that contains specifications of the attribute values to be set.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.5.4 CORBA_Context_get_values()

Name

CORBA_Context_get_values

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_Context_get_values(
     CORBA_Object  ctx,
     CORBA_Identifier  start_scope,
     CORBA_Flags  op_flags,
     CORBA_Identifier  prop_name,
     CORBA_NVList  *value,
     CORBA_Environment  *env );
typedef  CORBA_Object  CORBA_NVList;
typedef  CORBA_string  CORBA_Identifier;
```

Description

This function acquires the Context attribute of the object reference ctx to the Context object obtained by CORBA_ORB_get_default_context() or CORBA_Context_create_child(), and then stores the acquired value in the value area.

Parameters

ctx

The object reference to the Context object.

start_scope

This value indicates the hierarchical level of the Context object at which to start searching for and retrieving the specified attribute. When no scope name is specified (NULL), retrieval starts with the specified Context object. If no attribute is found within the specified scope, and the CTX_RESTRICT_FLAG is not specified, retrieval continues up the Context tree. If the specified scope name is found, an exception code is returned.

op_flags

The CTX_RESTRICT_SCOPE flag can be specified so that retrieval is restricted by the specified retrieval scope, or the Context object.

prop_name

Specify the Context attribute value to be acquired. Specifying a wildcard character (*) returns all matching attribute names and their values.

value

A pointer to the NVList object that will contain attribute values.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned, and attribute is set in value.

For abnormal termination, CORBA_FAILED is returned.

1.5.5 CORBA_Context_delete_values()

Name

CORBA_Context_delete_values

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_Context_delete_value(
     CORBA_Object  ctx,
     CORBA_Identifier  prop_name,
     CORBA_Environment  *env );
```

Description

This function deletes the value of the attribute specified in prop_name from the Context object specified in ctx.

Parameters

ctx

The object reference to the Context object.

prop_name

Specify the attribute value to be deleted. If a wild card character (*) is specified, all matching attribute names and their values will be deleted.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.5.6 CORBA_Context_delete()

Name

CORBA_Context_delete

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_Context_delete(
     CORBA_Object  ctx,
     CORBA_Flags  del_flags,
     CORBA_Environment  *env );
```

Description

This function deletes the Context object specified in ctx. OD invalidates the corresponding object reference, in a similar way to the CORBA_BOA_dispose() function.

Parameters

ctx

The object reference to the Context object.

del_falgs

If there are any Child Context objects, set the CTX_DELETE_DESCENDENTS flag in del_flags, otherwise this function terminates abnormally. This ensures that descendent Context objects are also deleted.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.6 Request Interface

This section describes the request interface functions.

1.6.1 CORBA_Request_add_arg()

Name

CORBA_Request_add_arg

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_Request_add_arg(
CORBA_Request  req,
        CORBA_Identifier  name,
        CORBA_TypeCode  arg_type,
        void  *value,
CORBA_long  len,
        CORBA_Flags  arg_flags,
        CORBA_Environment  *env );
typedef  CORBA_string  CORBA_Identifer;
```

Description

This function adds the parameter information to the request specified in req.

Value and len must be specified in the parameter, whereas, arg_type, name and arg_flags are optional.

To bind the parameter to the request object while calling CORBA_Object_create_request(), there are two methods of specification. These two methods cannot be used together.

Parameters

req

The reference of the request object returned from the CORBA_Object_create_request() function is specified.

name

The parameter name.

arg_type

The TypeCode object for the parameter.

value

A pointer to the parameter memory area.

len

The length of the parameter memory area.

arg_flags

The following flags can be specified in arg_flags:

CORBA_ARG_IN	Input parameter only
CORBA_ARG_OUT	Output parameter only

CORBA_ARG_INOUT	Input output parameter
CORBA_IN_COPY_VALUE	The argument value is copied, and the copy is used instead of the original. This flag is ignored when CORBA_ARG_INOUT and CORBA_ARG_OUT are specified.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.6.2 CORBA_Request_invoke()

Name

CORBA_Request_invoke

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_Request_invoke(
     CORBA_Request  req,
     CORBA_Flags  invoke_flags,
     CORBA_Environment  *env );
```

Description

This function sends the request specified in req.

OD finds and calls the appropriate operation. The operation result is set in the result argument specified in the CORBA_Object_create_request function.

The caller waits for the end of the operation.

However, when using the method in which user exception is defined, first register the contents of the IDL file in an Interface Repository. For details refer to the *IDLc* command in the Reference Manual (Command Edition).

Parameters

req

The reference of the request object returned from the CORBA_Object_create_request() function is specified.

invoke_flags

0 is specified.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.6.3 CORBA_Request_send()

Name

CORBA_Request_send

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_Request_send(
     CORBA_Request  req,
     CORBA_Flags  invoke_flags,
     CORBA_Environment  *env );
```

Description

This function sends the request specified in req.

Unlike CORBA_Request_invoke(), this function returns control to the caller without waiting for the end of the operation. If the operation is defined as one way, or INV_NO_RESPONSE is specified in invoke_flags, the CORBA_Request_get_response() function need not be invoked.

The operation result is reported by CORBA_Request_get_response().

It is not necessary to call CORBA_Request_get_response() if CORBA_INV_NO_RESPONSE is specified in invoke_flags.

Parameters

req

The reference of the request object returned from the CORBA_Object_create_request() function is specified.

invoke_flags

The following flag can be specified in invoke_flags:

CORBA_INV_NO_RESPONSE

The calling program neither waits for a response, nor expects the output parameter (inout or out) to be updated.

This option can be specified even when the operation is not defined as one way.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.6.4 CORBA_Request_delete()

Name

CORBA_Request_delete

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_Request_delete(
     CORBA_Request  req,
     CORBA_Environment  *env );
```

Description

This function deletes the request object specified in req. It releases all NVList objects related to the request object.

Parameters

req

The reference of the request object returned from the CORBA_Object_create_request() function is specified.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.6.5 CORBA_Request_get_response()

Name

CORBA_Request_get_response

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_Request_get_response(
     CORBA_Request  req,
     CORBA_Flags  response_flags,
     CORBA_Environment  *env );
```

Description

This function determines whether the request specified in req has been terminated. It can be used to determine whether a request is terminated by calling this function after the CORBA_Request_send() function.

When the request has been terminated, the output parameters defined in the request and return values are valid.

The parameter and value can be treated as if the CORBA_Request_invoke() function was executed.

The CORBA_Request_get_response() function returns control even when the specified request is in progress at server application function level.

Parameters

req

The reference of the request object returned from the CORBA_Object_create_request() function is specified.

invoke_flags

The following flag can be specified in invoke_flags:

CORBA_INV_NO_WAIT

env

A structure that may contain exception information.

Return Values

For normal request termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.7 ServerRequest Interface

This section explains the ServerRequest interface used by the dynamic skeleton interface.

1.7.1 CORBA_ServerRequest_op_name()

Name

CORBA_ServerRequest_op_name

Synopsis

```
#include <orb.h>
CORBA_Identifier  CORBA_ServerRequest_op_name(
     CORBA_Object       request,
     CORBA_Environment  *env);
```

Description

This function fetches the function name from the ServerRequest object specified in request.

Parameters

request

The reference of the request object returned from the CORBA_Object_create_request() function is specified.

env

A structure that may contain exception information.

Return Values

For normal termination, the function name is returned, and CORBA_NO_EXCEPT is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.7.2 CORBA_ServerRequest_params()

Name

CORBA_ServerRequest_params

Synopsis

```
#include <orb.h>
void  CORBA_ServerRequest_params(
     CORBA_Object       request,
     CORBA_NVList       arg_list,
     CORBA_Environment  *env);
```

Description

This function fetches parameter information from the ServerRequest object specified in request, and sets it in the NVList object specified in arg_list.

Parameters

request

The target ServerRequest object. Specify the object reference posted as the second parameter to the gateway function.

arg_list

Specify the object reference created by CORBA_ORB_create_list(). You must allocate as many parameter information areas as the number of parameters using CORBA_NVList_add_item().

env

A structure that may contain exception information.

Return Values

For normal termination, parameter information is set in the NVList object specified in arg_list, and CORBA_NO_EXCEPT is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.7.3 CORBA_ServerRequest_result()

Name

CORBA_ServerRequest_result

Synopsis

```
#include <orb.h>
void  CORBA_ServerRequest_result(
     CORBA_Object       request,
     CORBA_any          any_value,
     CORBA_Environment  *env);
```

Description

This function sets the return value of the server application method for the ServerRequest object specified in request.

Parameters

request

The target ServerRequest object.

any_value

Set the any-type variable assigned the return value.

env

A structure that may contain exception information.

Return Values

For normal termination, the function name is returned, and CORBA_NO_EXCEPT is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.7.4 CORBA_ServerRequest_exception()

Name

CORBA_ServerRequest_exception

Synopsis

```
#include <orb.h>
void CORBA_ServerRequest_exception(
     CORBA_Object            request,
     CORBA_exception_type    flag,
```

```
    CORBA_char              *id,
    CORBA_any               *any_value,
    CORBA_Environment       *env);
```

Description

This function sets server application user exceptions for the ServerRequest object specified by "request".

Parameters

request

The target ServerRequest object.

flag

Sets CORBA_USER_EXCEPTION.

id

Sets an identifier for the exception.

any_value

Specify the any-type variable assigned the user exception.

env

A structure in which exception information is stored when this function ends abnormally.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in env structure _major.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in env structure _major, and detailed information is set in env structures _id and _minor.

For the meanings of _id and _minor, refer to Exception Information and Minor Codes Posted from the CORBA Service in the Messages manual.

1.7.5 CORBA_ServerRequest_ctx()

Name

CORBA_ServerRequest_ctx

Synopsis

```
#include <orb.h>
CORBA_Context CORBA_ServerRequest_ctx(
    CORBA_Object       request,
    CORBA_Environment  *env);
```

Description

This function fetches the context object for the ServerRequest object specified in request.

Parameters

request

The target ServerRequest object.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference for the Context object is returned, and CORBA_NO_EXCEPT is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.8 TypeCode Interface

This section explains the TypeCode interface that controls TypeCode.

1.8.1 CORBA_TypeCode_equal()

Name

CORBA_TypeCode_equal

Synopsis

```
#include <orb.h>
CORBA_boolean  CORBA_TypeCode_equal(
     CORBA_TypeCode  Object,
     CORBA_TypeCode  tc,
     CORBA_Environment  *env);
```

Description

This function compares the TypeCode object specified in Object with the TypeCode object specified in tc.

Parameters

Object

The target TypeCode object.

tc

The TypeCode object that Object is to be compared with.

env

A structure that may contain exception information.

Return Values

If the TypeCodes match, CORBA_TRUE is returned, otherwise, CORBA_FALSE is returned.

1.8.2 CORBA_TypeCode_kind()

Name

CORBA_TypeCode_kind

Synopsis

```
#include <orb.h>
CORBA_TCKind CORBA_TypeCode_kind(
     CORBA_TypeCode  Object,
     CORBA_Environment  *env);
```

Description

This function returns attribute information about the TypeCode object specified in Object. The attribute information is as follows:

CORBA_tk_null	0L
CORBA_tk_void	1L
CORBA_tk_short	2L
CORBA_tk_long	3L
CORBA_tk_ushort	4L
CORBA_tk_ulong	5L
CORBA_tk_float	6L
CORBA_tk_double	7L
CORBA_tk_boolean	8L
CORBA_tk_char	9L
CORBA_tk_octet	10L
CORBA_tk_any	11L
CORBA_tk_TypeCode	12L
CORBA_tk_Principal	13L
CORBA_tk_objref	14L
CORBA_tk_struct	15L
CORBA_tk_union	16L
CORBA_tk_enum	17L
CORBA_tk_string	18L
CORBA_tk_sequence	19L
CORBA_tk_array	20L
CORBA_tk_alias	21L
CORBA_tk_except	22L
CORBA_tk_longlong	23L
CORBA_tk_ulonglong	24L
CORBA_tk_longdouble	25L
CORBA_tk_wchar	26L
CORBA_tk_wstring	27L

Parameters

Object

The target TypeCode object.

env

A structure that may contain exception information.

Return Values

For normal termination, attribute information is returned.

For abnormal termination, CORBA_tk_null is returned.

1.8.3 CORBA_TypeCode_id()

Name

CORBA_TypeCode_id

Synopsis

```
#include <orb.h>
CORBA_RepositoryId  CORBA_TypeCode_id(
     CORBA_TypeCode  Object,
     CORBA_Environment  *env);
```

Description

This function returns the Repository ID of the TypeCode object specified in Object. The Repository ID becomes valid for the TypeCode with the following attribute information:

CORBA_tk_objref	14L
CORBA_tk_struct	15L
CORBA_tk_union	16L
CORBA_tk_enum	17L
CORBA_tk_alias	21L
CORBA_tk_except	22L

Since this function acquires area to store report IDs, use CORBA_free () to release the area as soon as it is no longer needed.

Parameters

Object

The target TypeCode object.

env

A structure that may contain exception information.

Return Values

For normal termination, the Repository ID is returned.

For abnormal termination, CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id.

The value of _id and its meaning is as follows:

ex_CORBA_TypeCode_BadKind /* TypeCode attribute information is invalid. */

1.8.4 CORBA_TypeCode_name()

Name

CORBA_TypeCode_name

Synopsis

```
#include <orb.h>
CORBA_Identifier  CORBA_TypeCode_name(
     CORBA_TypeCode  Object,
     CORBA_Environment  *env);
```

Description

This function returns the name of the TypeCode object specified in Object. The name becomes valid for the TypeCode with the following attribute information:

CORBA_tk_objref	14L
CORBA_tk_struct	15L
CORBA_tk_union	16L
CORBA_tk_enum	17L
CORBA_tk_alias	21L
CORBA_tk_except	22L

Since this function acquires area to store TypeCode object names, use CORBA_free () to release the area as soon as it is no longer needed.

Parameters

Object

The target TypeCode object.

env

A structure that may contain exception information.

Return Values

For normal termination, the member name (CORBA_Identifier:String type) is returned.

For abnormal termination, CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id.

The value of _id and its meaning is as follows:

ex_CORBA_TypeCode_BadKind /* TypeCode attribute information is invalid. */

1.8.5 CORBA_TypeCode_member_count()

Name

CORBA_TypeCode_member_count

Synopsis

```
#include <orb.h>
CORBA_unsigned_long  CORBA_TypeCode_member_count(
     CORBA_TypeCode  Object,
     CORBA_Environment *env);
```

Description

This function returns the member count of the TypeCode object specified in Object. The member count becomes valid for the TypeCode with the following attribute information:

CORBA_tk_struct	15L
CORBA_tk_union	16L
CORBA_tk_enum	17L
CORBA_tk_except	22L

Parameters

Object

The target TypeCode object.

env

A structure that may contain exception information.

Return Values

For normal termination, the member count is returned.

For abnormal termination, CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id.

The value of _id and its meaning is as follows:

ex_CORBA_TypeCode_BadKind /* TypeCode attribute information is invalid. */

1.8.6 CORBA_TypeCode_member_name()

Name

CORBA_TypeCode_member_name

Synopsis

```
#include <orb.h>
CORBA_Identifier  CORBA_TypeCode_member_name(
     CORBA_TypeCode  Object,
     unsigned long  index,
     CORBA_Environment  *env);
```

Description

This function returns the name of the member (specified in index) of the TypeCode object specified in Object.

The name becomes valid for the TypeCode with the following attribute information:

CORBA_tk_struct	15L
CORBA_tk_union	16L
CORBA_tk_enum	17L
CORBA_tk_except	22L

Since this function acquires area to store member names, use CORBA_free () to release the area as soon as it is no longer needed.

Parameters

Object

The target TypeCode object.

index

The index of the member.

env

A structure that may contain exception information.

Return Values

For normal termination, the name of the member (CORBA_Identifier:String type) is returned.

For abnormal termination, CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id.

The values of _id and their meaning are as follows:

ex_CORBA_TypeCode_BadKind /* TypeCode attribute information is invalid. */

ex_CORBA_TypeCode_Bounds /* The index value is invalid. */

1.8.7 CORBA_TypeCode_member_type()

Name

CORBA_TypeCode_member_type

Synopsis

```
#include <orb.h>
CORBA_TypeCode  CORBA_TypeCode_member_type(
     CORBA_TypeCode  Object,
     unsigned long  index,
     CORBA_Environment  *env);
```

Description

This function returns the TypeCode of the member (specified in index) of the TypeCode object specified in Object. The TypeCode becomes valid for the TypeCode with the following attribute information:

CORBA_tk_struct	15L
CORBA_tk_union	16L
CORBA_tk_except	22L

Since this function acquires area to store TypeCodes, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

Object

The target TypeCode object.

index

The index of the member.

env

A structure that may contain exception information.

Return Values

For normal termination, the TypeCode of the member is returned.

For abnormal termination, CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id.

The values of _id and their meaning are as follows:

ex_CORBA_TypeCode_BadKind /* TypeCode attribute information is invalid. */

ex_CORBA_TypeCode_Bounds /* The index value is invalid. */

1.8.8 CORBA_TypeCode_member_label()

Name

CORBA_TypeCode_member_label

Synopsis

```
#include <orb.h>
CORBA_any  CORBA_TypeCode_member_label(
     CORBA_TypeCode  Object,
     unsigned long  index,
     CORBA_Environment  *env);
```

Description

This function returns the label of the member (specified in index) of the TypeCode object specified in Object. The label becomes valid for the TypeCode with the following attribute information:

CORBA_tk_union	16L

Since this function acquires area to store member labels, use CORBA_free () to release the area as soon as it is no longer needed.

Parameters

Object

The target TypeCode object.

index

The index of the member.

env

A structure that may contain exception information.

Return Values

For normal termination, the member label is returned.

For abnormal termination, CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id.

The values of _id and their meaning are as follows:

ex_CORBA_TypeCode_BadKind /* TypeCode attribute information is invalid. */

ex_CORBA_TypeCode_Bounds /* The index value is invalid. */

1.8.9 CORBA_TypeCode_discriminator_type()

Name

CORBA_TypeCode_discriminator_type

Synopsis

```
#include <orb.h>
CORBA_TypeCode  CORBA_TypeCode_discriminator_type(
     CORBA_TypeCode  Object,
     CORBA_Environment *env);
```

Description

This function returns the discrimination information definition TypeCode of the TypeCode object specified in Object. The discrimination information definition TypeCode becomes valid for the TypeCode with the following attribute information:

CORBA_tk_union	16L

Since this function acquires area to store discrimination information definition TypeCodes, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

Object

The target TypeCode object.

env

A structure that may contain exception information.

Return Values

For normal termination, the discrimination information definition TypeCode is returned.

For abnormal termination, CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id.

The value of _id and its meaning is as follows:

ex_CORBA_TypeCode_BadKind /* TypeCode attribute information is invalid. */

1.8.10 CORBA_TypeCode_default_index()

Name

CORBA_TypeCode_default_index

Synopsis

```
#include <orb.h>
CORBA_long  CORBA_TypeCode_default_index(
     CORBA_TypeCode  Object,
     CORBA_Environment  *env);
```

Description

This function returns the index of the default member of the TypeCode object specified in Object. The index becomes valid for the TypeCode with the following attribute information:

CORBA_tk_union	16L

Parameters

Object

The target TypeCode object.

env

A structure that may contain exception information.

Return Values

For normal termination, the index of the default member is returned. If the default member is not available, -1 is returned.

For abnormal termination, CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id.

The value of _id and its meaning is as follows:

ex_CORBA_TypeCode_BadKind /* TypeCode attribute information is invalid. */

1.8.11 CORBA_TypeCode_length()

Name

CORBA_TypeCode_length

Synopsis

```
#include <orb.h>
CORBA_unsigned_long  CORBA_TypeCode_length(
     CORBA_TypeCode  Object,
     CORBA_Environment  *env);
```

Description

This function returns the length of the TypeCode object specified in Object (the string length for CORBA_tk_string, and the number of elements for CORBA_tk_sequence and CORBA_tk_array). The length becomes valid for the TypeCode with the following attribute information:

CORBA_tk_string	18L
CORBA_tk_sequence	19L
CORBA_tk_array	20L
CORBA_tk_wstring	27L

Parameters

Object

The target TypeCode object.

env

A structure that may contain exception information.

Return Values

For normal termination, the TypeCode object length is returned.

For abnormal termination, CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id.

The value of _id and its meaning is as follows:

ex_CORBA_TypeCode_BadKind /* TypeCode attribute information is invalid. */

1.8.12 CORBA_TypeCode_content_type()

Name

CORBA_TypeCode_content_type

Synopsis

```
#include <orb.h>
CORBA_TypeCode  CORBA_TypeCode_content_type(
     CORBA_TypeCode  Object,
     CORBA_Environment  *env);
```

Description

This function returns the member TypeCode object of the TypeCode object specified in Object. The member TypeCode object becomes valid for the TypeCode with the following attribute information:

CORBA_tk_sequence	19L
CORBA_tk_array	20L
CORBA_tk_alias	21L

Since this function acquires area to store member TypeCode objects, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

Object

The target TypeCode object.

env

A structure that may contain exception information.

Return Values

For normal termination, the TypeCode object is returned.

For abnormal termination, CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id.

The value of _id and its meaning is as follows:

ex_CORBA_TypeCode_BadKind /* TypeCode attribute information is invalid. */

1.9 Naming Service Interface

This section describes the Naming Service functions.

1.9.1 Naming Context Interface

This section describes the following functions:

CosNaming_NamingContext_bind()

CosNaming_NamingContext_rebind()

CosNaming_NamingContext_bind_context()

CosNaming_NamingContext_rebind_context()

CosNaming_NamingContext_resolve()

CosNaming_NamingContext_unbind()

CosNaming_NamingContext_new_context()

CosNaming_NamingContext_bind_new_context()

CosNaming_NamingContext_destroy()

CosNaming_NamingContext_list()

1.9.1.1 CosNaming_NamingContext_bind()

Name

CosNaming_NamingContext_bind

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
void  CosNaming_NamingContext_bind(
     CosNaming_NamingContext nc,
     CosNaming_Name  *n,
     CORBA_Object  obj,
     CORBA_Environment  *env ) ;
```

Description

This function binds the name specified in n, to the object reference specified in obj, then registers the binding in the naming context specified in nc.

When a compound name is specified in n, the binding is registered in the last naming context specified in the compound name.

Parameters

nc

The object reference of a naming context.

n

The name.

obj

The object reference which binds the name specified for n.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set.

Detailed information is also set in _id of the env structure.

The values of _id and their meaning are as follows:

IDL: CosNaming**/NamingContext/NotFound:1.0**

The naming context specified in n is not found

IDL: CosNaming**/NamingContext/CannotProceed:1.0**

The naming context specified in nc is not found

IDL: CosNaming**/NamingContext/InvalidName:1.0**

The specified name is invalid

IDL: CosNaming**/NamingContext/AlreadyBound:1.0**

The specified name has already been bound to an object reference

1.9.1.2 CosNaming_NamingContext_rebind()

Name

CosNaming_NamingContext_rebind

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
void  CosNaming_NamingContext_rebind(
     CosNaming_NamingContext nc,
     CosNaming_Name *n,
     CORBA_Object obj,
     CORBA_Environment  *env );
```

Description

This function binds the name specified in n, to the object reference specified in obj, then registers the binding in the naming context specified in nc. No error occurs if the specified name has already been bound to an object reference.

When a compound name is specified in n, the binding is registered in the last naming context specified in the compound name.

Parameters

nc

The object reference of a naming context.

n

The name.

obj

The object reference which binds the name specified for n.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set.

Detailed information is also set in _id of the env structure.

The values of _id and their meaning are as follows:

IDL: CosNaming**/NamingContext/NotFound:1.0**

The naming context specified in n is not found

IDL: CosNaming**/NamingContext/CannotProceed:1.0**

The naming context specified in nc is not found

IDL: CosNaming**/NamingContext/InvalidName:1.0**

The specified name is invalid

1.9.1.3 CosNaming_NamingContext_bind_context()

Name

CosNaming_NamingContext_bind_context

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
void  CosNaming_NamingContext_bind_context(
     CosNaming_NamingContext  nc1,
     CosNaming_Name  *n,
     CosNaming_NamingContext  nc2,
     CORBA_Environment  *env );
```

Description

This function binds the name specified in n, to the object reference of the naming context specified in obj, then registers the binding in the naming context specified in nc.

When a compound name is specified in n, the binding is registered in the last naming context specified in the compound name.

Parameters

nc1

The registered object reference of a naming context.

n

The name.

nc2

The object reference which binds the name specified for n.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set.

Detailed information is also set in _id of the env structure.

The values of _id and their meaning are as follows:

IDL: CosNaming**/NamingContext/NotFound:1.0**

The naming context specified in n is not found

IDL: CosNaming**/NamingContext/CannotProceed:1.0**

The naming context specified in nc is not found

IDL: CosNaming**/NamingContext/InvalidName:1.0**

The specified name is invalid

IDL: CosNaming**/NamingContext/AlreadyBound:1.0**

The specified name has already been bound to an object reference

1.9.1.4 CosNaming_NamingContext_rebind_context()

Name

CosNaming_NamingContext_rebind_context

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
void  CosNaming_NamingContext_rebind_context(
     CosNaming_NamingContext  nc1,
     CosNaming_Name  *n,
     CosNaming_NamingContext  nc2,
     CORBA_Environment  *env ) ;
```

Description

This function binds the name specified in n, to the object reference of the naming context specified in obj, then registers the binding in the naming context specified in nc. No error occurs if the specified name has already been bound to an object reference.

When a compound name is specified in n, the binding is registered in the last naming context specified in the compound name.

Parameters

nc1

The registered object reference of a naming context.

n

The name.

nc2

The object reference which binds the name specified for n.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set.

Detailed information is also set in _id of the env structure.

The values of _id and their meaning are as follows:

IDL: CosNaming**/NamingContext/NotFound:1.0**

The naming context specified in n is not found

IDL: CosNaming**/NamingContext/CannotProceed:1.0**

The naming context specified in nc is not found

IDL: CosNaming**/NamingContext/InvalidName:1.0**

The specified name is invalid

1.9.1.5 CosNaming_NamingContext_resolve()

Name

CosNaming_NamingContext_resolve

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
CORBA_Object CosNaming_NamingContext_resolve(
     CosNaming_NamingContext  nc,
     CosNaming_Name  *n,
     CORBA_Environment  *env ) ;
```

Description

This function returns the object reference bound to the name specified in n, in the naming context specified in nc.

Since this function acquires area to store object references, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

nc

The object reference of a naming context.

n

The name.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set.

Detailed information is also set in _id of the env structure.

The values of _id and their meaning are as follows:

IDL: CosNaming**/NamingContext/NotFound:1.0**

The naming context specified in n is not found

IDL: CosNaming**/NamingContext/CannotProceed:1.0**

The naming context specified in nc is not found

IDL: CosNaming**/NamingContext/InvalidName:1.0**

The specified name is invalid

1.9.1.6 CosNaming_NamingContext_unbind()

Name

CosNaming_NamingContext_unbind

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
void CosNaming_NamingContext_unbind(
     CosNaming_NamingContext  nc,
     CosNaming_Name   *n,
     CORBA_Environment  *env ) ;
```

Description

This function deletes the binding of the name specified in n from the naming context specified in nc.

Parameters

nc

The object reference of a naming context.

n

The name which deletes the binding of the naming context.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set.

Detailed information is also set in _id of the env structure.

The values of _id and their meaning are as follows:

IDL: CosNaming**/NamingContext/NotFound:1.0**

The naming context specified in n is not found

IDL: CosNaming**/NamingContext/CannotProceed:1.0**

The naming context specified in nc is not found

IDL: CosNaming**/NamingContext/InvalidName:1.0**

The specified name is invalid

1.9.1.7 CosNaming_NamingContext_new_context()

Name

CosNaming_NamingContext_new_context

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
     CosNaming_NamingContext CosNaming_NamingContext_new_context(
     CosNaming_NamingContext  nc,
     CORBA_Environment *env );
```

Description

This function generates a new naming context in the naming server that manages the naming context specified in nc. It then returns the object reference of the generated naming context.

Since this function acquires area to store object references, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

nc

The object reference of a naming context.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set.

Notes

- Fujitsu recommends using the CosNaming_NamingContext_bind_new_context() method to create and register a naming context object.
- When you create a naming context object using this method, create the binding, then register it in the naming context using the CosNaming_NamingContext_bind_context() method.

1.9.1.8 CosNaming_NamingContext_bind_new_context()

Name

CosNaming_NamingContext_bind_new_context

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
CosNaming_NamingContext  CosNaming_NamingContext_bind_new_context(
     CosNaming_NamingContext nc,
     CosNaming_Name  *n,
     CORBA_Environment  *env );
```

Description

This function generates a new naming context, and binds its object reference to the name specified in n. It then registers the binding in an existing naming context, and returns the object reference of the generated naming context to the calling program.

When a compound name is specified in n, the binding is registered in the last naming context specified in the compound name. The new naming context is generated in the naming server that manages the existing naming context in which the name has been registered.

Since this function acquires area to store object references, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

nc

The object reference of a naming context.

n

The name which binds the object reference of a new naming context.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set.

Detailed information is also set in _id of the env structure.

The values of _id and their meaning are as follows:

IDL: CosNaming**/NamingContext/NotFound:1.0**

The naming context specified in n is not found

IDL: CosNaming**/NamingContext/CannotProceed:1.0**

The naming context specified in nc is not found

IDL: CosNaming**/NamingContext/InvalidName:1.0**

The specified name is invalid

IDL: CosNaming**/NamingContext/AlreadyBound:1.0**

The specified name has already been bound to an object reference

1.9.1.9 CosNaming_NamingContext_destroy()

Name

CosNaming_NamingContext_destroy

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
void  CosNaming_NamingContext_destroy(
     CosNaming_NamingContext  nc,
     CORBA_Environment  *env ) ;
```

Description

This function deletes the naming context specified in nc.

Parameters

nc

The deleted object reference of a naming context.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set.

Detailed information is also set in _id of the env structure.

The value of _id and its meaning is as follows:

IDL:CosNaming/NamingContext/NotEmpty:1.0

Binding is already found in the naming context specified in nc

1.9.1.10 CosNaming_NamingContext_list()

Name

CosNaming_NamingContext_list

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
void  CosNaming_NamingContext_list (
     CosNaming_NamingContext  nc,
     CORBA_unsigned_long  how_many,
     CosNaming_BindingList  **bl,
     CosNaming_BindingIterator  *bi,
     CORBA_Environment  *env ) ;
```

Description

This function returns a list of bindings, up to the maximum value specified in how_many, from a naming context. If the value specified in how_many exceeds the maximum number of bindings specified in the bl_how_many parameter of the nsconfig file (refer to the *nsconfig* command), the Naming Service returns the maximum number of individual bindings specified in the bl_how_many parameter. If 0 is specified in how_many, the client returns bi for access to bindings and bl sequence of length 0.

The list is set in the CosNaming_BindingList structure specified in bi.

CosNaming_BindingList is declared in CosNaming.h as follows:

```
    enum CosNaming_BindingType { nobject, ncontext } ;
    struct CosNaming_Binding {
    Name binding_name ;
    BindingType binding_type ;
    };
    typedef struct {
    CORBA_long _maximum ;
    CORBA_long _length ;
    struct CosNaming_Binding *_buffer ;
    } CosNaming_BindingList ;
```

If the number of bindings in the naming context exceeds the maximum value specified in how_many, an object is generated. This object indicates the current position in the naming context, and the object's reference is returned to bi. The object is called as a binding iterator. The object reference, set in bi, is used while calling the BindingIterator_next_one or BindingIterator_next_n function.

Since this function acquires area to store the binding iterator and the binding list, use CosNaming_BindingIterator_destroy() and CORBA_free() to release the area as soon as it is no longer needed.

Parameters

nc

The object reference of the Naming Service (obtained by the CORBA_ORB_resolve_initial_reference function). (The object reference indicates the initial naming context of the Naming Service.)

Alternatively, the object reference of the naming context (generated by the CosNaming_NamingContext_new_context function).

how_many

The number of obtained bindings lists.

bl

The set area of the bindings list.

bi

The set area of the bindings iterator.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_USER_EXCEPTION is set.

1.9.2 Binding Iterator Interface

This section details the following functions:

CosNaming_BindingIterator_next_one()

CosNaming_BindingIterator_next_n()

CosNaming_BindingIterator_destroy()

1.9.2.1 CosNaming_BindingIterator_next_one()

Name

CosNaming_BindingIterator_next_one

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
CORBA_boolean  CosNaming_BindingIterator_next_one(
     CosNaming_BindingIterator  bi,
     CosNaming_Binding  **b,
     CORBA_Environment  *env ) ;
```

Description

This function fetches a binding from the current position in the naming context (indicated by the binding iterator specified in bi), and sets it in b.

Since this function acquires area to store the binding iterator and the binding, use CosNaming_BindingIterator_destroy() and CORBA_free() to release the area as soon as it is no longer needed.

Parameters

bi

The set area of the bindings iterator.

b

The set area of the bindings.

env

A structure that may contain exception information.

Return Values

When the naming context includes a binding that is not yet listed, CORBA_TRUE is returned. For abnormal termination, CORBA_FALSE is returned.

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.9.2.2 CosNaming_BindingIterator_next_n()

Name

CosNaming_BindingIterator_next_n

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
CORBA_boolean  CosNaming_BindingIterator_next_n(
     CosNaming_BindingIterator  bi,
     CORBA_unsigned_long  how_many,
     CosNaming_BindingList  **bl,
     CORBA_Environment  *env ) ;
```

Description

This function fetches the bindings that follow the current position in the naming context (indicated by the binding iterator specified in bi), and sets them in bi. If the value specified in how_many exceeds the maximum number of bindings specified in the bl_how_many parameter of the nsconfig file (refer to the *nsconfig* command), the Naming Service returns the maximum number of individual bindings specified in the bl_how_many parameter. If 0 is specified in how_many, a BAD_PARAM system exception is returned.

Since this function acquires area to store the binding iterator and the binding list, use CosNaming_BindingIterator_destroy() and CORBA_free() to release the area as soon as it is no longer needed.

Parameters

bi

The set area of the bindings iterator.

how_many

The number of obtained bindings lists.

bl

The set area of the bindings list.

env

A structure that may contain exception information.

Return Values

When the naming context includes a binding that is not yet listed, CORBA_TRUE is returned. For abnormal termination, CORBA_FALSE is returned.

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.9.2.3 CosNaming_BindingIterator_destroy()

Name

CosNaming_BindingIterator_destroy

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
void  CosNaming_BindingIterator_destroy(
     CosNaming_BindingIterator  bi,
     CORBA_Environment  *env ) ;
```

Description

This function discards the binding iterator specified in bi.

Parameters

bi

The discarded bindings iterator.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_USER_EXCEPTION is set.

1.9.3 Naming Context Extended Interface

This section details the following functions:

CosNaming_NamingContextExt_to_string()

CosNaming_NamingContextExt_to_name()

CosNaming_NamingContextExt_to_url()

CosNaming_NamingContextExt_resolve_str()

1.9.3.1 CosNaming_NamingContextExt_to_string()

Name

CosNaming_NamingContextExt_to_string

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
CosNaming_NamingContextExt_StringName  CosNaming_NamingContextExt_to_string(
        CosNaming_NamingContextExt  nce,
        CosNaming_Name  *n,
        CORBA_Environment  *env ) ;
#include <orb.h>
```

Description

This function converts the structural type binding name specified in n into a character string binding name.

This function fetches multiple storage areas to hold return values. When these areas are no longer required, use CORBA_free() to release them.

Parameters

nce

The naming context object reference.

n

The structural type binding name.

env

A structure that may contain exception information.

Return Values

For normal termination, sets CORBA_NO_EXCEPTION in major in the env structure.

For abnormal termination, sets CORBA_USER_EXCEPTION, and detailed information in _id in the env structure.

The value and meaning of _id are as follows:

IDL:CosNaming/NamingContext/InvalidName:1.0

Error in specification of name

1.9.3.2 CosNaming_NamingContextExt_to_name()

Name

CosNaming_NamingContextExt_to_name

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
CosNaming_Name  *CosNaming_NamingContextExt_to_name(
        CosNaming_NamingContextExt  nce,
        CosNaming_NamingContextExt_StringName  sn,
        CORBA_Environment  *env ) ;
```

Description

This function converts the character string type binding name specified in sn into a structural type binding name.

This function fetches multiple storage areas to hold return values. When these areas are no longer required, use CORBA_free() to release them.

Parameters

nce

The naming context object reference.

sn

The character string type binding name.

env

A structure that may contain exception information.

Return Values

For normal termination, sets CORBA_NO_EXCEPTION in major in the env structure.

For abnormal termination, sets CORBA_USER_EXCEPTION, and detailed information in _id in the env structure.

The value and meaning of _id are as follows:

IDL:CosNaming/NamingContext/InvalidName:1.0

Error in specification of name

1.9.3.3 CosNaming_NamingContextExt_to_url()

Name

CosNaming_NamingContextExt_to_url

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
CosNaming_NamingContextExt_URLString  CosNaming_NamingContextExt_to_url(
        CosNaming_NamingContextExt  nce,
        CosNaming_NamingContextExt_Address  addrkey,
        CosNaming_NamingContextExt_StringName  sn,
        CORBA_Environment  *env ) ;
```

Description

This function creates a URL schema from the address specified in addrkey and the character string binding name specified in sn.

This function fetches multiple storage areas to hold return values. When these areas are no longer required, use CORBA_free() to release them.

Parameters

nce

The naming context object reference.

addrkey

The address of the naming context.

sn

The character string type binding name.

env

A structure that may contain exception information.

Return Values

For normal termination, sets CORBA_NO_EXCEPTION in major in the env structure.

For abnormal termination, sets CORBA_USER_EXCEPTION, and detailed information in _id in the env structure.

The value and meaning of _id are as follows:

IDL:CosNaming/NamingContext/InvalidAddress:1.0

Error in specification of address

IDL:CosNaming/NamingContext/InvalidName:1.0

Error in specification of name

1.9.3.4 CosNaming_NamingContextExt_resolve_str()

Name

CosNaming_NamingContextExt_resolve_str

Synopsis

```
#include <orb.h>
#include <CosNaming.h>
CORBA_Object  CosNaming_NamingContextExt_resolve_str(
        CosNaming_NamingContextExt  nce,
        CosNaming_NamingContextExt_StringName  sn,
        CORBA_Environment  *env ) ;
```

Description

This function returns to the naming context specified in nce the object reference bound to the character string binding name specified in sn.

This function fetches multiple storage areas to hold object references. When these areas are no longer required, use CORBA_free() to release them.

Parameters

nce

The naming context object reference.

sn

The character string type binding name.

env

A structure that may contain exception information.

Return Values

For normal termination, sets CORBA_NO_EXCEPTION in major in the env structure.

For abnormal termination, sets CORBA_USER_EXCEPTION, and detailed information in _id in the env structure.

The value and meaning of _id are as follows:

IDL:CosNaming/NamingContext/NotFound:1.0

The name specified in sn was not found.

IDL:CosNaming/NamingContext/CannotProceed:1.0

The naming context specified in nce does not exist.

IDL:CosNaming/NamingContext/InvalidName:1.0

There is an error in the name specification.

1.10 Interface Repository Interface

This section describes the Interface Repository functions.

1.10.1 Type Definitions

Synopsis

```
typedef  CORBA_string  CORBA_Identifier;      /* Identifier */
typedef  CORBA_string  CORBA_ScopedName;      /* Scope name */
typedef  CORBA_string  CORBA_RepositoryId     /* Global name */
enum  CORBA_DefinitionKind {                  /* Object kind */
     dk_none, dk _all,
     dk_Attribute, dk_Constant, dk_Exception, dk_Interface,
     dk_Module, dk_Operation, dk_Typedef,
     dk_Alias, dk_Struct, dk_Union, dk_Enum,
     dk_Primitive, dk_String, dk_Sequence, dk_Array,
     dk_Repository, dk_Wstring
} ;
typedef CORBA_string  CORBA_VersionSpec;      /* Version information */
typedef CORBA_Object  CORBA_IRObject;         /* Object reference of
IRObject */
typedef CORBA_Object  CORBA_Contained;        /* Contained object reference
*/
typedef CORBA_Object  CORBA_Container;        /* Container object reference
*/
typedef CORBA_Object  CORBA_IDLType;          /* IDLType object reference */
typedef CORBA_Object  CORBA_Repository;       /* Repository object reference
*/
typedef CORBA_Object  CORBA_ModuleDef;        /* ModuleDef object reference
*/
typedef CORBA_Object  CORBA_ConstantDef;      /* ConstantDef object
reference */
typedef CORBA_Object  CORBA_TypedefDef;       /* TypedefDef object reference
*/
typedef CORBA_Object  CORBA_StructDef;        /* StructDef object reference
*/
typedef CORBA_Object  CORBA_UnionDef;         /* UnionDef object reference
*/
typedef CORBA_Object  CORBA_EnumDef;          /* EnumDef object reference*/
typedef CORBA_Object  CORBA_AliasDef;         /* AliasDef object reference
*/
typedef CORBA_Object  CORBA_PrimitiveDef;     /* PrimitiveDef object
reference */
typedef CORBA_Object  CORBA_StringDef;        /* StringDef object reference
*/
typedef CORBA_Object  CORBA_SequenceDef;      /* SequenceDef object
reference */
typedef CORBA_Object  CORBA_ArrayDef;         /* ArrayDef object reference
*/
typedef CORBA_Object  CORBA_ExceptionDef;     /* ExceptionDef object
reference */
typedef CORBA_Object  CORBA_AttributeDef;     /* AttributeDef object
reference */
typedef CORBA_Object  CORBA_OperationDef;     /* OperationDef object
reference */
typedef CORBA_Object  CORBA_InterfaceDef;     /* InterfaceDef object
reference */
typedef CORBA_Object  CORBA_ WstringDef;      /* WstringDef object reference
*/

typedef struct {                              /* Object information
```

```
structure */
     CORBA_DefinitionKind  kind;             /* Object kind */
     CORBA_any  value;                       /* Information classified by
object */
} CORBA_Contained_Description;
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     CORBA_Contained  *_buffer;
} CORBA_ContainedSeq                         /* Contained object reference
sequence */
typedef struct {                             /* Object information structure */
     CORBA_Contained  contained_object;      /* Object reference */
     CORBA_DefinitionKind  kind;             /* Object kind */
     CORBA_any  value;                       /* Information classified by
object */
} CORBA_Container_Description;
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     CORBA_Container_Description  *_buffer;
} CORBA_Container_DescriptionSeq;            /* Container_Description
 structure sequence */
typedef struct {                             /* ModuleDef information
structure */
     CORBA_Identifer  name;                  /* Identifier */
     CORBA_RepositoryId  id;                 /* Global name */
     CORBA_RepositoryId  defined_in;         /* Global name of parent
object */
     CORBA_VersionSpec  version;           /* Version information */
} CORBA_ModuleDescription;
typedef struct {                           /* ConstantDef information
structure */
     CORBA_Identifer  name;                /* Identifier */
     CORBA_RepositoryId  id;               /* Global name */
     CORBA_RepositoryId  defined_in;       /* Global name of parent object
*/
     CORBA_VersionSpec  version;           /* Version information */
     CORBA_TypeCode  type;                 /* Type code */
     CORBA_any  value;                     /* Value */
} CORBA_ConstantDescription;
typedef struct {                           /* TypeDef information structure */
     CORBA_Identifer  name;                /* Identifier */
     CORBA_RepositoryId  id;               /* Global name */
     CORBA_RepositoryId  defined_in;       /* Global name of parent object */
     CORBA_VersionSpec  version;           /* Version information */
     CORBA_TypeCode  type;                 /* Type code  */
} CORBA_TypeDescription;
typedef struct {                           /* Structure member structure */
     CORBA_Identifer  name;                /* Identifier */
     CORBA_TypeCode  type;                 /* Type code  */
     CORBA_IDLType  type_def;              /* Member object reference */
} CORBA_StructMember;
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     struct CORBA_StructMember  *_buffer;
} CORBA_StructMemberSeq;                   /* Structure member structure sequence */
typedef struct {                           /* union member structure */
     CORBA_Identifer  name;                /* Identifier */
     CORBA_any  label;                     /* Discriminative value */
     CORBA_TypeCode  type;                 /* Type code */
     CORBA_IDLType  type_def;              /* Member object reference */
```

```
} CORBA_UnionMember;
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     struct CORBA_UnionMember  *_buffer;
} CORBA_UnionMemberSeq;                    /* union member structure sequence */
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     CORBA_string  *_buffer;
} CORBA_EnumMemberSeq;                     /* Enum member sequence */
enum  CORBA_PrimitiveKind {                /* PrimitiveDef kind */
     pk_null, pk_void, pk_short, pk_long, pk_ushort, pk_ulong,
     pk_float, pk_double, pk_boolean, pk_char, pk_octet,
     pk_any, pk_TypeCode, pk_Principal, pk_string, pk_objref,
pk_longlong, pk_longdouble,pk_wchar, pk_wstring
} ;
typedef struct {                           /* ExceptionDef information structure */
     CORBA_Identifer  name;                /* Identifier */
     CORBA_RepositoryId  id;               /* Global name */
     CORBA_RepositoryId  defined_in;       /* Global name of parent object */
     CORBA_VersionSpec  version;           /* Version information */
     CORBA_TypeCode  type;                 /* Type code */
} CORBA_ExceptionDescription;
enum  CORBA_AttributeMode { ATTR_NORMAL, ATTR _READONLY } ; /* Atribute kind */
typedef struct {                           /* AttributeDef information structure */
     CORBA_Identifer  name;                /* Identifier */
     CORBA_RepositoryId  id;               /* Global name */
     CORBA_RepositoryId  defined_in        /* Global name of parent object */
     CORBA_VersionSpec  version;           /* Version information */
     CORBA_TypeCode  type;                 /* Type code */
     CORBA_AttributeMode  mode;            /* Attribute */
} CORBA_AttributeDescription;
enum CORBA_OperationMode { OP_NORMAL, OP _ONEWAY } ;  /* Operation attribute kind */
enum CORBA_ParameterMode { PARAM_IN, PARAM_OUT, PARAM_INOUT };   /* Parameter
                                                                    attribute kind */
typedef struct {                            /* Parameter information structure */
     CORBA_Identifer  name;                 /* Identifier */
     CORBA_TypeCode  type;                  /* Type code */
     CORBA_IDLType   type_def:
     CORBA_ParameterMode  mode;             /* Attribute */
} CORBA_ParameterDescription;
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     struct CORBA_ParameterDescription  *_buffer;
} CORBA_ParDescriptionSeq;                 /* Parameter info structure sequence */
typedef  CORBA_Identifier  CORBA_ContextIdentifier;
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     CORBA_string  *_buffer;
} CORBA_ContextIdSeq;                      /* Context Identifier sequence */
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     CORBA_ExceptionDef  *_buffer;
} CORBA_ExceptionDefSeq;                   /* ExceptionDef object reference sequence */
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     struct CORBA_ExceptionDescription  *_buffer;
} CORBA_ExcDescriptionSeq;                 /* ExceptionDef information structure
```

```
                                        sequence */
typedef struct {                        /* ExceptionDef information structure */
     CORBA_Identifer  name;             /* Identifier */
     CORBA_RepositoryId  id;            /* Global name */
     CORBA_RepositoryId  defined_in;    /* Global name of parent object */
     CORBA_VersionSpec  version;        /* Version information */
     CORBA_TypeCode  result;            /* Type code */
     CORBA_OperationMode  mode;         /* Attribute */
     CORBA_ContextIdSeq  contexts;      /* Context */
     CORBA_ParDescriptionSeq  parameters;   /* Parameter information */
     CORBA_ExcDescriptionSeq  exceptions;   /* Exception information */
} CORBA_OperationDescription;
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     CORBA_InterfaceDef  *_buffer;
} CORBA_InterfaceDefSeq;                /* InterfaceDef info structure sequence */
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     CORBA_string  *_buffer;
} CORBA_RepositoryIdSeq;                /* RepositoryId sequence */
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     struct CORBA_OperationDescription  *_buffer;
} CORBA_OpDescriptionSeq;                /* OperationDef info structure sequence */
typedef struct {
     CORBA_unsigned_long  _maximum;
     CORBA_unsigned_long  _length;
     struct CORBA_AttributeDescription  *_buffer;
} CORBA_AttrDescriptionSeq;              /* AttributeDef info structure sequence */
typedef struct {                         /* Interface information structure */
     CORBA_Identifer  name;              /* Identifier */
     CORBA_RepositoryId  id;              /* Global name */
     CORBA_RepositoryId  defined_in;      /* Global name of parent object */
     CORBA_VersionSpec  version;          /* Version information */
     CORBA_OpDescriptionSeq  operations;  /* Operation information */
     CORBA_AttrDescriptionSeq  attributes;   /* Attribute information */
     CORBA_RepositoryIdSeq  base_interfaces; /* Inherited information */
     CORBA_TypeCode  type;                /* Type code */
} CORBA_InterfaceDef_FullInterfaceDescription;
typedef struct {                          /* InterfaceDef information structure */
     CORBA_Identifer  name;               /* Identifier */
     CORBA_RepositoryId  id;              /* Global name */
     CORBA_RepositoryId  defined_in;      /* Global name of parent object */
     CORBA_VersionSpec  version;          /* Version information */
     CORBA_RepositoryIdSeq  base_interfaces; /* Inherited  information */
} CORBA_InterfaceDescription;
```

1.10.2 IRObject Common Interface

This section explains the functions that can be inherited by each interface object in the Interface Repository, and used as functions of the interface object.

1.10.2.1 CORBA_IRObject__get_def_kind()

Name

CORBA_IRObject__get_def_kind

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_DefinitionKind  CORBA_IRObject__get_def_kind(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the interface type of the Interface Repository object specified in obj.

Parameters

obj

The Interface Repository object.

env

A structure that may contain exception information.

Return Values

For normal termination, this function returns the interface type of the Interface Repository object.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned to _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.3 Contained Common Interface

This section explains the functions that can be inherited by each interface object (which can be contained in another interface object) and used as functions of the interface object. Interface objects contained within another interface object are referred to as Contained Interface Objects.

1.10.3.1 CORBA_Contained__get_id()

Name

CORBA_Contained__get_id

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_RepositoryId  CORBA_Contained__get_id(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the repository ID of the Interface Repository object specified in obj.

Since this function acquires area to store the repository ID, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The Interface Repository object.

env

A structure that may contain exception information.

Return Values

For normal termination, this function returns the repository ID of the specified Interface Repository object.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.3.2 CORBA_Contained__get_name()

Name

CORBA_Contained__get_name

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_Identifier  CORBA_Contained__get_name(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the name of the Interface Repository object specified in obj.

Since this function acquires area to store the object name, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The Interface Repository object.

env

A structure that may contain exception information.

Return Values

For normal termination, the Interface Repository object name (CORBA_Identifier:String type) is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.3.3 CORBA_Contained__get_defined_in()

Name

CORBA_Contained__get_defined_in

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_Container  CORBA_Contained__get_defined_in(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the object reference of an object (container) containing the Interface Repository object indicated by obj.

Since this function acquires area to store the object reference, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

obj

The Interface Repository object.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.3.4 CORBA_Contained_describe()

Name

CORBA_Contained_describe

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_Contained_Description  *CORBA_Contained_describe(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the CORBA_Contained_Description structure that contains the information for the Interface Repository object specified in obj. This information differs with each Interface Repository object, with the unique information being held in the structure's _value member. Value is a structure of CORBA_any type.

The following structure addresses are set at the _value member, depending on the Interface Repository object type. Refer to 1.10.1 Type Definitions for details about each structure.

CORBA_ModuleDef object

ModuleDescription structure

CORBA_ConstantDef object

ConstantDescription structure

CORBA_StructDef object

TypeDescription structure

CORBA_UnionDef object

TypeDescription structure

CORBA_EnumDef object

TypeDescription structure

CORBA_AliasDef object

TypeDescription structure

CORBA_ExceptionDef object

ExceptionDescription structure

CORBA_AttributeDef object

AttributeDescription structure

CORBA_OperationDef object

OperationDescription structure

CORBA_InterfaceDef object

InterfaceDescription structure

Since this function acquires area to store the CORBA_Contained_Description structure, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The Interface Repository object.

env

A structure that may contain exception information.

Return Values

For normal termination, the address of the CORBA_Description structure is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.3.5 Functions Usable when Inherited

(1) CORBA_Contained__get_def_kind()

Refer to 1.10.2 IRObject Common Interface for details about (1).

1.10.4 Container Common Interface

This section explains the functions that interface objects (containers) that can contain other interface objects, can inherit and use.

1.10.4.1 CORBA_Container_lookup()

Name

CORBA_Container_lookup

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_Contained  CORBA_Container_lookup(
     CORBA_Object  obj,
     CORBA_ScopedName  search_name,
     CORBA_Environment  *env );
```

Description

This function retrieves the object of the specified name, from the interface object containing the Interface Repository object specified at obj, and returns its object reference.

If the object to be returned cannot be found, the object reference of NIL (null) is returned and operation ends normally.

Since this function acquires area to store the object reference, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

obj

The Interface Repository object.

search_name

The name (ScopedName) to be used as the retrieval key.

env

A structure that may contain exception information.

Return Values

For normal termination, the specified object's reference is returned.

For abnormal termination, NIL is returned. CORBA_SYSTEM_EXCEPTION is returned in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.4.2 CORBA_Container_contents()

Name

CORBA_Container_contents

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_ContainedSeq  *CORBA_Container_contents(
     CORBA_Object  obj,
     CORBA_DefinitionKind  limit_type,
     CORBA_boolean  exclude_inherited,
     CORBA_Environment  *env );
```

Description

This function returns the object references of the objects (included in the Interface Repository) specified by obj, or included in the Interface Repository by inheritance. The object is specified in obj. The object references are returned in list format.

If the object to be returned cannot be found, 0 is set to _length of the return list and the object reference becomes undefined.

Since this function acquires area to store the list of the object reference, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The Interface Repository object.

limit_type

The included objects of the interface type.

When CORBA_dk_all is specified in limit_type and FALSE is specified in exclude_inherited, the object references of all included and inherited objects are returned.

exclude_inherited

TRUE:

The inherited object is not targeted in the retrieval.

FALSE:

The inherited object is targeted in the retrieval.

env

A structure that may contain exception information.

Return Values

For normal termination, a list of object references of objects found by retrieval is returned.

For abnormal termination, a list of NULL object references is returned. CORBA_SYSTEM_EXCEPTION is returned in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.4.3 CORBA_Container_lookup_name()

Name

CORBA_Container_lookup_name

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_ContainedSeq  *CORBA_Container_lookup_name(
     CORBA_Object  obj,
     CORBA_Identifier  search_name,
     CORBA_long  levels_to_search,
     CORBA_DefinitionKind  limit_type,
     CORBA_boolean  exclude_inherited,
     CORBA_Environment  *env );
```

Description

This function returns the object references of objects included in the Interface Repository specified by obj, or included in the Interface Repository by inheritance. The object is specified in obj. The object references are returned in list format.

If the object to be returned, cannot be found, 0 is set to _length of the return list and the object reference becomes undefined.

Since this function acquires area to store the list of the object reference, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The Interface Repository object.

search_name

The name (Identifier) to be used as the retrieval key.

levels_to_search

The number of hierarchical levels to be searched. If -1 is specified, objects are retrieved in all hierarchical levels. If 1 is specified, only the objects directly under the specified object are retrieved.

limit_type

The included objects of the interface type.

When CORBA_dk_all is specified in limit_type and FALSE is specified in exclude_inherited, the object reference of the specified object is returned along with those of all objects included in (or inherited by) the specified object, in list format.

exclude_inherited

TRUE:

The inherited object is not targeted in the retrieval.

FALSE:

The inherited object is targeted in the retrieval.

env

A structure that may contain exception information.

Return Values

For normal termination, a list of the object references of the objects found is returned.

For abnormal termination, a list of NULL object references is returned. CORBA_SYSTEM_EXCEPTION is returned in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.4.4 CORBA_Container_describe_contents()

Name

CORBA_Container_describe_contents

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_Container_DescriptionSeq  *CORBA_Container_describe_contents(
     CORBA_Object  obj,
     CORBA_DefinitionKind  limit_type,
     CORBA_boolean  exclude_inherited,
     CORBA_long  max_returned_objs,
     CORBA_Environment  *env );
```

Description

This function returns the definition information of the objects (included in the Interface Repository) specified by obj, or included in the Interface Repository by inheritance. The object is specified in obj.

The definition information is returned in Description structure list format (refer to 1.10.1 Type Definitions).

If the object to be returned, cannot be found, 0 is set to _length of the return list and the object reference becomes undefined.

Since this function acquires area to store the list of the Description structure, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The Interface Repository object.

limit_type

The included objects of the interface type.

When CORBA_dk_all is specified in limit_type and FALSE is specified in exclude_inherited, the definition information of all included and inherited objects is returned (up to the maximum number specified in max_returned_objs). To obtain the definition information of all registered objects, specify -1 in max_returned_objs.

exclude_inherited

TRUE:

The inherited object is not targeted in the retrieval.

FALSE:

The inherited object is targeted in the retrieval.

max_returned_objs

The number of objects whose definition information is to be returned.

env

A structure that may contain exception information.

Return Values

For normal termination, a list of object definition information is returned.

For abnormal termination, a list of NULL definition information is returned. CORBA_SYSTEM_EXCEPTION is returned in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.4.5 Functions Usable when Inherited

(1) CORBA_Container__get_def_kind

Refer to 1.10.2 IRObject Common Interface for details about (1).

1.10.5 IDLType Common Interface

This section details the following function:

CORBA_IDLType__get_type()

1.10.5.1 CORBA_IDLType__get_type()

Name

CORBA_IDLType__get_type

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_TypeCode  CORBA_IDLType__get_type(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the type code (TypeCode) of the IDLType object.

Since this function acquires area to store the type code, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

obj

The IDLType object.

env

A structure that may contain exception information.

Return Values

For normal termination, the object TypeCode is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.5.2 Functions Usable when Inherited

(1) CORBA_IDLType__get_def_kind

Refer to 1.10.2 IRObject Common Interface for details about (1).

1.10.6 Repository Interface

This section details the following function:

CORBA_Repository_lookup_id()

1.10.6.1 CORBA_Repository_lookup_id()

Name

CORBA_Repository_lookup_id

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_Contained  CORBA_Repository_lookup_id(
     CORBA_Object  obj,
     CORBA_RepositoryId  search_id,
     CORBA_Environment  *env );
```

Description

This function retrieves objects of the Repository ID specified in search_id, and returns their object references.

If the object to be returned cannot be found, object reference of NIL (null) is returned and the operation ends normally.

Since this function acquires area to store the object reference, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

obj

The Interface Repository object.

search_id

The object with retrieved Repository ID.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference is returned.

For abnormal termination, NULL is returned. CORBA_SYSTEM_EXCEPTION is returned in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.6.2 Functions Usable when Inherited

(1) CORBA_Repository__get_def_kind

(2) CORBA_Repository_lookup

(3) CORBA_Repository_contents

(4) CORBA_Repository_lookup_name

(5) CORBA_Repository_describe_contents

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.4 Container Common Interface for details about (2) - (5).

1.10.7 ModuleDef Interface

This section details the ModuleDef Interface.

1.10.7.1 Functions Usable when Inherited

(1) CORBA_ModuleDef__get_def_kind

(2) CORBA_ModuleDef__get_id

(3) CORBA_ModuleDef__get_name

(4) CORBA_ModuleDef__get_defined_in

(5) CORBA_ModuleDef_describe

(6) CORBA_ModuleDef_lookup

(7) CORBA_ModuleDef_contents

(8) CORBA_ModuleDef_lookup_name

(9) CORBA_ModuleDef_describe_contents

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.3 Contained Common Interface for details about (2) - (5).

Refer to 1.10.4 Container Common Interface for details about (6) - (9).

1.10.8 ConstantDef Interface

This section details the following functions:

CORBA_ConstantDef__get_type()

CORBA_ConstantDef__get_value()

1.10.8.1 CORBA_ConstantDef__get_type()

Name

CORBA_ConstantDef__get_type

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_TypeCode  CORBA_ConstantDef__get_type(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the type code (TypeCode) of the constant definition object specified in obj.

Since this function acquires area to store the type code, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

obj

The constant definition object.

env

A structure that may contain exception information.

Return Values

For normal termination, the TypeCode is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.8.2 CORBA_ConstantDef__get_value()

Name

CORBA_ConstantDef__get_value

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_any  *CORBA_ConstantDef__get_value(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the constant value of the specified ConstantDef object, in any structure format.

Since this function acquires area to store the constant value (any structure), use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The ConstantDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, the constant value (structure of type any) is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.8.3 Functions Usable when Inherited

(1) CORBA_ConstantDef__get_def_kind

(2) CORBA_ConstantDef__get_id

(3) CORBA_ConstantDef__get_name

(4) CORBA_ConstantDef__get_defined_in

(5) CORBA_ConstantDef_describe

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.3 Contained Common Interface for details about (2) - (5).

1.10.9 StructDef Interface

This section details the functions associated with the StructDef Interface.

1.10.9.1 CORBA_StructDef__get_members()

Name

CORBA_StructDef__get_members

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_StructMemberSeq  *CORBA_StructDef__get_members(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns member information of the StructDef object specified in obj, in StructMember structure list format. Refer to 1.10.1 Type Definitions.

Since this function acquires area to store the list of StructMember structure, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The StructDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, the StructDef object member information (StructMember structure) is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.9.2 Functions Usable when Inherited

(1) CORBA_StructDef__get_def_kind

(2) CORBA_StructDef__get_id

(3) CORBA_StructDef__get_name

(4) CORBA_StructDef__get_defined_in

(5) CORBA_StructDef_describe

(6) CORBA_StructDef__get_type

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.3 Contained Common Interface for details about (2) - (5).

Refer to 1.10.5 IDLType Common Interface for details about (6).

1.10.10 UnionDef Interface

This section details the following functions:

CORBA_UnionDef__get_discriminator_type()

CORBA_UnionDef__get_members()

1.10.10.1 CORBA_UnionDef__get_discriminator_type()

Name

CORBA_UnionDef__get_discriminator_type

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_TypeCode  CORBA_UnionDef__get_discriminator_type(
     CORBA_object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the TypeCode of the discriminator information definition of the UnionDef object specified in obj.

Since this function acquires area to store the type code, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

obj

The UnionDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, the TypeCode of the discriminator information definition is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.10.2 CORBA_UnionDef__get_members()

Name

CORBA_UnionDef__get_members

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_UnionMemberSeq  *CORBA_UnionDef__get_members(
     CORBA_object  obj,
     CORBA_Environment  *env );
```

Description

This function returns member information of the UnionDef object specified in obj in UnionMember structure list format (Refer to 1.10.1 Type Definitions).

Since this function acquires area to store the list of UnionMember structure, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The UnionDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, the UnionDef object member information (UnionMember structure) is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.10.3 Functions Usable when Inherited

(1) CORBA_UnionDef__get_def_kind

(2) CORBA_UnionDef__get_id

(3) CORBA_UnionDef__get_name

(4) CORBA_UnionDef__get_defined_in

(5) CORBA_UnionDef_describe

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.3 Contained Common Interface for details about (2) - (5).

1.10.11 EnumDef Interface

This section details the functions associated with the EnumDef Interface.

1.10.11.1 CORBA_EnumDef__get_members()

Name

CORBA_EnumDef__get_members

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_EnumMemberSeq  *CORBA_EnumDef__get_members(
     CORBA_object  obj,
     CORBA_Environment  *env );
```

Description

This function returns member information of the EnumDef object specified in obj, in list format. Refer to 1.10.1 Type Definitions.

Since this function acquires area to store the member information list of the EnumDef object, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The EnumDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, the EnumDef object member information is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.11.2 Functions Usable when Inherited

(1) CORBA_EnumDef__get_def_kind

(2) CORBA_EnumDef__get_id

(3) CORBA_EnumDef__get_name

(4) CORBA_EnumDef__get_defined_in

(5) CORBA_EnumDef_describe

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.3 Contained Common Interface for details about (2) - (5).

1.10.12 AliasDef Interface

This section details the following function:

CORBA_AliasDef__get_original_type_def()

1.10.12.1 CORBA_AliasDef__get_original_type_def()

Name

CORBA_AliasDef__get_original_type_def

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_IDLType  CORBA_AliasDef__get_original_type_def(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the object reference that defines the original type of AliasDef object specified in obj.

Since this function acquires area to store the object reference, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

obj

The AliasDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, this function returns the object reference of the specified IDLType object.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.12.2 Functions Usable when Inherited

(1) CORBA_AliasDef__get_def_kind

(2) CORBA_AliasDef__get_id

(3) CORBA_AliasDef__get_name

(4) CORBA_AliasDef__get_defined_in

(5) CORBA_AliasDef_describe

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.3 Contained Common Interface for details about (2) to (5).

1.10.13 StringDef Interface

This section details the following function:

CORBA_StringDef__get_bound()

1.10.13.1 CORBA_StringDef__get_bound()

Name

CORBA_StringDef__get_bound

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_unsigned_long  CORBA_StringDef__get_bound(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the maximum number of characters of the StringDef object specified in obj.

Parameters

obj

The StringDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, this function returns the maximum number of characters of the StringDef object specified in obj.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.13.2 Functions Usable when Inherited

(1) CORBA_StringDef__get_def_kind

(2) CORBA_StringDef__get_type

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.5 IDLType Common Interface for details about (2).

1.10.14 SequenceDef Interface

This section details the following functions:

CORBA_SequenceDef__get_bound()

CORBA_SequenceDef__get_element_type()

1.10.14.1 CORBA_SequenceDef__get_bound()

Name

CORBA_SequenceDef__get_bound

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_unsigned_long  CORBA_SequenceDef__get_bound(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the maximum number of sequence elements defined in the SequenceDef object specified in obj.

Parameters

obj

The SequenceDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, this function returns the maximum number of sequence elements defined in the SequenceDef object specified in obj.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.14.2 CORBA_SequenceDef__get_element_type()

Name

CORBA_SequenceDef__get_element_type

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_TypeCode  CORBA_SequenceDef__get_element_type(
     CORBA_Object  obj,
     CORBA_Environment  *env);
```

Description

This function returns the type code (TypeCode) that indicates the sequence element type defined in the SequenceDef object specified in obj.

Since this function acquires area to store the type code, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

obj

The SequenceDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, this function returns the TypeCode indicating sequence element type.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env , and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.14.3 Functions Usable when Inherited

(1) CORBA_SequenceDef__get_def_kind

(2) CORBA_SequenceDef__get_type

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.5 IDLType Common Interface for details about (2).

1.10.15 ArrayDef Interface

This section details the following functions:

CORBA_ArrayDef__get_length()

CORBA_ArrayDef__get_element_type()

1.10.15.1 CORBA_ArrayDef__get_length()

Name

CORBA_ArrayDef__get_length

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_unsigned_long  CORBA_ArrayDef__get_length(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the number of array elements of the ArrayDef object specified in obj.

Parameters

obj

The ArrayDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, this function returns the number of array elements.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.15.2 CORBA_ArrayDef__get_element_type()

Name

CORBA_ArrayDef__get_element_type

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_TypeCode  CORBA_ArrayDef__get_element_type(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the type code (TypeCode) that indicates the array element type of the ArrayDef object specified in obj.

Since this function acquires area to store the type code, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

obj

The ArrayDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, this function returns a TypeCode indicating the array element type.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.15.3 Functions Usable when Inherited

(1) CORBA_ArrayDef__get_def_kind

(2) CORBA_ArrayDef__get_type

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.5 IDLType Common Interface for details about (2).

1.10.16 WstringDef Interface

This section details the following function:

CORBA_WstringDef__get_bound()

1.10.16.1 CORBA_WstringDef__get_bound()

Name

CORBA_WstringDef__get_bound

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_unsigned_long  CORBA_WstringDef__get_bound(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the maximum number of characters of the WstringDef object specified in obj.

Parameters

obj

The WstringDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, the maximum number of characters is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.16.2 Functions Usable when Inherited

(1) CORBA_WstringDef__get_def_kind

(2) CORBA_WstringDef__get_type

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.5 IDLType Common Interface for details about (2).

1.10.17 InterfaceDef Interface

This section details the following function:

CORBA_InterfeceDef_describe_interface()

1.10.17.1 CORBA_InterfeceDef_describe_interface()

Name

CORBA_InterfeceDef_describe_interface

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_InterfaceDef_FullInterfaceDescription *CORBA_InterfaceDef_describe_interface(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns definition information of the InterfaceDef object specified in obj, and information about inheritance relationships. The information is returned in FullinterfaceDescription structure format. Refer to 1.10.1 Type Definitions.

Since this function acquires area to store the FullinterfaceDescription structure, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The InterfaceDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, a pointer to the Definition information of InterfaceDef object (including inheritance information) (FullinterfaceDescription structure) is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.17.2 Functions Usable when Inherited

(1) CORBA_InterfaceDef__get_def_kind

(2) CORBA_InterfaceDef__get_id

(3) CORBA_InterfaceDef__get_name

(4) CORBA_InterfaceDef__get_defined_in

(5) CORBA_InterfaceDef_describe

(6) CORBA_InterfaceDef_lookup

(7) CORBA_InterfaceDef_contents

(8) CORBA_InterfaceDef_lookup_name

(9) CORBA_InterfaceDef_describe_contents

(10) CORBA_InterfaceDef__get_type

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.3 Contained Common Interface for details about (2) - (5).

Refer to 1.10.4 Container Common Interface for details about (6) - (9).

Refer to 1.10.5 IDLType Common Interface for details about (10).

1.10.18 OperationDef Interface

This section details the following functions:

CORBA_OperationDef__get_result()

CORBA_OperationDef__get_params()

CORBA_OperationDef__get_contexts()

CORBA_OperationDef__get_exceptions()

1.10.18.1 CORBA_OperationDef__get_result()

Name

CORBA_OperationDef__get_result

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_TypeCode  CORBA_OperationDef__get_result(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the type code (TypeCode) that indicates the operation return value type, and is defined in the OperationDef object specified in obj.

Since this function acquires area to store the type code, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

obj

The OperationDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, the TypeCode indicating the type of operation is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.18.2 CORBA_OperationDef__get_params()

Name

CORBA_OperationDef__get_params

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_ParDescriptionSeq  *CORBA_OperationDef__get_params(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the operation parameter information defined in the OperationDef object specified in obj, in CORBA_ParDescriptionSeq structure list format. Refer to 1.10.1 Type Definitions.

Since this function acquires area to store the list of CORBA_ParDescriptionSeq structure, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The OperationDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, a pointer to the Operation parameter information (CORBA_ParDescriptionSeq structure) is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.18.3 CORBA_OperationDef__get_contexts()

Name

CORBA_OperationDef__get_contexts

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_ContextIdSeq  *CORBA_OperationDef__get_contexts(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the context identifiers for the operation defined in the OperationDef object specified in obj, in list format. Refer to 1.10.1 Type Definitions.

Since this function acquires area to store the list of context identifiers, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The OperationDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, a pointer to the context identifier list (CORBA_ContextIdSeq) is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.18.4 CORBA_OperationDef__get_exceptions()

Name

CORBA_OperationDef__get_exceptions

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_ExceptionDefSeq  *CORBA_OperationDef__get_exceptions(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the definition information of operation exception events defined in the OperationDef object specified in obj, in list format. Refer to 1.10.1 Type Definitions.

Since this function acquires area to store the list of the definition information of exception events, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The OperationDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, a pointer to the definition information of operation exception events is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.18.5 Functions Usable when Inherited

(1) CORBA_OperationDef__get_def_kind

(2) CORBA_OperationDef__get_id

(3) CORBA_OperationDef__get_name

(4) CORBA_OperationDef__get_defined_in

(5) CORBA_OperationDef_describe

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.3 Contained Common Interface for details about (2) - (5).

1.10.19 AttributeDef Interface

This section details the following function:

CORBA_AttributeDef__get_type()

1.10.19.1 CORBA_AttributeDef__get_type()

Name

CORBA_AttributeDef__get_type

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_TypeCode  CORBA_AttributeDef__get_type(
     CORBA_object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the type code (TypeCode) of the AttributeDef object specified in obj.

Since this function acquires area to store the type code, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

obj

The AttributeDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, the AttributeDef object TypeCode is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.19.2 Functions Usable when Inherited

(1) CORBA_AttributeDef__get_def_kind

(2) CORBA_AttributeDef__get_id

(3) CORBA_AttributeDef__get_name

(4) CORBA_AttributeDef__get_defined_in

(5) CORBA_AttributeDef_describe

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.3 Contained Common Interface for details about (2) to (5).

1.10.20 ExceptionDef Interface

This section details the following functions:

CORBA_ExceptionDef__get_type()

CORBA_ExceptionDef__get_members()

1.10.20.1 CORBA_ExceptionDef__get_type()

Name

CORBA_ExceptionDef__get_type

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_TypeCode  CORBA_ExceptionDef__get_type(
     CORBA_Object  obj,
     CORBA_Environment  *env );
```

Description

This function returns the type code (TypeCode) of the ExceptionDef object specified in obj.

Since this function acquires area to store the type code, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

obj

The ExceptionDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, this function returns the TypeCode of the ExceptionDef object.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.20.2 CORBA_ExceptionDef__get_members()

Name

CORBA_ExceptionDef__get_members

Synopsis

```
#include <orb.h>
#include <InterfaceRep.h>
CORBA_StructMemberSeq  *CORBA_ExceptionDef__get_members(
     CORBA_object  obj,
     CORBA_Environment  *env );
```

Description

This function returns definition information of the exception event (CORBA_StructMemberSeq) defined by the ExceptionDef object specified in obj, in list format.

Since this function acquires area to store the list of definition information of the exception event, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

obj

The ExceptionDef object.

env

A structure that may contain exception information.

Return Values

For normal termination, definition information of the exception event (CORBA_StructMemberSeq) is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is returned in _major of the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.10.20.3 Functions Usable when Inherited

(1) CORBA_ExceptionDef__get_def_kind

(2) CORBA_ExceptionDef__get_id

(3) CORBA_ExceptionDef__get_name

(4) CORBA_ExceptionDef__get_defined_in

(5) CORBA_ExceptionDef_describe

Refer to 1.10.2 IRObject Common Interface for details about (1).

Refer to 1.10.3 Contained Common Interface for details about (2) to (5).

1.11 Other Functions

This section describes the CORBA defined functions not included in the interfaces of the previous sections, and the Fujitsu extended interface.

1.11.1 CORBA_send_multiple_requests()

Name

CORBA_send_multiple_requests

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_send_multiple_requests(
     CORBA_Request  req[],
     CORBA_Environment  *env,
     CORBA_long  count,
     CORBA_Flags  invoke_flags);
```

Description

This function sends one or more requests in parallel.

Like the CORBA_Request_send() function, this function returns control to the calling program, without waiting for the operation to complete.

Parameters

req

The value specified in count as the request object corresponding to the number of requests to be sent.

env

A structure that may contain exception information.

count

The number of request objects for the request to be sent.

invoke_flags

The following flags can be specified in invoke_flags:

CORBA_INV_NO_RESPONSE:

The calling program neither waits for a response, nor expects that the output parameter (inout or out) will be updated. This can be specified even when the operation is not defined as one way.

CORBA_INV_TERM_ON_ERR:

If one request causes an error, the remaining requests are not sent.

Completion of the server application function can be determined by using the CORBA_Request_get_response() and CORBA_Request_get_next_response() functions together.

Return Values

For normal termination, CORBA_OK is returned. For abnormal termination, CORBA_FAILED is returned.

1.11.2 CORBA_get_next_response()

Name

CORBA_get_next_response

Synopsis

```
#include <orb.h>
CORBA_ORBStatus  CORBA_get_next_response(
     CORBA_Environment  *env,
     CORBA_Flags  response_flags,
     CORBA_Request  *req);
```

Description

This function returns the object reference of the next request (sent using CORBA_send_multiple_requests()) to be terminated. The sequence of requests that have already been terminated is not assured.

Parameters

env

A structure that may contain exception information.

invoke_flags

The following flag can be specified in response_flags:

CORBA_RESP_NO_WAIT:

This function returns control even when the specified request is in progress.

req

A pointer to the area for the request object that will end next.

Return Values

For normal termination, CORBA_OK is returned.

For abnormal termination, CORBA_FAILED is returned.

1.11.3 CORBA_xx_alloc()

Name

CORBA_xx_alloc

Allocates the data type memory area.

Synopsis

```
#include <orb.h>
     CORBA_short              *CORBA_short_alloc( ) ;
     CORBA_long               *CORBA_long_alloc( ) ;
     CORBA_unsigned_short     *CORBA_unsigned_short_alloc( ) ;
     CORBA_unsigned_long      *CORBA_unsigned_long_alloc( ) ;
     CORBA_long_long          *CORBA_long_long_alloc( ) ;
     CORBA_float              *CORBA_float_alloc( ) ;
     CORBA_double             *CORBA_double_alloc( ) ;
     CORBA_long_double        *CORBA_long_double_alloc( ) ;
     CORBA_char               *CORBA_char_alloc( ) ;
     CORBA_wchar              *CORBA_wchar_alloc( ) ;
     CORBA_boolean            *CORBA_boolean_alloc( ) ;
     CORBA_octet              *CORBA_octet_alloc( ) ;
     CORBA_enum               *CORBA_enum_alloc( ) ;
     CORBA_any                *CORBA_any_alloc( ) ;
     CORBA_string             CORBA_string_alloc( CORBA_unsigned_long len );
     CORBA_wstring            CORBA_wstring_alloc( CORBA_unsigned_long len );
     CORBA_string             *CORBA_string_ptr_alloc() ;
```

```
    CORBA_wstring            *CORBA_wstring_ptr_alloc() ;
    CORBA_Object             CORBA_Object_alloc() ;
```

Description

The CORBA_string_alloc() function and CORBA_wstring_alloc() allocate the string area with the size specified in len (bytes) + 1 byte, and returns the first pointer of the string.

The CORBA_string_ptr_alloc() function and CORBA_wstring_ptr_alloc() allocate the storage area with the pointer of CORBA_string or CORBA_wstring, and returns its address.

Other CORBA_xx_alloc() functions allocate an area with the size of data type xx, and return its address. Use the CORBA_free() function to release the retrieved area.

Note

This function always returns NULL if CORBA_ORB_init() has not been invoked.

1.11.4 CORBA_xx_allocbuf()

Name

CORBA_xx_allocbuf

Retrieves memory for data type array

Synopsis

```
#include <orb.h>
CORBA_octet   *CORBA_octet_seq_allocbuf(
     CORBA_unsigned_long  len );
CORBA_char    *CORBA_char_seq_allocbuf(
     CORBA_unsigned_long  len );
CORBA_Request *CORBA_Request_seq_allocbuf(
     CORBA_unsigned_long  len );
CORBA_octet   *CORBA_ReferenceData_allocbuf(
     CORBA_unsigned_long  len );
CORBA_string  *CORBA_sequence_string_allocbuf(
     CORBA_unsigned_long  len );
```

Description

The CORBA_xx_allocbuf() function retrieves the area specified at len for an array specified in _buffer of sequence type data (defined by name CORBA_xx), and returns its pointer.

Use the CORBA_free() function to release the retrieved data.

Parameters

len

The number of areas.

Return value

A pointer to the beginning of the allocated areas.

Note

The IDL compiler automatically generates a basic type sequence by defining it in each IDL file.

This function always returns NULL if CORBA_ORB_init() has not been invoked.

1.11.5 CORBA_free()

Name

CORBA_free

Releases the memory area

Synopsis

```
#include <orb.h>
void  CORBA_free(
void  *mem ) ;
```

Description

This function releases the pointer area specified in mem.

Parameters

mem

A pointer to the memory area to be freed.

Return value

None.

1.11.6 CORBA_any_get_release()

Name

CORBA_any_get_release

References the release flag

Synopsis

```
#include <orb.h>
CORBA_boolean  CORBA_any_get_release(
     CORBA_any  *any ) ;
```

Description

This function references the release flag for the area identified by any, and then returns the value of the flag.

Refer to information on any Type and sequence Type Release Flags in the Distributed Application Development Guide (CORBA Service Edition) for details about the Release Flag.

Parameters

any

A pointer to the CORBA_any area.

Return value

The release flag for the area identified by any. Valid values include one of the following:

CORBA_TRUE : Released.

CORBA_FALSE : Not released.

1.11.7 CORBA_any_set_release()

Name

CORBA_any_set_release

Sets the release flag

Synopsis

```
#include <orb.h>
void CORBA_any_set_release(
     CORBA_any  *any,
     CORBA_boolean  bl ) ;
```

Description

This function sets the release flag that specifies whether to release the area specified in any.

Refer to information on any Type and sequence Type Release Flags in the Distributed Application Development Guide (CORBA Service Edition) for details about the Release Flag.

Parameters

any

A pointer to the CORBA_any area.

bl

The following values can be specified:

CORBA_TRUE : Released.

CORBA_FALSE : Not released.

Return value

None.

1.11.8 CORBA_sequence_get_release()

Name

CORBA_sequence_get_release

References the release flag

Synopsis

```
#include <orb.h>
CORBA_boolean  CORBA_sequence_get_release(
     void  *seq ) ;
```

Description

This function references the release flag for the area identified by seq, and then returns the value of the flag.

Refer to information on any Type and sequence Type Release Flags in the Distributed Application Development Guide (CORBA Service Edition) for details about the Release Flag.

Parameters

seq

A pointer to the sequence-type area.

Return value

The release flag of the area specified in seq. One of the following values:

CORBA_TRUE : Released.

CORBA_FALSE : Not released.

1.11.9 CORBA_sequence_set_release()

Name

CORBA_sequence_set_release

Sets the release flag

Synopsis

```
#include <orb.h>
void  CORBA_sequence_set_release(
     void  *seq,
     CORBA_boolean  bl ) ;
```

Description

This function sets the release flag that specifies whether to release the area specified in seq.

Refer to information on any Type and sequence Type Release Flags in the Distributed Application Development Guide (CORBA Service Edition) for details about the Release Flag.

Parameters

seq

A pointer to the sequence-type area.

bl

The release flag to be set for seq. The following values can be specified for bl:

CORBA_TRUE : Released.

CORBA_FALSE : Not released.

Return value

None.

1.11.10 CORBA_ORB_net_disconnect()

Name

CORBA_ORB_net_disconnect (release communication resource)

Synopsis

```
void
CORBA_ORB_net_disconnect(
     CORBA_ORB orb,
          CORBA_Object obj,
          CORBA_Environment  *env ) ;
```

Description

CORBA_ORB_net_disconnect releases the communication resource used by the object specified in obj.

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

obj

The object reference to be disconnected.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXECEPTION is returned. For abnormal termination, CORBA_SYSTEM_EXECPTION is returned as follows.

IDL:CORBA/StExcep/INV_OBJREF:1.0

Invalid Object specified.

IDL:CORBA/StExcep/COMM_FAILURE:1.0

Already disconnected.

Notes

- Depending on the status of the client application, the following error messages may be issued when this function is invoked.
  - CORBA_COMM_FAILURE is issued if the application is using multithreading, and a thread other than the one invoking this function is in communication with the server.
  - Asynchronous request results cannot be received when an asynchronous request has been issued.
  - If this function is issued for the same obj, then CORBA_INV_OBJREF is reported.
  - If the connection information for the server machine does not exist, CORBA_INV_OBJREF or CORBA_COMM_FAILURE is posted.
- At the same time, this operation also releases instance information held by other objects sharing the server which hosts the object to be released, obj, specified in the parameters of the invoking client process.

1.11.11 CORBA_ORB_set_client_timer()

Name

CORBA_ORB_set_client_timer

Sets server method wait time

Synopsis

```
void
CORBA_ORB_set_client_timer(
        CORBA_ORB               orb,
        CORBA_Object            obj,
        CORBA_long              time,
        CORBA_Environemnt       *env );
```

Description

Sets in "time" the wait time until the server method for the server object specified in obj is returned in client applications. The wait time specified is valid for all server objects operating on the host specified in "obj".

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

obj

The object reference to the server object whose response time is to be set.

time

Response time until the server method returns (seconds). The response time can be specified by a value from 0 to 100000000. If 0 is specified, the response time until the server method returns will not be monitored.

env

A structure that may contain exception information.

Return Values

There is no return value at normal termination.

The following SystemExecptions are caused by abnormal termination:

IDL:CORBA/StExcep/INV_OBJREF:1.0

An invalid object was specified in obj.

IDL:CORBA/StExcep/BAD_PARAM:1.0

An invalid value was specified in time.

IDL:CORBA/StExcep/NO_MEMORY:1.0

There is insufficient memory.

Notes

- This method is used when the standby time of the client processing must be modified from the standby time set in the config file (period_receive_timeout x 5 seconds).
- Wait time is specified in process units by issuing this method after the server object object reference has been fetched, and before requests are sent to the server.
- To modify the wait time while a task is running, release the connection information for the relevant object with CORBA_ORB_net_disconnect, and then issue this method.
- However, if a request is made to the same server before the server object object reference has been fetched or while it is being fetched, the connection information must be released with CORBA_ORB_net_disconnect() before this method is issued. This would apply, for example, if CosNaming_NamingContext_resolve() was used when the server object object reference was fetched, and the Naming Service and server application were both running on the same machine.

1.11.12 CORBA_ORB_set_client_request_timer()

Name

CORBA_ORB_set_client_request_timer

Set the server method response time for the thread

Synopsis

```
void
CORBA_ORB_set_client_request_timer(
        CORBA_ORB              orb,
        CORBA_long             time,
        CORBA_Environemnt      *env );
```

Description

This function sets the response time during which the client application will wait until the server method returns. This time is actually specified by the parameter time. The response time set here is effective on all the requests that are issued by the thread that invokes this function. Once this function is invoked, CORBA_ORB_clear_client_request_timer() must also be invoked before the thread terminates.

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

time

Response time until the server method returns (seconds). The response time can be specified by a value from 0 to 100000000. If 0 is specified, the response time until the server method returns will not be monitored.

env

A structure that may contain exception information.

Return Values

When this method terminates normally, no value is returned.

If this method does not terminate normally, CORBA_SYSTEM_EXCEPTION is assigned to _major and detailed information is assigned to _id and _minor in the env structure. The meanings of _id values are indicated below. For the meanings of _minor values, refer to 'Exception Information Minor Codes to be Reported from the CORBA Service' in the Messages.

IDL:CORBA/StExcep/BAD_PARAM:1.0

An invalid value was specified in time.

IDL:CORBA/StExcep/NO_MEMORY:1.0

There is insufficient memory.

Notes

- Use this method if you need to change the response time during which the client process waits.
- When invoked before requests are sent to the server, this method sets the response time for the thread.
- If this method is invoked, CORBA_ORB_clear_client_request_timer() must also be invoked before the thread terminates; otherwise, a memory leak will occur.
- To change the timeout value once this method is invoked, re-issue CORBA_ORB_set_client_request_timer(). When doing so, you do not need to invoke CORBA_ORB_clear_client_request_timer() beforehand.
- The priorities of response time settings are indicated below, in descending order.
  1. Setting made by CORBA_ORB_set_client_request_timer().
  2. Setting made by CORBA_ORB_set_client_timer().
  3. Setting made by the period_receive_timeout parameter in the CORBA service environment setup file (config).

1.11.13 CORBA_ORB_get_client_request_timer()

Name

CORBA_ORB_get_client_request_timer

Acquire the response time of a server method.

Synopsis

```
CORBA_long
CORBA_ORB_get_client_request_timer(
```

```
        CORBA_ORB               orb,
        CORBA_Object            obj,
        CORBA_Environemnt       *env );
```

Description

This method acquires the server method response time (in seconds).

The priorities of response time settings are indicated below, in descending order. If CORBA_OBJECT_NIL is assigned to obj, this method will acquire the setting 1 or 3.

1. Setting made by CORBA_ORB_set_client_request_timer()
2. Setting made by CORBA_ORB_set_client_timer()
3. Setting made by the period_receive_timeout parameter in the CORBA service environment setup file (config)

Use this function when you want to see the response time of a server method in a client application.

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

obj

The object reference to the server object whose response time is to be acquired.

env

A structure that may contain exception information.

Return Values

The server method response time returns.

If this function abnormally terminates, -1 returns. In the env structure, CORBA_SYSTEM_EXCEPTION is assigned to _major and detailed information is assigned to _id and _minor. For the meanings of _id and _minor values, refer to 'Exception Information Minor Codes to be Reported from the CORBA Service' in the Messages.

1.11.14 CORBA_ORB_clear_client_request_timer()

Name

CORBA_ORB_clear_client_request_timer

Reset the server method response time for the thread.

Synopsis

```
void
CORBA_ORB_clear_client_request_timer(
        CORBA_ORB               orb,
        CORBA_Environemnt       *env );
```

Description

This function resets the server method response time set by CORBA_ORB_set_client_request_timer(). After execution of this function, the server method response time during which requests issued by the thread that invoked this function will wait returns to the system default value.

If the server method response time was set by CORBA_ORB_set_client_request_timer(), you need to invoke this function before terminating the thread. This function does nothing if it is invoked by a thread that has not set any server method response time by CORBA_ORB_set_client_request_timer().

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

env

A structure that may contain exception information.

Return Values

When this method terminates normally, no value is returned.

If this method does not terminate normally, CORBA_SYSTEM_EXCEPTION is assigned to _major and detailed information is assigned to _id and _minor in the env structure. For the meanings of _id and _minor values, refer to 'Exception Information Minor Codes to be Reported from the CORBA Service' in the Messages.

1.11.15 CORBA_ORB_register_reply_interceptor()

Name

CORBA_ORB_register_reply_interceptor

Synopsis

```
void
CORBA_ORB_register_reply_interceptor(
        CORBA_ORB               orb,
        void                    (*funcptr)(void),
        CORBA_Environemnt       *env );
```

Description

The function specified in the function pointer (funcptr) of the argument is registered as the exit function. The exit function that is registered is executed after the server application interface implementation function completes processing and sends a reply to the client.

The exit function is executed after interface implementation function processing is complete even if no reply is sent to the client, for example when 'oneway' is specified in the interface implementation function.

Parameters

orb

The ORB object reference acquired by CORBA_ORB_init().

funcptr

This is the function pointer of the exit function.

env

A structure that may contain exception information.

Return Values

When this method terminates normally, no value is returned.

If this method does not terminate normally, CORBA_SYSTEM_EXCEPTION is assigned to _major and detailed information is assigned to _id and _minor in the env structure. For the meanings of _id and _minor values, refer to 'Exception Information Minor Codes to be Reported from the CORBA Service' in the Messages.

IDL:CORBA/StExcep/BAD_INV_ORDER:1.0

This function was issued after the activation function (CORBA_BOA_impl_is_ready) was issued.

IDL:CORBA/StExcep/BAD_PARAM:1.0

NULL was specified in the function pointer (funcptr) of the exit function.

IDL:CORBA/StExcep/NO_IMPLEMENT:1.0

The client library (ODWINCPP.LIB) is being used.

IDL:CORBA/StExcep/NO_RESOURCE:1.0

The exit function has already been registered.

Notes

- The exit function format must be as follows:

```
void exit function name()
```

- One exit function can be registered per application.
- This function must be issued before the activation function (CORBA_BOA_impl_is_ready) is issued.
- If the exit function processing registered in this function is not completed within the monitoring time (set in 'reply_interceptor_timeout' of the implementation repository definition; the default is 300 seconds), the application is stopped by force. For this reason, it is recommended that the application is run on a WorkUnit that contains the automatic application restart functionality when using this function.
- This function cannot be used in the client library (ODWINCPP.LIB).

Example

```
void
exitfunc()
{
    /* exit function processing */
}

int
main( int argc, char *argv[] )
{
    CORBA_ORB  orb;
    CORBA_Environment  env;

    orb = CORBA_ORB_init( &argc, argv, FJ_OM_ORBid, &env );

    CORBA_ORB_register_reply_interceptor( orb, exitfunc, &env );

    ...
}
```

1.11.16 CORBA_exception_id()

Name

CORBA_exception_id

Synopsis

```
#include <orb.h>
CORBA_string  CORBA_exception_id(
        CORBA_Environment      *env );
```

Description

Returns an identifier used to identify the exception set in the env structure.

Parameters

env

The structure that contains the target exception information.

Return Values

When an exception is set, an identifier to identify the env structure exception is returned. When an exception is not set (when CORBA_NO_EXCEPTION is set in _major in the env structure), an empty string is returned.

1.11.17 CORBA_exception_value()

Name

CORBA_exception_value

Synopsis

```
#include <orb.h>
 void*  CORBA_exception_value(
        CORBA_Environment      *env );
```

Description

Returns a parameter used to identify the exception set in the env structure.

Parameters

env

The structure that contains the target exception information.

Return Values

When an exception is set, a parameter to identify the env structure exception is returned. When an exception is not set (when CORBA_NO_EXCEPTION is set in _major in the env structure), a NULL pointer is returned.

1.11.18 CORBA_exception_free()

Name

CORBA_exception_free

Synopsis

```
#include <orb.h>
void  CORBA_exception_free(
        CORBA_Environment      *env );
```

Description

Frees the parameter area set in the env structure.

Parameters

env

The structure that contains the target exception information.

Return Values

None.

1.11.19 FJ_ImplementationRep_lookup_id()

Name

FJ_ImplementationRep_lookup_id

Synopsis

```
#include <orb.h>
CORBA_Object  FJ_ImplementationRep_lookup_id(
     CORBA_Object obj,
          FJ_RepositoryID id,
          CORBA_Environment  *env ) ;
```

Description

This function returns the CORBA_ImplementationDef object that corresponds to the Implementation Repository ID specified in impl_id.

The CORBA_ImplementationDef object is the Implementation Repository ID of the server application registered by the OD_impl_inst command.

Windows32/64

If this function is executed on an application linked with Odwin.dll, it cannot find the implementation repository ID. This will result in a user exception whose _id value is "IDL:FJ/NameDoesntExist:1.0".

Parameters

impl_rep

The object reference to implementation information.

impl_id

The implementation repository ID.

env

The structure that contains the target exception information.

Return Values

For normal termination, the CORBA_ImplementationDef object corresponding to impl_id is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id and _minor. Refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service for information on _id and _minor.

1.12 Server Application Interface Windows32 Solaris32 Linux32

This section describes the memory management application interfaces (API) required by server applications to transfer data to and from skeletons.

The following APIs are available:

- string type memory acquisition API(TD_string_alloc)
- memory release API(TD_free)
- SMO_name acquisition API(TD_get_smo_name)
- client identifier acquisition API(TD_get_client_id)
- user identifier acquisition API(TD_get_user_information)
- session ID notice API(TD_getsessionid)

- session continuation declaration API(TD_setcontcvt)
- session ID reference API(TD_refsessionid)

Each API is described below.

1.12.1 TD_string_alloc Windows32 Solaris32 Linux32

Name

TD_string_alloc

Synopsis

```
#include "td_alc.h"
CORBA_string  TD_string_alloc( unsigned long length );
```

Description

This is equivalent to CORBA_string_alloc. It acquires a dynamic string area of the size specified in length

Return Values

In the case of normal termination, a pointer to the acquired data is returned.

In the event of abnormal termination, a NULL pointer is returned.

1.12.2 TD_wstring_alloc Windows32 Solaris32 Linux32

Name

TD_wstring_alloc

Synopsis

```
#include "td_alc.h"
CORBA_string  TD_wstring_alloc( unsigned long length );
```

Description

This is equivalent to CORBA_wstring_alloc. It acquires a dynamic string area of the size specified in length

Return Values

In the case of normal termination, a pointer to the acquired data is returned.

In the event of abnormal termination, a NULL pointer is returned.

1.12.3 TD_free Windows32 Solaris32 Linux32

Name

TD_free

Synopsis

```
#include "td_alc.h"
void TD_free( void* );
```

Description

This API is equivalent to CORBA_free. It releases the area for the specified pointer.

Return Values

None.

1.12.4 TD_get_smo_name Windows32 Solaris32 Linux32

Name

TD_get_smo_name

Synopsis

```
#include td_alc.h
long TD_ get_smo_name ( char *smoname );
```

Description

This API stores the SMO name in the memory location set in smoname. Allocate 256 bytes of memory for the SMO name.

Return Values

For normal termination, NULL is returned.

For abnormal termination, one of the following values are returned:

1: Parameter error

2: Protocol error

3: Sequence error

99: System error

1.12.5 TD_get_client_id Windows32 Solaris32 Linux32

Name

TD_get_client_id

Synopsis

```
#include td_alc.h
long TD_get_client_id( CORBA_sequence_octet *addr);
```

Description

This API sets the client identifier to the buffer area of the sequence control area specified by addr.

Return Values

For normal termination, NULL is returned.

For abnormal termination, one of the following values are returned:

1: Parameter error

2: Protocol error

3: Sequence error

99: System error

1.12.6 TD_get_user_information Windows32 Solaris32 Linux32

Name

TD_get_user_information

Synopsis

```
#include "td_alc.h"
long TD_get_user_information ( TD_USER_INFORMATION *addr );
```

Description

This API sets the length of the user identifier information to the length of the TD_USER_INFO area specified by addr, and the user identifier to userinformation.

/* TD_USER_INFORMATION AREA */

typedef struct TD_user_information_t{

long length;

char userinformation[TD_USER_INFORMATION_MAXIMAM_LENGTH];

} TD_USER_INFORMATION;

Return Values

For normal termination, NULL is returned.

For abnormal termination, one of the following values are returned:

1: Parameter error

2: Protocol error

3: Sequence error

99: System error

1.12.7 TD_getsessionid() Windows32 Solaris32 Linux32

Name

TD_getsessionid()

Synopsis

```
#include "td_alc.h"
long TD_getsessionid(TD_SESSION_ID *sessionid_p)
```

Description

This API acquires and transmits the key (session ID) required to implement the following functions:

- Process Binding Function
- Session information management
- Windows32 Solaris32
  AIM linkage session inheritance

In sessionid_p, set the pointer to the session ID structure.

Return Values

If the operation terminates normally, 0 is transmitted as return information and the session ID is stored in the area specified by the parameter.

In the event of abnormal termination, one of the following values is returned.

1 : Parameter error

2 : Protocol error (double execution of TD_getsessionid)

99 : System error

1.12.8 TD_setcontcvt() Windows32 Solaris32 Linux32

Name

TD_setcontcvt()

Synopsis

```
#include "td_alc.h"
long TD_setcontcvt()
```

Description

When this API is issued, a session continuation is declared to the system.

Return Values

For normal termination, 0 is transmitted as return information.

In the event of abnormal termination, one of the following values is returned:

1 : Parameter error

2 : Protocol error (Invoked by non-resident application, process binding definition not made, or session ID not issued.)

99 : System error

1.12.9 TD_refsessionid() Windows32 Solaris32 Linux32

Name

TD_refsessionid()

Synopsis

```
#include  "td_alc.h"
long  TD_refsessionid(TD_SESSION_ID *sessionid_p)
```

Description

This operation is used when the user application refers to a previously acquired session ID.

The session information storage area must be acquired before this API is issued. In sessionid_p, specify the area address for locating the session continuation information.

Return Values

If the operation terminates normally, 0 is transmitted as return information and the session ID is stored in the area specified by sessionid_p. In the event of abnormal termination, one of the following values is returned:

1 : Parameter error

99 : System error

1.13 Current Interface Windows32/64 Solaris32 Linux32/64

This section details the following functions:

CosTransactions_Current_begin()

CosTransactions_Current_commit()

CosTransactions_Current_rollback()

CosTransactions_Current_rollback_only()

CosTransactions_Current_get_status()

CosTransactions_Current_get_transaction_name()

CosTransactions_Current_set_timeout()

CosTransactions_Current_get_control()

CosTransactions_Current_suspend()

CosTransactions_Current_resume()

1.13.1 CosTransactions_Current_begin() Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Current_begin

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
void  CosTransactions_Current_begin(
     CORBA_Object current,
     CORBA_Environment       *env);
```

Description

This function creates a new transaction. Specify the object reference of the 1.13 Current Interface obtained by the CORBA_ORB_resolve_initial_references method (CORBA_ORB_ObjectId_TransactionCurrent), in current.

If no timeout is set in CosTransactions_Current_set_timeout, the timeout specified for TRAN_TIME_OUT in the operating environment file is assumed.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of structure env.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major of structure env, and detailed information is set in _id of structure env.

The values of _id and their meanings are described below:

ex_CosTransactions_SubtransactionsUnavailable:

An attempt was made to create a nested transaction.

ex_CORBA_StExcep_NO_IMPLEMENT:

The Database Linkage Service system is not started.

ex_CORBA_StExcep_COMM_FAILURE:

Possible causes are shown below.

- A communication error occurred.
- Host information, on which the Database Linkage Service system exists, is not defined in the OD environment file initial_hosts (inithost on PC).
- The Link order of the library provided by the Database Linkage Service is specified after the library provided by the OD.

ex_CORBA_StExcep_NO_RESOURCES:

Insufficient resources. The maximum number of transactions may have been exceeded.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

1.13.2 CosTransactions_Current_commit() Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Current_commit

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
void  CosTransactions_Current_commit(
     CORBA_Object current,
     CORBA_boolean           report_heuristics,
     CORBA_Environment       *env);
```

Description

This function commits a transaction. Specify the object reference of the 1.13 Current Interface obtained by the

CORBA_ORB_resolve_initial_references method

(CORBA_ORB_ObjectId_TransactionCurrent), in current.

If CORBA_TRUE is specified for report_heuristics, a heuristic error is reported.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of structure env.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major of structure env, and detailed information is set in _id of structure env.

The values of _id and their meanings are described below:

ex_CosTransactions_NoTransaction:

No transaction has been created.

ex_CORBA_StExcep_NO_PERMISSION:

Permission to commit transactions is not denied.

ex_CosTransactions_HeuristicMixed:

Some transactions are in the commit completion state, and others are in the rollback completion state.

ex_CosTransactions_HeuristicHazard:

The transaction completion status is unknown.

ex_CORBA_StExcep_TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

ex_CORBA_StExcep_NO_IMPLEMENT:

The Database Linkage Service system or Resource Manager is not started.

ex_CORBA_StExcep_COMM_FAILURE:

A communication error occurred.

ex_CORBA_StExcep_NO_RESOURCES:

Insufficient resources. The maximum number of transactions may have been exceeded.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

1.13.3 CosTransactions_Current_rollback() Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Current_rollback

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
void  CosTransactions_Current_rollback(
     CORBA_Object current,
     CORBA_Environment       *env);
```

Description

This function rolls back a transaction. Specify the object reference of the 1.13 Current Interface obtained by

CORBA_ORB_resolve_initial_references method

(CORBA_ORB_ObjectId_TransactionCurrent), in current.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of structure env.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major of structure env, and detailed information is set in _id of structure env.

The values of _id and their meanings are described below:

ex_CosTransactions_NoTransaction:

No transaction has been created.

ex_CORBA_StExcep_NO_PERMISSION:

Permission to roll back transactions is denied.

ex_CORBA_StExcep_TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

ex_CORBA_StExcep_NO_IMPLEMENT:

The Database Linkage Service system or Resource Manager is not started.

ex_CORBA_StExcep_COMM_FAILURE:

A communication error occurred.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

1.13.4 CosTransactions_Current_rollback_only() Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Current_rollback_only

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
```

```
void  CosTransactions_Current_rollback_only(
     CORBA_Object current,
     CORBA_Environment       *env);
```

Description

This function enables transaction rollback. Specify the object reference of the 1.13 Current Interface obtained using the CORBA_ORB_resolve_initial_references method

(CORBA_ORB_ObjectId_TransactionCurrent), in current.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of structure env.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major of structure env, and detailed information is set in _id of structure env.

The values of _id and their meanings are described below:

ex_CosTransactions_NoTransaction:

No transaction has been created.

ex_CORBA_StExcep_TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

ex_CORBA_StExcep_NO_IMPLEMENT:

The Database Linkage Service system is not started.

ex_CORBA_StExcep_COMM_FAILURE:

A communication error occurred.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

1.13.5 CosTransactions_Current_get_status() Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Current_get_status

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
CosTransactions_Status  CosTransactions_Current_get_status(
     CORBA_Object current,
     CORBA_Environment       *env);
```

Description

This function returns the status of a transaction. Specify the object reference of the 1.13 Current Interface obtained using the

CORBA_ORB_resolve_initial_references method

(CORBA_ORB_ObjectId_TransactionCurrent), in current.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of structure env, and one of the following values are returned:

ex_CosTransactions_StatusActive:

The transaction is being executed.

ex_CosTransactions_StatusMarkedRollback:

The transaction is marked rollback.

ex_CosTransactions_StatusPrepared:

The transaction is prepared.

ex_CosTransactions_StatusCommited:

The transaction has been committed.

ex_CosTransactions_StatusRolledBack:

The transaction has been rolled back.

ex_CosTransactions_StatusUnknown:

The transaction status is unknown.

ex_CosTransactions_StatusNoTransaction:

No transaction is being executed.

ex_CosTransactions_StatusPreparing:

Prepare processing is being executed.

ex_CosTransactions_StatusCommitting:

Commit processing is being executed.

ex_CosTransactions_StatusRollingBack:

Rollback processing is being executed.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major of structure env, and detailed information is set in _id of structure env.

The values of _id and their meanings are described below:

ex_CORBA_StExcep_TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

ex_CORBA_StExcep_NO_IMPLEMENT:

The Database Linkage Service system is not started.

ex_CORBA_StExcep_COMM_FAILURE:

A communication error occurred.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

1.13.6 CosTransactions_Current_get_transaction_name() Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Current_get_transaction_name

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
CORBA_char *CosTransactions_Current_get_transaction_name(
     CORBA_Object current,
     CORBA_Environment       *env);
```

Description

This function returns a character string for identifying a transaction. Specify the object reference of the 1.13 Current Interface obtained using the

CORBA_ORB_resolve_initial_references method

(CORBA_ORB_ObjectId_TransactionCurrent), in current.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of structure env, and the character string for identifying the transaction is returned. If no transaction has been created, a NULL character is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major of structure env, and detailed information is set in _id of structure env.

ex_CORBA_StExcep_TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

ex_CORBA_StExcep_NO_IMPLEMENT:

The Database Linkage Service system is not started.

ex_CORBA_StExcep_COMM_FAILURE:

A communication error occurred.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

1.13.7 CosTransactions_Current_set_timeout() 

Name

CosTransactions_Current_set_timeout

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
void  CosTransactions_Current_set_timeout(
CORBA_Object current,
     CORBA_unsigned_long     seconds,
     CORBA_Environment       *env);
```

Description

This function sets the specified seconds as the transaction timeout observation time. If processing is not completed within that time, the transaction is rolled back.

When 0 is specified for seconds, no timeout observation is performed.

If the function is called while a transaction is being executed, the specified timeout value is not valid until the next transaction starts.

Specify the object reference of the 1.13 Current Interface obtained using the

CORBA_ORB_resolve_initial_references method

(CORBA_ORB_ObjectId_TransactionCurrent), in current.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of structure env.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major of structure env, and detailed information is set in _id of structure env.

When a system exception occurs, the following detailed information is set:

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

ex_CORBA_StExcep_INTERNAL:

A CORBA service fault occurred.

1.13.8 CosTransactions_Current_get_control() Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Current_get_control

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
CosTransactions_control CosTransactions_Current_get_control(
                         CosTransactions_Current current,
                         CORBA_Environment       *env);
```

Description

This function returns the Control object that manages transaction text.

This Control object can be used to complete transactions in server applications.

Specify in "current" the 1.13 Current Interface object reference fetched by the CORBA_ORB_resolve_initial_references method (CORBA_ORB_ObjectId_TransactionCurrent).

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of structure env, and a character string to identify the transaction is returned in the return value.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major of structure env, and detailed information is set in _id of structure env.

When a system exception occurs, the following detailed information is set:

ex_CORBA_StExcep_NO_IMPLEMENT:

The Database Linkage Service system did not start.

ex_CORBA_StExcep_COMM_FAILURE:

A communications failure occurred.

ex_CORBA_StExcep_NO_RESOURCES:

There are insufficient resources.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

1.13.9 CosTransactions_Current_suspend() Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Current_suspend

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
CosTransactions_control CosTransactions_Current_suspend(
                                CosTransactions_Current current,
                                CORBA_Environment       *env);
```

Description

This function removes the association between a transaction generated by an application and a thread, and stops the transaction. The function returns the Control object that manages the transaction context that manages the transaction.

When this function terminates normally, the transaction remains valid until the resume function is executed.

Specify in "current" the 1.13 Current Interface object reference fetched by the CORBA_ORB_resolve_initial_references method (CORBA_ORB_ObjectId_TransactionCurrent).

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of structure env, and a character string to identify the transaction is returned in the return value. A NULL character is returned if no transaction was generated.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major of structure env, and detailed information is set in _id of structure env.

When a system exception occurs, the following detailed information is set:

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

ex_CORBA_StExcep_UNKNOWN:

An initialization error occurred in the CORBA Service.

ex_CORBA_StExcep_INTERNAL

CORBA Service error.

1.13.10 CosTransactions_Current_resume() Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Current_resume

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
void CosTransactions_Current_resume(
                        CosTransactions_Current     current,
                        CosTransactions_control     control,
                        CORBA_Environment       *env);
```

Description

This function associates a transaction and a current thread.

Specify in "control" the Control object object reference fetched by the get_control method. Otherwise use the Control object returned by the suspend function. The transaction is associated with the current thread in the Control object. If a null object is specified in control, the association between the transaction and the current thread is removed.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of structure env.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major of structure env, and detailed information is set in _id of structure env.

When a user exception occurs, the following detailed information is set:

ex_cosTransactions_InvalidControl

An object other than a Control object was specified.

When a system exception occurs, the following detailed information is set:

ex_CORBA_StExcep_INTERNAL

CORBA Service error.

1.14 Transaction Initialization Interface Windows32/64 Solaris32 Linux32/64

This section details the following function:

OTS_init

1.14.1 OTS_init Windows32/64 Solaris32 Linux32/64

Name

OTS_init

Synopsis

```
#include  "OTS.h"
void  OTS_init(
     CORBA_Object             ots;
     CORBA_Object             obj,
     CORBA_char              *impl,
     CORBA_Environment       *env);
```

Description

This function connects a server application to the Database Linkage Service. Specify the Implementation Repository ID of a server application in impl.

Specify the object reference of the Database Linkage Service interface obtained using the

CORBA_ORB_resolve_initial_references method

(CORBA_ORB_ObjectId_TransactionServerInit), in ots.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of structure env.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major of structure env, and detailed information is set in _id of structure env.

ex_OTS_Bad_Description:

A description in the resource definition file is invalid.

ex_OTS_No_DefFile:

The resource definition file cannot be found. Or, the resource definition file name is not specified to be Implementation Repository.

ex_OTS_Rm_Error:

A temporary error occurred during an attempt to open a database.

ex_OTS_Permission_Denied:

Permission to use the resource definition file is denied.

ex_OTS_IOError:

An I/O error occurred during an attempt to read the resource definition file.

Note

Do not call this function more than once in one process.

1.15 Transaction Termination Interface Windows32/64 Solaris32 Linux32/64

This section details the following function:

OTS_term

1.15.1 OTS_term Windows32/64 Solaris32 Linux32/64

Name

OTS_term

Synopsis

```
#include  "OTS.h"
void  OTS_term(
     CORBA_Object             ots;
     CORBA_Environment       *env);
```

Description

This function disconnects a server application and the Database Linkage Service. Specify the object reference of the Database Linkage Service interface obtained using the

CORBA_ORB_resolve_initial_reference method

(CORBA_ORB_ObjectId_TransactionServerInit), in ots.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of structure env.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major of structure env, and detailed information is set in _id of structure env.

Note

Do not call this function more than once in one process.

1.16 Load Balance Function Interface Windows32 Solaris32 Linux32

This is not supported in Insterstage Application Server for Linux (64 bit).

This section describes the interfaces provided by the load balance function.

Note

The load balance function can only be used in the Interstage Application Server Enterprise Edition.

1.16.1 Load Balance Option Interface Windows32 Solaris32 Linux32

This section details the following functions:

ISOD_LBO_create_LBG()

ISOD_LBO_resolve_LBG()

ISOD_LBO_delete_LBG()

ISOD_LBO_list_LBG()

ISOD_LBO_notify_down()

ISOD_LBO_notify_recover()

1.16.1.1 ISOD_LBO_create_LBG() Windows32 Solaris32 Linux32

Name

ISOD_LBO_create_LBG

Synopsis

```
#include <orb.h>
#include <OM_LBO.h>
ISOD_LBG  ISOD_LBO_create_LBG(
     ISOD_LBO  lbo,
     CosNaming_NamingContext  nc,
     CosNaming_Name  *n,
     ISOD_LBO_LoadBalanceType  loadbalancetype,
     CORBA_Object  defaultobjectref,
     CORBA_Environment  *env );
```

Description

This function creates a load balance object group, and registers this group in the Naming Service. The group is registered under the name specified by n, in the naming context specified by nc. If the load balance object group is generated successfully, then the object reference of the created load balance object group is returned.

Since this function acquires area to store the object reference, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

lbo

The object reference of the load balance object.

nc

The naming context object.

n

The created load balance object group name.

loadbalancetype

Specify ISOD_LBO_roundrobin

defaultobjref

if the load balancing function is not in operation, specify the object reference to be returned by the Naming Service (default object reference).

If you do not specify defaultobjectref and the load balancing function is not in operation, then no object reference is returned by the Naming Service. This object reference cannot be registered or deleted by ISOD_LBG_bind() or ISOD_LBG_unbind().

If defaultobjectref is specified, then it can be altered using the ISOD_LBG_rebind_default() operation, after the load balance object group has been created.

env

A structure that may contain exception information.

Return Values

If processing terminates normally, then CORBA_NO_EXCEPTION is set in _major in the env structure. In the event of abnormal termination, CORBA_USER_EXECUTION is set, and detailed information is set in _id in the env structure. Values for _id and their meanings are as follows:

IDL:ISOD/LBO/NotFound:1.0

Naming context specified by n does not found.

IDL:ISOD/LBO/CannotProceed:1.0

Naming context specified by nc does not exist.

IDL:ISOD/LBO/InvalidName:1.0

Error in name specification.

IDL:ISOD/LBO/AlreadyExist:1.0

Binding for specified name already exists.

IDL:ISOD/LBO/InvalidType:1.0

Error in load balance type specification.

IDL:ISOD/LBO/BadObject:1.0

Invalid Object specified in defaultobjectref.

IDL:ISOD/LBO/OperationBusy:1.0

Request for multi-processing has exceeded maximum. Retry later.

1.16.1.2 ISOD_LBO_resolve_LBG() Windows32 Solaris32 Linux32

Name

ISOD_LBO_resolve_LBG

Synopsis

```
#include <orb.h>
#include <OM_LBO.h>
ISOD_LBG  ISOD_LBO_resolve_LBG(
     CORBA_Object obj,
     CosNaming_NamingContext nc,
     CosNaming_Name *n,
     CORBA_Environment *env );
```

Description

This function searches the Naming Service for the load balance object group (specified at n) under the naming context specified at nc.

If load balance object group is successfully searched, an object reference of load balance object group, which is searched as a return value, is returned.

Since this function acquires area to store the object reference, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

lbo

The object reference of the load balance object.

nc

The naming context object.

n

The searched load balance object group name.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_USER_EXCEPTION is set, and detailed information is set in _id of the env structure.

The values of _id and their meanings are described below:

IDL:ISOD/LBO/NotFound:1.0

The name specified in n was not found.

IDL:ISOD/LBO/CannotProceed:1.0

Naming context specified in nc does not exist.

IDL:ISOD/LBO/InvalidName:1.0

Invalid name specification.

IDL:ISOD/LBO/OperationBusy:1.0

Request for multi-processing has reached the maximum limit. Retry later.

1.16.1.3 ISOD_LBO_delete_LBG() Windows32 Solaris32 Linux32

Name

ISOD_LBO_delete_LBG

Synopsis

```
#include <orb.h>
#include <OM_LBO.h>
CORBA_Object   ISOD_LBO_delete_LBG(
     CORBA_Object obj,
     CosNaming_NamingContext nc,
     CosNaming_Name *n,
     CORBA_Environment *env );
```

Description

This function deletes the load balance object group from the Naming Service.

The load balance object group specified at n under the naming context specified at nc is deleted, and the default object reference (if it is set) is returned. If an object other than the default object is registered in the load balance object group to be deleted, exception is returned.

Since this function acquires area to store the object reference, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

lbo

The object reference of the load balance object.

nc

The naming context object.

n

The deleted load balance object group name.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_USER_EXCEPTION is set, and detailed information is set to _id of the env structure.

The values of _id and their meanings are described below:

IDL:ISOD/LBO/NotFound:1.0

The name specified in n was not found.

IDL:ISOD/LBO/CannotProceed:1.0

Naming context specified in nc does not exist.

IDL:ISOD/LBO/InvalidName:1.0

Invalid name specification.

IDL:ISOD/LBO/NotEmpty:1.0

More than one object is registered in the object group.

IDL:ISOD/LBO/OperationBusy:1.0

Request for multi-processing has reached the maximum limit. Retry later.

1.16.1.4 ISOD_LBO_list_LBG() Windows32 Solaris32 Linux32

Name

ISOD_LBO_list_LBG

Synopsis

```
#include <orb.h>
#include <OM_LBO.h>
void  ISOD_LBO_list_LBG(
     CORBA_Object obj,
     CosNaming_NamingContext nc,
     ISOD_LBO_LBGList **LBGlist,
     CORBA_Environment *env );
```

Description

This function returns a list of the names of the load balance object groups under the naming context specified at nc, and the sequence of the structure (BindingObjectGroup) load balance type.

The maximum limit of the load balance object groups that can be retrieved by this function is 128.

Since this function acquires area to store the list of the load balance object groups, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

lbo

The object reference of the load balance object.

nc

The naming context object.

LBGlist

The set area of the list of the load balance object groups.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set to _major of the env structure.

For abnormal termination, CORBA_USER_EXCEPTION is set.

1.16.1.5 ISOD_LBO_notify_down() Windows32 Solaris32 Linux32

Name

ISOD_LBO_notify_down

Synopsis

```
#include <orb.h>
#include <OM_LBO.h>
void  ISOD_LBO_notify_down(
     CORBA_Object  obj,
     CORBA_char  *HostName,
     CORBA_Environment  *env );
```

Description

This function notifies the load balance function that the server is down.

When this function is called, it stops the return of the object reference from the server whose address is specified at HostName.

Parameters

lbo

The object reference of the load balance object.

HostName

The host name or IP address of the server that is down.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set to _major of the env structure.

For abnormal termination, CORBA_USER_EXCEPTION is set, and detailed information is set to _id of the env structure.

The values of _id and their meanings are described below:

IDL:ISOD/LBO/InvalidArgument:1.0

Hostname specification is invalid.

IDL:ISOD/LBO/CannotProceed2:1.0

An abnormal error has occurred in the DB process of the load balance function.

1.16.1.6 ISOD_LBO_notify_recover() Windows32 Solaris32 Linux32

Name

ISOD_LBO_notify_recover

Synopsis

```
#include <orb.h>
#include <OM_LBO.h>
void  ISOD_LBO_notify_recover(
     CORBA_Object  obj,
     CORBA_char   *HostName,
     CORBA_Environment  *env );
```

Description

This function notifies the load balance function that the server has been restored.

When this function is called, it starts the return of the object reference from the server whose address is specified at HostName.

Parameters

lbo

The object reference of the load balance object.

HostName

The host name or IP address of the restored server.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set to _major of the env structure.

For abnormal termination, CORBA_USER_EXCEPTION is set, and detailed information is set to _id of the env structure.

The values of _id and their meanings are described below:

IDL:ISOD/LBO/InvalidArgument:1.0

Hostname specification is invalid.

IDL:ISOD/LBO/CannotProceed2:1.0

An abnormal error has occurred in the DB process of the load balance function.

1.16.2 Load Balance Object Group Interface Windows32 Solaris32 Linux32

This section details the following functions:

ISOD_LBG_bind()

ISOD_LBG_unbind()

ISOD_LBG_rebind_default()

ISOD_LBG_list()

1.16.2.1 ISOD_LBG_bind() Windows32 Solaris32 Linux32

Name

ISOD_LBG_bind

Synopsis

```
#include <orb.h>
#include <OM_LBO.h>
void  ISOD_LBG_bind(
     ISOD_LBG  lbg,
```

```
    CORBA_Object  objectref,
    CORBA_Environment  *env );
```

Description

This operation registers the object for load balancing specified by objectref in the load balance object group specified by lbg.

Parameters

lbg

The object reference of the load balance object group.

objectref

The object reference registered to the load balance object group.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_USER_EXCEPTION is set, and detailed information is set in _id in the env structure.

Values for _id and their meanings are as follows:

IDL:ISOD/LBG/AlreadyBound:1.0

The specified object contains the same information as an object that has already been registered.

IDL:ISOD/LBG/CannotProceed2:1.0

Either an error has occurred in the DB process of the load balancing function, or the objectref specification is incorrect.

IDL:ISOD/LBG/BadObject:1.0

Invalid object specified in objectref.

1.16.2.2 ISOD_LBG_unbind() Windows32 Solaris32 Linux32

Name

ISOD_LBG_unbind

Synopsis

```
#include <orb.h>
#include <OM_LBO.h>
void  ISOD_LBG_unbind(
     ISOD_LBG  lbg,
     CORBA_Object  objectref,
     CORBA_Environment  *env );
```

Description

This operation deletes the object for load balance specified by objectref, from the load balance object group specified at lbg.

Parameters

lbg

The object reference of the load balance object group.

objectref

The object reference deleted from the load balance object group.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_USER_EXECUTION is set, and detailed information is set in _id in the env structure.

Values for _id and their meanings are as follows:

IDL:ISOD/LBG/NotFound:1.0

Object specified by objectref not found.

IDL:ISOD/LBG/CannotProceed2:1.0

Either an error has occurred in the DB process of the load balance function or the objectref specification is incorrect.

IDL:ISOD/LBG/BadObject:1.0

Invalid object specified in objectref.

1.16.2.3 ISOD_LBG_rebind_default() Windows32 Solaris32 Linux32

Name

ISOD_LBG_rebind_default

Synopsis

```
#include <orb.h>
#include <OM_LBO.h>
CORBA_Object ISOD_LBG_rebind_default(
     ISOD_LBG lbg,
     CORBA_Object  objectref,
     CORBA_Environment  *env );
```

Description

This operation changes the default object in the load balance object group specified by lbg, to the object specified by objectref.

Since this function acquires area to store the object reference, use CORBA_Object_release() to release the area as soon as it is no longer needed.

Parameters

lbg

The object reference of the load balance object group.

objectref

The object reference registered as default object.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_USER_EXECUTION is set, and detailed information is set in _id in the env structure.

Values for _id and their meanings are as follows:

IDL:ISOD/LBG/CannotProceed:1.0

Default object is not specified.

IDL:ISOD/LBG/CannotProceed2:1.0

Either an error has occurred in the DB process of the load balance function or the objectref specification is incorrect.

IDL:ISOD/LBG/BadObject:1.0

Invalid object specified in objectref.

IDL:ISOD/LBG/OperationBusy:1.0

Request for multi-processing has exceeded maximum. Retry later.

1.16.2.4 ISOD_LBG_list() Windows32 Solaris32 Linux32

Name

ISOD_LBG_list

Synopsis

```
#include <orb.h>
#include <OM_LBO.h>
void ISOD_LBG_list(
     CORBA_Object  lbg,
     ISOD_LBG_ObjectList  **objectref,
     CORBA_Environment  *env );
```

Description

This function returns a list of object references for objects already registered in the load balance object group specified at lbg.

Since this function acquires area to store the list of the object, use CORBA_free() to release the area as soon as it is no longer needed.

Parameters

lbg

The object reference of the load balance object group.

objectref

The set area of the list of the object.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set to _major of the env structure.

For abnormal termination, CORBA_USER_EXCEPTION is set, and detailed information is set to _id of the env structure.

The values of _id and their meanings are described below:

IDL:ISOD/LBG/CannotProceed2:1.0

An abnormal error has occurred in the DB process of the load balance function.

1.17 Event Service Interface

It is possible to communicate with the Event Service and notification service event channels.

To communicate with the event channel, include "EventService.h".

To communicate with the notification service event channel, include "NotificationService.h".

1.17.1 CosEventComm Interface

This section details the functions associated with the CosEventComm Interface.

1.17.1.1 CosEventComm_PushConsumer_push()

Name

CosEventComm_PushConsumer_push

Synopsis

```
#include  <EventService.h>
void  CosEventComm_PushConsumer_push(
     CosEventComm_PushConsumer  obj,
     CORBA_any  *data,
     CORBA_Environment  *env );
```

Description

Transmits the event data specified by data to the consumer.

Parameters

obj

The object reference returned by CosEventChannelAdmin_SupplierAdmin_obtain_push_consumer.

data

The event data transmitted to the consumer

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the structure env. For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is specified.

ex_CosEventComm_Disconnected

Not connected to event channel.

If automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

1.17.1.2 CosEventComm_PushConsumer_disconnect_push_consumer()

Name

CosEventComm_PushConsumer_disconnect_push_consumer

Synopsis

```
#include  <EventService.h>
void  CosEventComm_PushConsumer_disconnect_push_consumer(
```

```
    CosEventComm_PushConsumer  obj,
    CORBA_Environment  *env );
```

Description

Declares termination of event communication by supplier.

Parameters

obj

The object reference returned by CosEventChannelAdmin_SupplierAdmin_obtain_push_consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.17.1.3 CosEventComm_PushSupplier_disconnect_push_supplier()

Name

CosEventComm_PushSupplier_disconnect_push_supplier

Synopsis

```
#include  <EventService.h>
void  CosEventComm_PushSupplier_disconnect_push_supplier(
     CosEventComm_PushSupplier  obj,
     CORBA_Environment  *env );
```

Description

Declares termination of event communication by consumer.

Parameters

obj

The object reference in CosEventChannelAdmin_SupplierAdmin_obtain_push_supplier.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.17.1.4 CosEventComm_PullSupplier_pull()

Name

CosEventComm_PullSupplier_pull

Synopsis

```
#include  <EventService.h>
CORBA_any  *CosEventComm_PullSupplier_pull(
     CosEventComm_PullSupplier  obj,
     CORBA_Environment  *env );
```

Description

Requests event data from supplier. This operation is blocked until event data can be fetched, or until an exception is generated. If the event data cannot be fetched and an immediate response is required, then please use CosEventComm_PullSupplier_try_pull.

When the area secured for any type event data is no longer needed, it must be released using CORBA_free().

Parameters

obj

The object reference returned by CosEventChannelAdmin_ConsumerAdmin_obtain_pull_supplier.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure, and event data from the supplier is returned. For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is specified.

ex_CosEventComm_Disconnected

Not connected to event channel.

If automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

1.17.1.5 CosEventComm_PullSupplier_try_pull()

Name

CosEventComm_PullSupplier_try_pull

Synopsis

```
#include  <EventService.h>
CORBA_any  *CosEventComm_PullSupplier_try_pull(
     CosEventComm_PullSupplier  obj,
     CORBA_boolean  *has_event,
     CORBA_Environment  *env );
```

Description

Requests event data from supplier. Instantly returned if event data cannot be fetched from supplier. If you want to block this operation until event data can be fetched, then please use CosEventComm_PullSupplier_pull.

When the area secured for any type event data is no longer needed, it must be released using CORBA_free().

Parameters

obj

The object reference returned by CosEventChannelAdmin_ConsumerAdmin_obtain_pull_supplier.

has_event (out parameter)

When event data has been fetched, CORBA_TRUE is set.

When event data has not been fetched, then CORBA_FALSE is set

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. When event data has been fetched, CORBA_TRUE is set in has_event and event data from the supplier is returned. If event data has not been fetched, then CORBA_FALSE is set in has_event. In this case, an area is secured for the any type event data, and when the area is no longer needed, it must be released using CORBA_free().

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is specified.

ex_CosEventComm_Disconnected

Not connected to event channel.

If automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

Notes

The cancellation need not be notified using ES_ChannelUtil_local_rollback() when CORBA_FALSE is set to has_event when a local transaction is operated.

Notifies the channel of completing using ES_ChannelUtil_local_commit().

Windows32/64 Solaris32 Linux32/64

The cancellation need not be notified using CosTransactions_Current_rollback() when CORBA_FALSE is set to has_event when a global transaction is operated.

Notifies the channel of completing using CosTransactions_Current_commit().

1.17.1.6 CosEventComm_PullSupplier_disconnect_pull_supplier()

Name

CosEventComm_PullSupplier_disconnect_pull_supplier

Synopsis

```
#include  <EventService.h>
void  CosEventComm_PullSupplier_disconnect_pull_supplier(
     CosEventComm_PullSupplier  obj,
     CORBA_Environment  *env );
```

Description

Declares termination of event communication by consumer.

Parameters

obj

The object reference returned by CosEventChannelAdmin_ConsumerAdmin_obtain_pull_supplier.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.17.1.7 CosEventComm_PullConsumer_disconnect_pull_consumer()

Name

CosEventComm_PullConsumer_disconnect_pull_consumer

Synopsis

```
#include  <EventService.h>
void  CosEventComm_PullConsumer_disconnect_pull_consumer(
     CosEventComm_PullConsumer  obj,
     CORBA_Environment  *env );
```

Description

Declares termination of event communication by consumer.

Parameters

obj

The object reference returned by CosEventChannelAdmin_SupplierAdmin_obtain_pull_consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.17.2 CosEventChannelAdmin Interface

This section details the functions associated with the CosEventChannelAdmin Interface

1.17.2.1 CosEventChannelAdmin_EventChannel_for_consumers()

Name

CosEventChannelAdmin_EventChannel_for_consumers

Synopsis

```
#include  <EventService.h>
CosEventChannelAdmin_ConsumerAdmin
CosEventChannelAdmin_EventChannel_for_consumers(
     CosEventChannelAdmin_EventChannel  obj,
     CORBA_Environment  *env );
```

Description

Acquires object reference of event channel from consumer.

Since this function acquires area to store object references, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

obj

The object reference of the connected event channel.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.17.2.2 CosEventChannelAdmin_EventChannel_for_suppliers()

Name

CosEventChannelAdmin_EventChannel_for_suppliers

Synopsis

```
#include  <EventService.h>
CosEventChannelAdmin_SupplierAdmin
CosEventChannelAdmin_EventChannel_for_suppliers(
     CosEventChannelAdmin_EventChannel   obj,
     CORBA_Environment                  *env );
```

Description

Acquires object reference of event channel from supplier.

Since this method acquires area to store object references, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

obj

The object reference of the connected event channel.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.17.2.3 CosEventChannelAdmin_EventChannel_destroy()

Name

CosEventChannelAdmin_EventChannel_destroy

Synopsis

```
#include  <EventService.h>
void  CosEventChannelAdmin_EventChannel_destroy(
     CosEventChannelAdmin_EventChannel  obj,
     CORBA_Environment  *env );
```

Description

Destroys the event channel specified by obj.

Parameters

obj

The object reference of the destroyed event channel.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.17.2.4 CosEventChannelAdmin_ConsumerAdmin_obtain_push_supplier()

Name

CosEventChannelAdmin_ConsumerAdmin_obtain_push_supplier

Synopsis

```
#include  <EventService.h>
CosEventChannelAdmin_ProxyPushSupplier
CosEventChannelAdmin_ConsumerAdmin_obtain_push_supplier(
     CosEventChannelAdmin_ConsumerAdmin   obj,
     CORBA_Environment                   *env );
```

Description

Acquires object reference for event channel, in order to connect Push model consumer to event channel.

Since this method acquires area to store object references, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

obj

The object reference returned by CosEventChannelAdmin_EventChannel_for_consumers.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.17.2.5 CosEventChannelAdmin_consumerAdmin_obtain_pull_supplier()

Name

CosEventChannelAdmin_ConsumerAdmin_obtain_pull_supplier

Synopsis

```
#include  <EventService.h>
CosEventChannelAdmin_ProxyPullSupplier
CosEventChannelAdmin_ConsumerAdmin_obtain_pull_supplier(
     CosEventChannelAdmin_ConsumerAdmin   obj,
     CORBA_Environment                   *env );
```

Description

Acquires object reference for event channel, in order to connect Pull model consumer to event channel.

Since this method acquires area to store object references, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

obj

The object reference returned by CosEventChannelAdmin_EventChannel_for_consumers.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.17.2.6 CosEventChannelAdmin_SupplierAdmin_obtain_push_consumer()

Name

CosEventChannelAdmin_SupplierAdmin_obtain_push_consumer

Synopsis

```
#include  <EventService.h>
CosEventChannelAdmin_ProxyPushConsumer
CosEventChannelAdmin_SupplierAdmin_obtain_push_consumer(
     CosEventChannelAdmin_SupplierAdmin   obj,
     CORBA_Environment                   *env );
```

Description

Acquires object reference for event channel, in order to connect Push model supplier to event channel.

Since this method acquires area to store object references, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

obj

The object reference returned by CosEventChannelAdmin_EventChannel_for_suppliers.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.17.2.7 CosEventChannelAdmin_SupplierAdmin_obtain_pull_consumer()

Name

CosEventChannelAdmin_SupplierAdmin_obtain_pull_consumer

Synopsis

```
#include  <EventService.h>
CosEventChannelAdmin_ProxyPullConsumer
CosEventChannelAdmin_SupplierAdmin_obtain_pull_consumer(
     CosEventChannelAdmin_SupplierAdmin   obj,
     CORBA_Environment                   *env );
```

Description

Acquires object reference for event channel, in order to connect Pull model supplier to event channel.

Since this method acquires area to store object references, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

obj

The object reference returned by CosEventChannelAdmin_EventChannel_for_suppliers.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.17.2.8 CosEventChannelAdmin_ProxyPushConsumer_connect_push_supplier()

Name

CosEventChannelAdmin_ProxyPushConsumer_connect_push_supplier

Synopsis

```
#include  <EventService.h>
void  CosEventChannelAdmin_ProxyPushConsumer_connect_push_supplier(
     CosEventChannelAdmin_ProxyPushConsumer   obj,
     CosEventComm_PushSupplier                push_supplier,
     CORBA_Environment                       *env );
```

Description

Connects Push model supplier to event channel.

Parameters

obj

The object reference returned by CosEventChannelAdmin_SupplierAdmin_obtain_push_consumer.

push_supplier

The object reference of the supplier itself.

If a disconnection notice is not required when the event channel is terminated, set CORBA_OBJECT_NIL in push_supplier.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is specified.

ex_CosEventChannelAdmin_AlreadyConnected

Event channel already connected.

Note

To reconnect to an event channel, start again from CosEventChannelAdmin_SupplierAdmin_obtain_push_consumer.

1.17.2.9 CosEventChannelAdmin_ProxyPullSupplier_connect_pull_consumer()

Name

CosEventChannelAdmin_ProxyPullSupplier_connect_pull_consumer

Synopsis

```
#include  <EventService.h>
void  CosEventChannelAdmin_ProxyPullSupplier_connect_pull_consumer(
     CosEventChannelAdmin_ProxyPullSupplier   obj,
     CosEventComm_PullConsumer                pull_consumer,
     CORBA_Environment                       *env );
```

Description

Connects Pull model consumer to event channel.

Parameters

obj

The object reference returned by CosEventChannelAdmin_ConsumerAdmin_obtain_pull_supplier.

pull_consumer

The object reference of the consumer itself.

If a disconnection notice is not required when the event channel is terminated, set CORBA_OBJECT_NIL in pull_consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is specified.

ex_CosEventChannelAdmin_AlreadyConnected

Event channel already connected.

Note

To reconnect to an event channel, start again from CosEventChannelAdmin_ConsumerAdmin _obtain_pull_supplier.

1.17.2.10 CosEventChannelAdmin_ProxyPullConsumer_connect_pull_supplier()

Name

CosEventChannelAdmin_ProxyPullConsumer_connect_pull_supplier

Synopsis

```
#include  <EventService.h>
void  CosEventChannelAdmin_ProxyPullConsumer_connect_pull_supplier(
     CosEventChannelAdmin_ProxyPullConsumer   obj,
     CosEventComm_PullSupplier                pull_supplier,
     CORBA_Environment                       *env );
```

Description

Connects Pull model supplier to event channel.

Parameters

obj

The object reference returned by CosEventChannelAdmin_SupplierAdmin_obtain_pull_consumer.

pull_supplier

The object reference of the supplier itself.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is specified.

ex_CosEventChannelAdmin_AlreadyConnected

Event channel already connected.

ex_CosEventChannelAdmin_TypeError

The specified object type is incorrect.

Note

To reconnect to an event channel, start again from CosEventChannelAdmin_SupplierAdmin_obtain_pull_consumer.

1.17.2.11 CosEventChannelAdmin_ProxyPushSupplier_connect_push_consumer()

Name

CosEventChannelAdmin_ProxyPushSupplier_connect_push_consumer

Synopsis

```
#include  <EventService.h>
void  CosEventChannelAdmin_ProxyPushSupplier_connect_push_consumer(
     CosEventChannelAdmin_ProxyPushSupplier   obj,
     CosEventComm_PushConsumer                push_consumer,
     CORBA_Environment                       *env );
```

Description

Connects Push model consumer to event channel.

Parameters

obj

The object reference returned by CosEventChannelAdmin_ConsumerAdmin_obtain_push_supplier.

push_consumer

The object reference of the consumer itself.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure. For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is specified.

ex_CosEventChannelAdmin_AlreadyConnected

Event channel already connected.

ex_CosEventChannelAdmin_TypeError

The specified object type is incorrect.

Note

To reconnect to an event channel, start again from CosEventChannelAdmin_ConsumerAdmin_obtain_push_supplier.

1.17.2.12 Interfaces Usable when Inherited

The following interfaces can be used by inheritance. For details refer to 1.17.1 CosEventComm Interface.

- CosEventChannelAdmin_ProxyPushConsumer_push
- CosEventChannelAdmin_ProxyPushConsumer_disconnect_push_consumer
- CosEventChannelAdmin_ProxyPushSupplier_disconnect_push_supplier
- CosEventChannelAdmin_ProxyPullSupplier_pull
- CosEventChannelAdmin_ProxyPullSupplier_try_pull
- CosEventChannelAdmin_ProxyPullSupplier_disconnect_pull_supplier
- CosEventChannelAdmin_ProxyPullConsumer_disconnect_pull_consumer

1.17.3 EventFactory Interface

This section details the functions associated with the EventFactory Interface.

1.17.3.1 EventFactory_create()

Name

EventFactory_create

Synopsis

```
#include  <EventService.h>
CosEventChannelAdmin_EventChannel EventFactory_create(
     EventFactory                      obj,
     CORBA_string                    key,
     EventFactory_Option         data,
     CORBA_Environment        *env );
typedef struct {
     CORBA_long                      max_queuing;
     CORBA_long                      life_time;
     EventFactory_Model         model;
} EventFactory_Option;
```

Description

This operation generates an event channel and returns the object reference of the generated event channel.

Since this method acquires area to store object references, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

obj

The object reference is acquired by specifying the "EventFactory_ObjectId_Factory" in the identifier parameter of the CORBA_ORB_resolve_initial_refernces().

key

The common keyword for consumer and supplier (maximum 64 characters).

When the common keyword is specified, an object reference indicating the same event channel is returned.

data

Specify the EventFactory_Option structure.

Specify the values in the following table for each member of the EventFactory_Option structure.

Table 1.1 EventFactory_Option Members

Member	Set value
max_queuing	When you set EventFactory_ES_DEFAULT_VALUE, it uses the "maximum value of event data number which can be stored in the event channel" set in the event service configuration information setting.
life_time	Data holding time (seconds). When you set EventFactory_ES_DEFAULT_VALUE, it uses the value of "existing time (sec.) of the stored event data" set in the event service configuration information setting. If 0 is set, then timeout monitoring is not implemented.
Model	Specifies the following connection models: EventFactory_ModelAny Determine when connected EventFactory_ModelPush Push model EventFactory_ModelPull Pull model EventFactory_ModelMixed Mixed model

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure and the object reference of the generated event channel, is returned. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.17.3.2 EventFactory_create_channel()

Name

EventFactory_create_channel

Synopsis

```
#include  <EventService.h>
CosEventChannelAdmin_EventChannel
     EventFactory_create_channel(
            EventFactory                obj,
            CORBA_string                key,
            EventFactory_Option         *data,
            EventFactory_EventProperty  *property,
            CORBA_boolean               *create,
            CORBA_Environment           *env );

typedef struct {
      CORBA_long                        max_queuing;
      CORBA_long                        life_time;
      EventFactory_Model                model;
} EventFactory_Option;

typedef struct EventFactory_Property{
      CORBA_string                      name;
      CORBA_any                         value;
} EventFactory_Property;

typedef struct {
      CORBA_unsigned_long               _maximum;
      CORBA_unsigned_long               _length;
      struct EventFactory_Property      *_buffer;
} EventFactory_EventProperty;
```

Description

This operation generates an event channel and returns the object reference of the generated event channel. This method is used to change the host name and port number for the generated Channel.

Since this method acquires an area to store object references, use CORBA_Object_release () to release the area as soon as it is no longer needed.

Parameters

obj

The object reference is acquired by specifying the "EventFactory_ObjectId_Factory" in the identifier parameter of the CORBA_ORB_resolve_initial_refernces().

key

The common keyword for consumer and supplier (maximum 64 characters).

When the common keyword is specified, an object reference indicating the same event channel is returned.

data

Specify the EventFactory_Option structure.

Specify the values in the following table for each member of the EventFactory_Option structure.

If the Event Channel has already been generated, the value for this parameter is ignored.

Table 1.2 EventFactory_Option Members

Member	Set value
max_queuing	When you set EventFactory_ES_DEFAULT_VALUE, it uses the "maximum value of event data number which can be stored in the event channel" set in the event service configuration information setting.

Member	Set value
life_time	Data holding time (seconds). When you set EventFactory_ES_DEFAULT_VALUE, it uses the value of "existing time (sec.) of the stored event data" set in the event service configuration information setting. If 0 is set, then timeout monitoring is not implemented.
Model	Specifies the following connection models: EventFactory_ModelAny Determine when connected EventFactory_ModelPush Push model EventFactory_ModelPull Pull model EventFactory_ModelMixed Mixed model

property

Specifies the value shown in the table below. If an invalid name is specified, the corresponding record is invalid. If either of HostName and PortNumber is set, the specified record is invalid.

If the Event Channel has already been generated, the value for this parameter is ignored.

Member (name)	Data type (value)	Set value
HostName	string	Specifies the host name or the IP address (maximum 64 characters),.
PortNumber	unsigned short	Specifies the port number.

create

If the Event Channel was generated, CORBA_TRUE is set.

If the Event Channel already exists, CORBA_FALSE is set.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure and the object reference of the generated event channel, is returned. For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

Notes

This method cannot be used if the following host names are set, as the Event Channel will fail to start.

- The "Corba Host Name" Interstage operating environment definition
- "IIOP_hostname" in the config file (CORBA Service) (The host name used by the CORBA Service)

When specifying the port number, select one of the following host names specified for the CORBA Service port number:

- If SSL communication is enabled
  - The "SSL Port Number" Interstage operating environment definition
  - "UNO_IIOP_ssl_port" in the config file (CORBA Service)
- If SSL communication is disabled
  - The "Corba Port Number" Interstage operating environment definition
  - "IIOP_port" in the config file (CORBA Service)

1.17.3.3 EventFactory_get_event_channel()

Name

EventFactory_get_event_channel

Synopsis

```
#include  <EventService.h>
CosEventChannelAdmin_EventChannel
     EventFactory_get_event_channel(
         EventFactory        obj,
         CORBA_string        key,
         CORBA_Environment   *env );
```

Description

The object reference of event channel specified with key obtains.

Parameters

obj

The object reference is acquired by specifying the "EventFactory_ObjectId_Factory" in the identifier parameter of the CORBA_ORB_resolve_initial_refernces().

key

The acquired Event Channel name.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure and the object reference of the event channel specified with key, is returned. For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

In the event of a user exception, the following detailed information is set:

ex_EventFactory_ChannelNotFound

The specified event channel could not be found.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.18 Notification Service Interface

The notification service interface enables communication with the notification service event channel.

If the notification service interface is used for the event service event channel, a BAD_OPERATION system exception occurs.

Note

The notification service function can only be used in Interstage Application Server Enterprise Edition.

1.18.1 CosNotification Interface

This section details the CosNotification Interface.

Type definitions

The formats for the data types used in this module are as follows:

```
typedef CORBA_string             CosNotification_Istring;
typedef CosNotification_Istring  CosNotification_PropertyName;
typedef CORBA_any                CosNotification_PropertyValue;

typedef struct {
    CosNotification_PropertyName      name;
    CosNotification_PropertyValue     value;
} CosNotification_Property;

typedef CORBA_sequence_CosNotification_Property    CosNotification_PropertySeq;
typedef CosNotification_PropertySeq    CosNotification_OptionalHeaderFilds;
typedef CosNotification_PropertySeq    CosNotification_FilterableEventBody;
typedef CosNotification_PropertySeq    CosNotification_QoSProperties;
typedef CosNotification_PropertySeq    CosNotification_AdminProperties;

typedef struct {
    CORBA_string      domain_name;
    CORBA_string      type_name;
} CosNotification_EventType;
typedef CORBA_sequence_CosNotification_EventType   CosNotification_EventTypeSeq;

typedef struct {
    CosNotification_PropertyValue    low_val;
    CosNotification_PropertyValue    high_val;
} CosNotification_PropertyRange;

typedef struct {
    CosNotification_PropertyName     name;
    CosNotification_PropertyRange    range;
} CosNotification_NamedPropertyRange;
typedef  CORBA_sequence_CosNotification_NamedPropertyRange
               CosNotification_NamedPropertyRangeSeq;

typedef struct {
    CosNotification_QoSError_code    code;
    CosNotification_PropertyName     name;
    CosNotification_PropertyRange    available_range;
} CosNotification_PropertyError;
typedef  CORBA_sequence_CosNotification_PropertyError
               CosNotification_PropertyErrorSeq;

typedef struct {
    CosNotification_EventType    event_type;
    CORBA_string                 event_name;
} CosNotification_FixedEventHeader;

typedef struct {
    CosNotification_FixedEventHeader     fixed_event;
    CosNotification_OptionalHeaderFilds  variable_header;
} CosNotification_EventHeader;

typedef struct {
    CosNotification_EventHeader          header;
    CosNotification_FilterableEventBody  filterable_data;
    CORBA_any                            remainder_of_body;
} CosNotification_StructuredEvent;
```

1.18.2 QoSAdmin Interface

This section details the functions associated with the QoSAdmin Interface.

1.18.2.1 CosNotification_QoSAdmin_get_qos()

Name

CosNotification_QoSAdmin_get_qos

Synopsis

```
#include  <NotificationService.h>
CosNotification_QoSProperties  *
CosNotification_QoSAdmin_get_qos(
               CosNotification_QoSAdmin  obj,
               CORBA_Environment         *env );
```

Description

Fetches QoSProperties. Refer to information about the operation of the Notification Service QoS function in the Distributed Application Development Guide (CORBA Service Edition) for details of QoSProperties.

This method secures an area for QoSProperties. Use CORBA_free() to release the area when it is no longer required.

Parameters

obj

The event channel object reference.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.18.2.2 CosNotification_QoSAdmin_set_qos()

Name

CosNotification_QoSAdmin_set_qos

Synopsis

```
#include  <NotificationService.h>
 void
 CosNotification_QoSAdmin_set_qos(
               CosNotification_QoSAdmin         obj,
               CosNotification_QoSProperties    *qos,
               CORBA_Environment                *env );
```

Description

Sets QoSProperties. Refer to information on Operation of the Notification Service QoS Function in the Distributed Application Development Guide (CORBA Service Edition) for details of QoSProperties.

Parameters

obj

The event channel object reference.

qos

The set QoSProperties.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosNotification_UnsupportedQoS

There is an error in the QoSProperties item or value.

1.18.3 CosNotifyComm Interface

This section details the functions associated with the CosNotifyComm Interface.

1.18.3.1 CosNotifyComm_StructuredPushConsumer_push_structured_event()

Name

CosNotifyComm_StructuredPushConsumer_push_structured_event

Synopsis

```
#include  <NotificationService.h>
void
CosNotifyComm_StructuredPushConsumer_push_structured_event(
                CosNotifyComm_StructuredPushConsumer  obj,
                CosNotification_StructuredEvent       *data,
                CORBA_Environment                     *env );
```

Description

Sends to the consumer the StructuredEvent type event data specified in data.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer.

data

The StructuredEvent type event data sended to consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosEventComm_Disconnected

The event channel was not connected.

If automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

1.18.3.2 CosNotifyComm_StructuredPullSupplier_pull_structured_event()

Name

CosNotifyComm_StructuredPullSupplier_pull_structured_event

Synopsis

```
#include  <NotificationService.h>
CosNotification_StructuredEvent *
CosNotifyComm_StructuredPullSupplier_pull_structured_event (
              CosNotifyComm_StructuredPullSupplier  obj,
              CORBA_Environment                     *env );
```

Description

Requests StructuredEvent type event data from the supplier. Event data can be fetched or it may be blocked until an exception occurs. Use CosNotifyComm_StructuredPullSupplier_try_pull_structured_event to return immediately if event data is not fetched.

Use CORBA_free() to release the area secured for the StructuredEvent type data when the area is no longer required.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_ConsumerAdmin_obtain_notification_pull_supplier.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosEventComm_Disconnected

The event channel was not connected.

If automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

1.18.3.3 CosNotifyComm_StructuredPullSupplier_try_pull_structured_event()

Name

CosNotifyComm_StructuredPullSupplier_try_pull_structured_event

Synopsis

```
#include  <NotificationService.h>
CosNotification_StructuredEvent *
CosNotifyComm_StructuredPullSupplier_try_pull_structured_event (
               CosNotifyComm_StructuredPullSupplier  obj,
               CORBA_boolean                         *has_event,
               CORBA_Environment                     *env );
```

Description

Requests StructuredEvent type event data from the supplier. Immediately returns if event data is not fetched. Use CosNotifyComm_StructuredPullSupplier_pull_structured_event to block until event data is fetched.

Use CORBA_free() to release the area secured for the StructuredEvent type data when the area is no longer required.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_ConsumerAdmin_obtain_notification_pull_supplier.

has_event (out parameter)

When event data is fetched, CORBA_TRUE is set.

When event data is not fetched, CORBA_FALSE is set.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure. When event data is fetched, CORBA_TRUE is set in has_event, and the event data from the supplier is returned. When event data is not fetched, CORBA_FALSE is set in has_event. In this case, because an area is secured for the StructuredEvent type event data, use CORBA_free() to release the area.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosEventComm_Disconnected

The event channel was not connected.

If automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

Notes

The cancellation need not be notified using ES_ChannelUtil_local_rollback() when CORBA_FALSE is set to has_event when a local transaction is operated.

Notifies the channel of completing using ES_ChannelUtil_local_commit().

Windows32/64 Solaris32 Linux32/64

The cancellation need not be notified using CosTransactions_Current_rollback() when CORBA_FALSE is set to has_event when a global transaction is operated.

Notifies the channel of completing using CosTransactions_Current_commit().

1.18.3.4 CosNotifyComm_StructuredPushConsumer_disconnect_structured_push_consumer()

Name

CosNotifyComm_StructuredPushConsumer_disconnect_structured_push_consumer

Synopsis

```
#include  <NotificationService.h>
void
CosNotifyComm_StructuredPushConsumer_disconnect_structured_push_consumer (
               CosNotifyChannelAdmin_StrucuturedProxyPushConsumer  obj,
               CORBA_Environment                                   *env );
```

Description

Declares the end of event transmission from the supplier.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.18.3.5 CosNotifyComm_StructuredPullSupplier_disconnect_structured_pull_supplier()

Name

CosNotifyComm_StructuredPullSupplier_disconnect_structured_pull_supplier

Synopsis

```
#include  <NotificationService.h>
void
CosNotifyComm_StructuredPullSupplier_disconnect_structured_pull_supplier (
```

```
                CosNotifyChannelAdmin_StrucuturedProxyPullSupplier  obj,
                CORBA_Environment                                   *env );
```

Description

Declares the end of event transmission from the consumer.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_pull_supplier.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and detailed information is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.18.3.6 Inherited Interfaces

The following interfaces can be inherited and used.

(1) CosNotifyComm_PushConsumer_push

(2) CosNotifyComm_PushConsumer_disconnect_push_consumer

(3) CosNotifyComm_PullSupplier_pull

(4) CosNotifyComm_PullSupplier_try_pull

(5) CosNotifyComm_PullSupplier_disconnect_pull_supplier

For details of these interfaces, refer to 1.17.1 CosEventComm Interface.

1.18.4 CosNotifyChannelAdmin Interface

This section details the functions associated with the CosNotifyChannelAdmin Interface.

1.18.4.1 CosNotifyChannelAdmin_EventChannel__get_default_consumer_admin()

Name

CosNotifyChannelAdmin_EventChannel__get_default_consumer_admin

Synopsis

```
#include  <NotificationService.h>
CosNotifyChannelAdmin_ConsumerAdmin
CosNotifyChannelAdmin_EventChannel__get_default_consumer_admin(
                CosNotifyChannelAdmin_EventChannel          obj,
                CORBA_Environment                           *env );
```

Description

Returns the object reference of the ConsumerAdmin object that is standard for the event channel.

This method secures an area for holding the object references. Use CORBA_Object_release() to release the area when it is no longer required.

Parameters

obj

The object reference for the event channel connected.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.18.4.2 CosNotifyChannelAdmin_EventChannel__get_default_supplier_admin()

Name

CosNotifyChannelAdmin_EventChannel__get_default_supplier_admin

Synopsis

```
#include  <NotificationService.h>
CosNotifyChannelAdmin_SupplierAdmin
CosNotifyChannelAdmin_EventChannel__get_default_supplier_admin(
               CosNotifyChannelAdmin_EventChannel  obj,
               CORBA_Environment                   *env );
```

Description

Returns the object reference of the SupplierAdmin object that is standard for the event channel.

This method secures an area for holding the object references. Use CORBA_Object_release() to release the area when it is no longer required.

Parameters

obj

The object reference for the event channel connected.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.18.4.3 CosNotifyChannelAdmin_ConsumerAdmin_obtain_notification_pull_supplier()

Name

CosNotifyChannelAdmin_ConsumerAdmin_obtain_notification_pull_supplier

Synopsis

```
#include  <NotificationService.h>
CosNotifyChannelAdmin_ProxySupplier
CosNotifyChannelAdmin_ConsumerAdmin_obtain_notification_pull_supplier(
                CosNotifyChannelAdmin_ConsumerAdmin  obj,
                CosNotifyChannelAdmin_ClientType     ctype,
                CosNotifyChannelAdmin_ProxyID        *proxy_id,
                CORBA_Environment                    *env );
```

Description

Creates a ProxySupplier object of the client type specified in ctype.

This method secures an area for holding the object references. Use CORBA_Object_release() to release the area when it is no longer required.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_EventChannel_get_default_supplier_admin.

ctype

The client type by which the ProxySupplier object is created is specified as follows:

CosNotifyChannelAdmin_ANY_EVENT: Handles any type event

CosNotifyChannelAdmin_STRUCTURED_EVENT: Handles StructuredEvent type event

proxy_id (out parameter)

The created ProxySupplier ID is set.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure, and returns the object reference for ProxySupplier.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system or user exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosNotifyChannelAdmin_AdminLimitExceeded

The upper limit for creating ProxySuppliers has been reached - no more can be created.

1.18.4.4 CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer()

Name

CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer

Synopsis

```
#include  <NotificationService.h>
CosNotifyChannelAdmin_ProxyConsumer
CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer(
```

```
                CosNotifyChannelAdmin_SupplierAdmin  obj,
                CosNotifyChannelAdmin_ClientType     ctype,
                CosNotifyChannelAdmin_ProxyID        *proxy_id,
                CORBA_Environment                    *env );
```

Description

Creates a ProxyConsumer object of the client type specified in ctype.

This method secures an area for holding the object references. Use CORBA_Object_release() to release the area when it is no longer required.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_EventChannel_get_default_consumer_admin.

ctype

The client type by which the ProxyConsumer object is created is specified as follows:

CosNotifyChannelAdmin_ANY_EVENT: Handles any type event

CosNotifyChannelAdmin_STRUCTURED_EVENT: Handles StructuredEvent type event

proxy_id (out parameter)

The created ProxyConsumer ID is set.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure, and returns the object reference for ProxyConsumer.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosNotifyChannelAdmin_AdminLimitExceeded

The upper limit for creating ProxyConsumers has been reached - no more can be created.

1.18.4.5 CosNotifyChannelAdmin_ConsumerAdmin__get_MyChannel()

Name

CosNotifyChannelAdmin_ConsumerAdmin__get_MyChannel

Synopsis

```
#include  <NotificationService.h>
CosNotifyChannelAdmin_EventChannel
CosNotifyChannelAdmin_ConsumerAdmin__get_MyChannel(
                CosNotifyChannelAdmin_ConsumerAdmin  obj,
                CORBA_Environment                    *env );
```

Description

Returns the object reference for the event channel that generated ConsumerAdmin.

This method secures an area for holding the object references. Use CORBA_Object_release() to release the area when it is no longer required.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_EventChannel__get_default_consumer_admin.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.18.4.6 CosNotifyChannelAdmin_SupplierAdmin__get_MyChannel()

Name

CosNotifyChannelAdmin_SupplierAdmin__get_MyChannel

Synopsis

```
#include  <NotificationService.h>
CosNotifyChannelAdmin_EventChannel
CosNotifyChannelAdmin_SupplierAdmin__get_MyChannel(
               CosNotifyChannelAdmin_SupplierAdmin  obj,
               CORBA_Environment                    *env );
```

Description

Returns the object reference for the event channel that generated SupplierAdmin.

This method secures an area for holding the object references. Use CORBA_Object_release() to release the area when it is no longer required.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_EventChannel__get_default_supplier_admin.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.18.4.7 CosNotifyChannelAdmin_ProxyPushConsumer_connect_any_push_supplier( )

Name

CosNotifyChannelAdmin_ProxyPushConsumer_connect_any_push_supplier

Synopsis

```
#include  <NotificationService.h>
void
CosNotifyChannelAdmin_ProxyPushConsumer_connect_any_push_supplier(
               CosNotifyChannelAdmin_ProxyPushConsumer  obj,
               CosEventComm_PushSupplier                push_supplier,
               CORBA_Environment                        *env );
```

Description

Connects an any type supplier to the event channel.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer.

push_supplier

The own object reference.

Specify CORBA_OBJECT_NIL in push_supplier if notification of disconnection is not required when the event channel is terminated.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosEventChannelAdmin_AlreadyConnected

The event channel is already connected.

Note

On reconnection to the event channel, start again from CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer.

1.18.4.8 CosNotifyChannelAdmin_ProxyPullSupplier_connect_any_pull_consumer()

Name

CosNotifyChannelAdmin_ProxyPullSupplier_connect_any_pull_consumer

Synopsis

```
#include  <NotificationService.h>
void
CosNotifyChannelAdmin_ProxyPullSupplier_connect_any_pull_consumer(
```

```
                CosNotifyChannelAdmin_ProxyPullSupplier  obj,
                CosEventComm_PullConsumer                pull_consumer,
                CORBA_Environment                        *env );
```

Description

Connects an any type consumer to the event channel.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_pull_supplier.

pull_consumer

The own object reference.

Specify CORBA_OBJECT_NIL in pull_consumer if notification of disconnection is not required when the event channel is terminated.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosEventChannelAdmin_AlreadyConnected

The event channel is already connected.

Note

On reconnection to the event channel, start again from CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_pull_supplier.

1.18.4.9 CosNotifyChannelAdmin_ProxyConsumer__get_MyType()

Name

CosNotifyChannelAdmin_ProxyConsumer__get _MyType

Synopsis

```
#include  <NotificationService.h>
CosNotifyChannelAdmin_ProxyType
CosNotifyChannelAdmin_ProxyConsumer__get_MyType(
                CosNotifyChannelAdmin_ProxyConsumer  obj,
                CORBA_Environment                    *env );
```

Description

Returns the following values for the Proxy object type:

CosNotifyChannelAdmin_PUSH_ANY: any type PUSH model

CosNotifyChannelAdmin_PULL_ANY: any type PULL model

CosNotifyChannelAdmin_PUSH_STRUCTURED: StructuredEvent type PUSH model

CosNotifyChannelAdmin_PULL_STRUCTURED: StructuredEvent type PULL model

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure, and the Proxy object type is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.18.4.10 CosNotifyChannelAdmin_ProxySupplier__get_MyType()

Name

CosNotifyChannelAdmin_ProxySupplier__get _MyType

Synopsis

```
#include  <NotificationService.h>
CosNotifyChannelAdmin_ProxyType
CosNotifyChannelAdmin_ProxySupplier__get_MyType(
               CosNotifyChannelAdmin_ProxySupplier  obj,
               CORBA_Environment                    *env );
```

Description

Returns the following values for the Proxy object type:

CosNotifyChannelAdmin_PUSH_ANY: any type PUSH model

CosNotifyChannelAdmin_PULL_ANY: any type PULL model

CosNotifyChannelAdmin_PUSH_STRUCTURED: StructuredEvent type PUSH model

CosNotifyChannelAdmin_PULL_STRUCTURED: StructuredEvent type PULL model

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_ConsumerAdmin_obtain_notification_pull_supplier.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure, and the Proxy object type is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.18.4.11 CosNotifyChannelAdmin_StructuredProxyPushConsumer_connect_structured_ push_supplier()

Name

CosNotifyChannelAdmin_StructuredProxyPushConsumer_connect_structured_push_supplier

Synopsis

```
#include  <NotificationService.h>
void
CosNotifyChannelAdmin_StructuredProxyPushConsumer_connect_structured_push_supplier(
               CosNotifyChannelAdmin_StrucuturedProxyPushConsumer  obj,
               CosNotifyComm_StructuredPushSupplier                push_supplier,
               CORBA_Environment                                   *env );
```

Description

Connects to the event channel as a StructuredEvent type supplier.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer.

push_supplier

The own object reference.

Specify CORBA_OBJECT_NIL in push_supplier if notification of disconnection is not required when the event channel is terminated.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure, and the Proxy object type is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosEventChannelAdmin_AlreadyConnected

The event channel is already connected.

Note

When reconnecting to the event channel, start again from CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer.

1.18.4.12 CosNotifyChannelAdmin_StructuredProxyPullSupplier_connect_structured_ pull_consumer()

Name

CosNotifyChannelAdmin_StructuredProxyPullSupplier_connect_structured_pull_consumer

Synopsis

```
#include  <NotificationService.h>
void
CosNotifyChannelAdmin_StructuredProxyPullSupplier_connect_structured_pull_consumer(
                CosNotifyChannelAdmin_StrucuturedProxyPullSupplier  obj,
                CosNotifyComm_StructuredPullConsumer                pull_consumer,
                CORBA_Environment                                   *env );
```

Description

Connects to the event channel as a StructuredEvent type consumer.

Parameters

obj

The object reference returned by CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_pull_supplier.

pull_consumer

The own object reference.

Specify CORBA_OBJECT_NIL in pull_consumer if notification of disconnection is not required when the event channel is terminated.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure, and the Proxy object type is returned.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosEventChannelAdmin_AlreadyConnected

The event channel is already connected.

Note

When reconnecting to the event channel, start again from CosNotifyChannelAdmin_ConsumerAdmin_obtain_notification_pull_supplier.

1.18.4.13 Interfaces Inherited

The following interfaces can be inherited and used.

(1) CosNotifyChannelAdmin_StructuredProxyPushConsumer_push_structured_event

(2) CosNotifyChannelAdmin_StructuredProxyPullSupplier_pull_structured_event

(3) CosNotifyChannelAdmin_StructuredProxyPullSupplier_try_pull_structured_event

(4) CosNotifyChannelAdmin_StructuredProxyPushConsumer_disconnect_structured_push_consumer

(5) CosNotifyChannelAdmin_StructuredProxyPullSupplier_disconnect_structured_pull_supplier

(6) CosNotifyChannelAdmin_ProxyPushConsumer_push

(7) CosNotifyChannelAdmin_ProxyPullSupplier_pull

(8) CosNotifyChannelAdmin_ProxyPullSupplier_try_pull

(9) CosNotifyChannelAdmin_ProxyPushConsumer_disconnect_push_consumer

(10) CosNotifyChannelAdmin_ProxyPullSupplier_disconnect_pull_supplier

For details of these interfaces, refer to the 1.18.3 CosNotifyComm Interface.

1.18.5 EventChannelFactory Interface

CosNotifyChannelAdmin_EventChannelFactory is the factory interface specified by the NotificationService.

Data type mapping

The data types used by this interface are mapped in the following way in C:

```
typedef CORBA_long CosNotifyChannelAdmin_ChannelID;
typedef struct {
    CORBA_unsigned_long               _maximum;
    CORBA_unsigned_long               _length;
    CosNotifyChannelAdmin_ChannelID   *_buffer;
} CosNotifyChannelAdmin_ChannelIDSeq;
```

1.18.5.1 CosNotifyChannelAdmin_EventChannelFactory_create_channel()

Name

CosNotifyChannelAdmin_EventChannelFactory_create_channel

Synopsis

```
#include  <NotificationService.h>
CosNotifyChannelAdmin_EventChannel
CosNotifyChannelAdmin_EventChannelFactory_create_channel(
                CosNotifyChannelAdmin_EventChannelFactory   obj,
                CosNotification_QoSProperties               *initial_qos,
                CosNotification_AdminProperties             *initial_admin,
                CosNotifyChannelAdmin_ChannelID             *id,
                CORBA_Environment                           *env );
```

Description

Generates an event channel with the QoS property items specified in initial_qos and initial_admin and the Admin property item, and specifies the event channel ID in id.

This method secures an area for holding the object references. Use CORBA_Object_release() to release the area when it is no longer required.

Parameters

obj

The object reference specifies and fetches the "NotificationService" in the identifier parameter in CORBA_ORB_resolve_initial_references.

initial_qos

The QoS property item which generates event channel.

initial_admin

The Admin property item which generates event channel.

id (out parameter)

The event channel ID is set.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosNotification_UnsupportedQoS

There is an error in the specified QoS property item or value.

ex_CosNotification_UnsupportedAdmin

There is an error in the specified Admin property item or value.

1.18.5.2 CosNotifyChannelAdmin_EventChannelFactory_get_all_channels()

Name

CosNotifyChannelAdmin_EventChannelFactory_get_all_channels

Synopsis

```
#include  <NotificationService.h>
CosNotifyChannelAdmin_ChannelIDSeq *
CosNotifyChannelAdmin_EventChannelFactory_get_all_channels(
               CosNotifyChannelAdmin_EventChannelFactory  obj,
               CORBA_Environment                          *env );
```

Description

Returns as sequence types the IDs for all event channels managed by the event factory.

This method secures sequence type areas for holding the object references. Use CORBA_Object_release() to release the areas when they are no longer required.

Parameters

obj

The object reference specifies and fetches the "NotificationService" in the identifier parameter in CORBA_ORB_resolve_initial_references.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure, and returns in sequence type the IDs for all event channels managed by the event factory.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.18.5.3 CosNotifyChannelAdmin_EventChannelFactory_get_event_channel()

Name

CosNotifyChannelAdmin_EventChannelFactory_get_event_channel

Synopsis

```
#include  <NotificationService.h>
CosNotifyChannelAdmin_EventChannel
CosNotifyChannelAdmin_EventChannelFactory_get_event_channel(
               CosNotifyChannelAdmin_EventChannelFactory  obj,
               CosNotifyChannelAdmin_ChannelID            id,
               CORBA_Environment                          *env );
```

Description

Returns the object references for event channels with the IDs specified in id.

This method secures areas for holding the object references. Use CORBA_Object_release() to release the area when it is no longer required.

Parameters

obj

The object reference specifies and fetches the "NotificationService" in the identifier parameter in CORBA_ORB_resolve_initial_references.

id

The acquired event channel ID.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in _major in the env structure, and returns object references for event channels that have the IDs specified in id.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosNotifyChannelAdmin_ChannelNotFound

The specified channel could not be found.

1.19 Connection Information Collection Function Interface

This interface is used to collect connection information for the event channel.

1.19.1 CosEventChannelAdmin Interface

This section details the functions associated with the CosEventChannelAdmin Interface.

1.19.1.1 CosEventChannelAdmin_EventChannel_create_util()

Name

CosEventChannelAdmin_EventChannel_create_util

Synopsis

```
#include  <EventService.h>
  CORBA_Object CosEventChannelAdmin_EventChannel_create_util(
```

```
                CosEventChannelAdmin_EventChannel           obj,
                CosEventChannelAdmin_EventChannel_UtilType  type,
                CORBA_Environment                           env );
```

Description

Creates the ES_ChannelUtil interace object reference.

This method secures an area for holding the object references. Use CORBA_Object_release() to release the area when it is no longer required.

Parameters

obj

The CosEventChannelAdmin_EventChannel object reference.

type

The interface type.

Specify the interface type in type.

Value specifiable in type	Return value information
CosEventChannelAdmin_CHANNEL_UTIL	Fetches ES_ChannelUtil interface object reference

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure, and returns the ESChannelUtil interface object reference.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.19.1.2 Interface Inherited

The following interface can be inherited and used:

CosNotifyChannelAdmin_EventChannel_create_util

For details, refer to CosEventChannelAdmin_EventChannel_create_util().

1.19.2 ES Interface

This section details the functions associated with the ES Interface.

1.19.2.1 ES_ChannelUtil_get_proxys()

Name

ES_ChannelUtil_get_proxys

Synopsis

```
#include  <EventService.h>
  ES_ChannelUtil_ProxyDataSeq *
  ES_ChannelUtil_get_proxys (
                ES_ChannelUtil              obj,
```

```
            ES_ChannelUtil_ProxyKind    kind,
            CORBA_Environment           *env );
typedef struct {
      CORBA_Object              proxy;
      CORBA_long                time;
      CORBA_long                ipaddress;
      ES_ChannelUtil_ProxyKind  kind;
} ES_ChannelUtil_ProxyData;
typedef sequence<ES_ChannelUtil_ProxyData> ES_ChannelUtil_ProxyDataSeq;
```

Description

Fetches consumers/suppliers connected to the event channel, and connection information.

This method secures areas for holding the connection information and suppliers/consumers connected to the event channel. Use CORBA_free() to release the areas when they are no longer required.

Parameters

obj

The ES_ChannelUtil object reference.

kind

The type of Proxy object fetched.

The connection information fetched depends on the value specified in kind. The following table lists the values specified in kind and the corresponding return inforrmation.

Table 1.3 kind Values and Corresponding Return Information

Value Specifiable in Kind	Return Value Information
ES_ChannelUtil_ALL_PROXYS	Fetches all connection information
ES_ChannelUtil_PROXY_CONSUMER	Fetches a list of all ProxyConsumers
ES_ChannelUtil_PROXY_SUPPLIER	Fetches a list of all ProxySuppliers
ES_ChannelUtil_PROXY_PULL_CONSUMER_ EVENT	Fetches a list of CosEventChannelAdmin ProxyPullConsumers
ES_ChannelUtil_PROXY_PULL_SUPPLIER _EVENT	Fetches a list of CosEventChannelAdmin ProxyPullSuppliers
ES_ChannelUtil_PROXY_PUSH_CONSUMER_ EVENT	Fetches a list of CosEventChannelAdmin ProxyPushConsumers
ES_ChannelUtil_PROXY_PUSH_SUPPLIER_EVENT	Fetches a list of CosEventChannelAdmin ProxyPushSuppliers
ES_ChannelUtil_PROXY_PULL_CONSUMER_ NOTIFY	Fetches a list of CosNotifyChannelAdmin ProxyPullConsumers
ES_ChannelUtil_PROXY_PULL_SUPPLIER_ NOTIFY	Fetches a list of CosNotifyChannelAdmin ProxyPullSuppliers
ES_ChannelUtil_PROXY_PUSH_CONSUMER_ NOTIFY	Fetches a list of CosNotifyChannelAdmin ProxyPushConsumers
ES_ChannelUtil_PROXY_PUSH_SUPPLIER_ NOTIFY	Fetches a list of CosNotifyChannelAdmin ProxyPushSuppliers
ES_ChannelUtil_STRUCTURED_PROXY_PULL_ CONSUMER	Fetches a list of CosNotifyChannelAdmin StructuredProxyPullConsumers
ES_ChannelUtil_STRUCTURED_PROXY_PULL_ SUPPLIER	Fetches a list of CosNotifyChannelAdmin StructuredProxyPullSuppliers

Value Specifiable in Kind	Return Value Information
ES_ChannelUtil_STRUCTURED_PROXY_PUSH_ CONSUMER	Fetches a list of CosNotifyChannelAdmin StructuredProxyPushConsumers
ES_ChannelUtil_STRUCTURED_PROXY_PUSH_ SUPPLIER	Fetches a list of CosNotifyChannelAdmin StructuredProxyPushSuppliers

The following table shows the values that can be specified for ES_ChannelUtil_ProxyData structure members:

Table 1.4 ES_ChannelUtil_ProxyData Structure Members and Settings

Member	Setting
Proxy	Object references for consumers/suppliers connected to the event channel.
time	Time connected to the event channel.
ipaddress	IP addresses of consumers/suppliers connected to the event channel.
kind	Proxy types of consumers/suppliers connected to the event channel. ES_ChannelUtil_PROXY_PULL_CONSUMER_EVENT ES_ChannelUtil_PROXY_PULL_SUPPLIER _EVENT ES_ChannelUtil_PROXY_PUSH_CONSUMER_EVENT ES_ChannelUtil_PROXY_PUSH_SUPPLIER _EVENT ES_ChannelUtil_PROXY_PULL_CONSUMER_NOTIFY ES_ChannelUtil_PROXY_PULL_SUPPLIER _NOTIFY ES_ChannelUtil_PROXY_PUSH_CONSUMER_NOTIFY ES_ChannelUtil_PROXY_PUSH_SUPPLIER _NOTIFY ES_ChannelUtil_STRUCTURED_PROXY_PULL_CONSUMER ES_ChannelUtil_STRUCTURED_PROXY_PULL_SUPPLIER ES_ChannelUtil_STRUCTURED_PROXY_PUSH_CONSUMER ES_ChannelUtil_STRUCTURED_PROXY_PUSH_SUPPLIER

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure, and returns the consumers/suppliers connected to the event channel and connection information.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

Notes

The 'Ipaddress' member is set to 0 when using the IP address in an IPv6 environment. In an IPv6 environment, use ES_ChannelUtil_get_proxys6().

1.19.2.2 ES_ChannelUtil_get_proxys6()

Name

ES_ChannelUtil_get_proxys6

Synopsis

```
#include  <EventService.h>
  ES_ChannelUtil_ProxyDataSeq *
  ES_ChannelUtil_get_proxys6 (
```

```
              ES_ChannelUtil              obj,
              ES_ChannelUtil_ProxyKind    kind,
              CORBA_Environment           *env );
  typedef struct {
        CORBA_Object              proxy;
        CORBA_long                time;
        CORBA_long                ipaddress[16];
        CORBA_long                ip_format;
        ES_ChannelUtil_ProxyKind  kind;
  } ES_ChannelUtil_ProxyData6;
  typedef sequence<ES_ChannelUtil_ProxyData6> ES_ChannelUtil_ProxyData6Seq;
```

Description

Fetches consumers/suppliers connected to the event channel and connection information.

When 'ipaddress' is set via the IPv6 form, 'ip_format' is set to 1. When 'ipaddress' is set via the IPv4 form, 'ip_format' is set to 0.

This method secures areas for storing the connection information and suppliers/consumers connected to the event channel. Use CORBA_free() to release these areas when they are no longer required.

Parameters

obj

The ES_ChannelUtil object reference.

kind

The type of Proxy object fetched.

The connection information fetched depends on the value specified in kind. For details of values that can be specified in kind, and the corresponding return inforrmation, refer to the Table 1.3 kind Values and Corresponding Return Information lists under 1.19.2.1 ES_ChannelUtil_get_proxys().

env

A structure that can contain exception information.

Return Values

For normal termination, the 'major' member of the env structure is set to CORBA_NO_EXCEPTION, and the consumers/suppliers connected to the event channel and the connection information are returned.

For abnormal termination, the '_major' member of the env structure is set to CORBA_SYSTEM_EXCEPTION, and the '_id' member of the env structure is set.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

Notes

The IPv4 client member 'Ipaddress' is set to the parallel IPv4 address when operating using the IP address in an IPv6 environment. For details of settings when operating in an IPv6 environment, refer to 'Operating in an IPv6 Environment' and 'config'(IP-version parameter) under 'CORBA Service Environment Definition' of the Tuning Guide.

1.19.2.3 ES_ChannelUtil_get_consumer_count()

Name

ES_ChannelUtil_get_consumer_count

Synopsis

```
#include  <EventService.h>
  CORBA_unsigned_long
```

```
  ES_ChannelUtil_get_consumer_count(
                ES_ChannelUtil       obj,
                CORBA_Environment    *env );
```

Description

Fetches the number of consumers connected to the event channel.

Parameters

obj

The ES_ChannelUtil object reference.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure, and returns the number of consumers connected to the event channel.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.19.2.4 ES_ChannelUtil_get_supplier_count()

Name

ES_ChannelUtil_get_supplier _count

Synopsis

```
#include  <EventService.h>
  CORBA_unsigned_long
  ES_ChannelUtil_get_supplier_count(
                ES_ChannelUtil       obj,
                CORBA_Environment    *env );
```

Description

Fetches the number of suppliers connected to the event channel.

Parameters

obj

The ES_ChannelUtil object reference.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure, and returns the number of suppliers connected to the event channel.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.19.2.5 ES_ChannelUtil_get_queue_length()

Name

ES_ChannelUtil_get_queue _length

Synopsis

```
#include  <EventService.h>
  CORBA_unsigned_long
  ES_ChannelUtil_get_queue_length_count(
               ES_ChannelUtil       obj,
               CORBA_Environment    *env );
```

Description

Fetches the number of event data items queued in the event channel.

Parameters

obj

The ES_ChannelUtil object reference.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure, and returns the number of event data items queued in the event channel.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.19.2.6 ES_ChannelUtil_local_begin()

Name

ES_ChannelUtil_local_begin

Synopsis

```
 #include  <NotificationServices.h>
 void
 ES_ChannelUtil_local_begin(
               ES_ChannelUtil  obj,
               CORBA_Object   proxy,
               CORBA_Environment  *env );
```

Description

Notifies the channel of starting the local transaction.

In the case of consumer, the following methods can be issued before ES_ChannelUtil_local_commit() or ES_ChannelUtil_local_rollback() is issued.

- CosEventComm_PullSupplier_pull()
- CosEventComm_PullSupplier_try_pull()
- CosNotifyComm_StructuredPullSupplier_pull_structured_event()

- CosNotifyComm_StructuredPullSupplier_try_pull_structured_event()

In the case of supplier, the following methods can be issued before ES_ChannelUtil_local_commit() or ES_ChannelUtil_local_rollback() is issued.

- CosEventComm_PushConsumer_push()
- CosNotifyComm_StructuredPushConsumer_push_structured_event()

If ES_ChannelUtil_local_commit() or ES_ChannelUtil_local_rollback() is not issued after issuance of this method, they are automatically rolled back by the notification service immediately after the local transaction timeout time passes.

This method cannot be issued twice for a proxy.

Parameters

obj

The ES_ChannelUtil object reference.

proxy

The object reference returned by CosNotifyChannelAdmin_ConsumerAdmin_obtain_notification_pull_supplier or CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.19.2.7 ES_ChannelUtil_local_commit()

Name

ES_ChannelUtil_local_commit

Synopsis

```
#include  <NotificationServices.h>
void
ES_ChannelUtil_local_commit(
              ES_ChannelUtil  obj,
              CORBA_Object   proxy,
              CORBA_Environment  *env );
```

Description

Notifies the channel of completing the local transaction.

In the case of consumer, message reception from the event channel completes when this method completes.

In the case of supplier, sending of data from the event channel completes when this method completes.

Parameters

obj

The ES_ChannelUtil object reference.

proxy

The object reference returned by CosNotifyChannelAdmin_ConsumerAdmin_obtain_notification_pull_supplier or CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.19.2.8 ES_ChannelUtil_local_rollback()

Name

ES_ChannelUtil_local_rollback

Synopsis

```
#include  <NotificationServices.h>
void
ES_ChannelUtil_local_rollback(
              ES_ChannelUtil  obj,
              CORBA_Object   proxy,
              CORBA_Environment  *env );
```

Description

Notifies the channel of cancellation of the local transaction. Issuance of this method allows the event channel status to return to that before ES_ChannelUtil_local_begin() is issued.

Parameters

obj

The ES_ChannelUtil object reference.

proxy

The object reference returned by CosNotifyChannelAdmin_ConsumerAdmin_obtain_notification_pull_supplier or CosNotifyChannelAdmin_SupplierAdmin_obtain_notification_push_consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.19.2.9 ES_ChannelUtil_pull_cancel()

Name

ES_ChannelUtil_pull_cancel

Synopsis

```
#include  <NotificationServices.h>
void
ES_ChannelUtil_pull_cancel(
              ES_ChannelUtil  obj,
              CORBA_Object   proxy,
              CORBA_Environment  *env );
```

Description

This method cancels the pull waiting status (waiting status by issuance of pull and ES_ChannelUtil_pull_wait()) of the target proxy.

Parameters

obj

The ES_ChannelUtil object reference.

proxy

The object reference returned by CosNotifyChannelAdmin_ConsumerAdmin_obtain_notification_pull_supplier.

Note: Specify the object reference specified in pull and ES_ChannelUtil_pull_wait().

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

1.19.2.10 ES_ChannelUtil_pull_wait()

Name

ES_ChannelUtil_pull_wait

Synopsis

```
#include  <NotificationServices.h>
void
ES_ChannelUtil_pull_wait(
              ES_ChannelUtil  obj,
              CORBA_Object   proxy,
              CORBA_Environment  *env );
```

Description

This method waits until the status of the target proxy becomes enabled to allow the proxy to fetch data with the pull method or the try_pull method. Returns to the invoking side when the status becomes enabled to allow fetching data after execution of this method.

Returns immediately if the target proxy is already enabled to fetch data with the pull method. If the status does not become enabled to allow fetching data within the waiting time, this method returns with timeout.

Parameters

obj

The ES_ChannelUtil object reference.

proxy

The object reference returned by CosNotifyChannelAdmin_ConsumerAdmin_obtain_notification_pull_supplier.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following detailed information is set:

ex_CosEventComm_Disconnected

Not connected to an event channel.

If automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

1.20 Other Distributed Transaction Linkage Interfaces Windows32/64 Solaris32 Linux32/64

This section details the following interfaces:

1.20.1 TransactionFactory Interface Windows32/64 Solaris32 Linux32/64

This section details the functions associated with the TransactionFactory Interface.

1.20.1.1 CosTransactions_TransactionFactory_create Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_TransactionFactory_create

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
CosTransaction_Control  CosTransactions_TransactionFactory_create(
                         CosTransactions_TransactionFactory  obj,
                        CORBA_unsigned_long  time_out,
                        CORBA_Environment           *env);
```

Description

Generates a new transaction and generates a Control object for the transaction context and returns the object reference.

Used to provide functions equivalent to the Current interface "begin", and therefore cannot be used in conjunction with the Current interface.

The object reference specified in obj specifies and fetches the following parameter in the Identifier parameter in CORBA_ORB_resolve_initial_references(): "TransactionService"

Specify the transaction timeout time in time_out. If the CosTransactions_Current_set_timeout method is used before the transaction starts, specify the time specified in that method.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure, and returns the Control object object reference.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

The following detailed information is set when a system exception occurs:

ex_CORBA_StExcep_NO_IMPLEMENT:

The event channel has not been started.

ex_CORBA_StExcep_COMM_FAILURE:

There was a communications failure.

ex_CORBA_StExcep_NO_RESOURCES:

There are insufficient resources.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

ex_CORBA_StExcep_INTERNAL:

CORBA Service error.

1.20.1.2 CosTransactions_TransactionFactory_recreate 

Name

CosTransactions_TransactionFactory_recreate

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
CosTransactions_Control  CosTransactions_TransactionFactory_recreate(
                        CosTransactions_TransactionFactory obj,
                        CosTransactions_PropagationContext *ctx,
                        CORBA_Environment           *env);
```

Description

Generates a Control object for the transaction context and returns the object reference.

Used to provide functions equivalent to the Current interface "begin", and therefore cannot be used in conjunction with the Current interface.

The object reference specified in obj specifies and fetches the following parameter in the Identifier parameter in CORBA_ORB_resolve_initial_references(): "TransactionService"

Specify the pointer to the PropagationContext fetched by CosTransactions_Coordinator_get_txcntext in ctx.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure, and returns the Control object object reference.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

The following detailed information is set when a system exception occurs:

ex_CORBA_StExcep_NO_IMPLEMENT:

The event channel has not been started.

ex_CORBA_StExcep_COMM_FAILURE:

There was a communications failure.

ex_CORBA_StExcep_NO_RESOURCES:

There are insufficient resources.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

ex_CORBA_StExcep_INTERNAL:

CORBA Service error.

1.20.2 Control Interface Windows32/64 Solaris32 Linux32/64

This section details the functions associated with the Control Interface.

1.20.2.1 CosTransactions_Control_get_terminator Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Control_get_terminator

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
CosTransactions_Terminator CosTransactions_Control_get_terminator(
                                CosTransactions_Control  Control,
                                CORBA_Environment       *env);
```

Description

Returns a Terminator object that terminates transactions.

This Terminator object can also be used to terminate transactions from server applications.

Specify the Control object fetched by the Current interface get_control function in Control.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure, and returns the Terminator object.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

The following detailed information is set when a user exception occurs:

ex_CosTransactions_Unavailable

The Terminator object cannot be used.

The following detailed information is set when a system exception occurs:

ex_CORBA_StExcep_NO_IMPLEMENT:

The event channel has not been started.

ex_CORBA_StExcep_COMM_FAILURE:

There was a communications failure.

ex_CORBA_StExcep_NO_RESOURCES:

There are insufficient resources.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

ex_CORBA_StExcep_INTERNAL:

CORBA Service error.

1.20.2.2 CosTransactions_Control_get_coordinator Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Control_get_coordinator

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
CosTransactions_Coordinator  CosTransactions_Control_get_coordinator(
                               CosTransactions_Control   Control,
                               CORBA_Environment       *env);
```

Description

Returns a Coordinator object that coordinates transactions.

This Coordinator object can also be used to coordinate transactions in server applications.

Specify the Control object fetched by the Current interface get_control function in Control.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure, and returns the Coordinator object.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

The following detailed information is set when a user exception occurs:

ex_CosTransactions_Unavailable

The Coordinator object cannot be used.

The following detailed information is set when a system exception occurs:

ex_CORBA_StExcep_NO_IMPLEMENT:

The event channel has not been started.

ex_CORBA_StExcep_COMM_FAILURE:

There was a communications failure.

ex_CORBA_StExcep_NO_RESOURCES:

There are insufficient resources.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

ex_CORBA_StExcep_INTERNAL:

CORBA Service error.

1.20.3 Coordinator Interface Windows32/64 Solaris32 Linux32/64

In the Coordinator interface, the following functions are the same functions as in the Current interface. Refer to the 1.13 Current Interface for details of these functions.

- get_status
- rollback_only
- get_transaction_name

1.20.3.1 CosTransactions_Coordinator_register_synchronization Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Coordinator_register_synchronization

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
void  CosTransactions_Coordinator_register_synchronization(
                                CosTransactions_Coordinator  coordinator,
                                CosTransactions_Synchronization sync,
                                CORBA_Environment       *env);
```

Description

Registers the Synchronization object in the transaction.

Specifies in coordinator the object reference for the Coordinator object fetched by the Control interface get_coordinator function.

The Synchronization object is created by using the Synchronization interface.

The Synchronization object is called when it is registered, and before and after a transaction is committed.

The user-created Synchronization object object reference is specified in sync.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

The following detailed information is set when a user exception occurs:

ex_CosTransactions_Inactive

The transaction has already been prepared.

ex_CosTransactions_SynchronizationUnavailable

The Coordinator does not support the Synchronization interface.

The following detailed information is set when a system exception occurs:

ex_CORBA_StExcep_NO_IMPLEMENT:

The event channel has not been started.

ex_CORBA_StExcep_COMM_FAILURE:

There was a communications failure.

ex_CORBA_StExcep_NO_RESOURCES:

There are insufficient resources.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

ex_CORBA_StExcep_INTERNAL:

CORBA Service error.

1.20.3.2 CosTransactions_Coordinator_get_txcontext Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Coordinator_get_txcontext

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
CosTransactions_PropagationContext  CosTransactions_Coordinator_get_txcontext(
                                CosTransactions_Coordinator  coordinator,
                                    CORBA_Environment       *env);
```

Description

Returns the transaction context that manages the transaction information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure, and returns the transaction context.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

The following detailed information is set when a user exception occurs:

ex_CosTransactions_Unavailable

The Coordinator object cannot be used.

The following detailed information is set when a system exception occurs:

ex_CORBA_StExcep_NO_IMPLEMENT:

The event channel has not been started.

ex_CORBA_StExcep_COMM_FAILURE:

There was a communications failure.

ex_CORBA_StExcep_NO_RESOURCES:

There are insufficient resources.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

ex_CORBA_StExcep_INTERNAL:

CORBA Service error.

1.20.4 Terminator Interface Windows32/64 Solaris32 Linux32/64

The functions provided by the Terminator interface are identical to the commit and rollback functions of the Current interface. This interface cannot be used in conjunction with the Current interface.

1.20.4.1 CosTransactions_Terminator_commit Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Terminator_commit

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
void  CosTransactions_Terminator_commit(
                        CosTransactions_Terminator  terminator,
                        CORBA_boolean           report_heuristics,
                        CORBA_Environment       *env);
```

Description

Commits a transaction.

Specifies in terminator the object reference for the Terminator object fetched by the Control_get_terminator method.

A heuristic error is posted if CORBA_TRUE is specified in report_heuristics.

The functions provided by the Terminator interface are identical to the commit function of the Current interface. This interface cannot be used in conjunction with the Current interface.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

The following detailed information is set when a user exception occurs:

ex_CosTransactions_HeuristicMixed:

The completion statuses of the transactions are that some have been committed and some have been rolled back.

ex_CosTransactions_HeuristicHazard:

The completion status of the transaction is uncertain.

The following detailed information is set when a system exception occurs:

ex_CORBA_StExcep_TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

ex_CORBA_StExcep_NO_IMPLEMENT:

The event channel has not been started.

ex_CORBA_StExcep_COMM_FAILURE:

There was a communications failure.

ex_CORBA_StExcep_NO_RESOURCES:

There are insufficient resources.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

ex_CORBA_StExcep_INTERNAL:

CORBA Service error.

1.20.4.2 CosTransactions_Terminator_rollback 

Name

CosTransactions_Terminator_rollback

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
void  CosTransactions_Terminator_rollback(
                        CosTransactions_Terminator  terminator,
                        CORBA_boolean           report_heuristics,
                        CORBA_Environment       *env);
```

Description

Rolls back a transaction.

Specifies in terminator the object reference for the Terminator object fetched by the Control_get_terminator method.

The functions provided by the Terminator interface are identical to the rollback function of the Current interface. This interface cannot be used in conjunction with the Current interface.

Return Values

For normal termination, CORBA_NO_EXCEPTION is set in major in the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION is set in _major in the env structure, and a system exception is set in _id in the env structure.

The following detailed information is set when a system exception occurs:

ex_CORBA_StExcep_TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

ex_CORBA_StExcep_NO_IMPLEMENT:

The event channel has not been started.

ex_CORBA_StExcep_COMM_FAILURE:

There was a communications failure.

ex_CORBA_StExcep_NO_RESOURCES:

There are insufficient resources.

ex_CORBA_StExcep_NO_MEMORY:

Could not secure dynamic memory.

ex_CORBA_StExcep_INTERNAL:

CORBA Service error.

1.20.5 Synchronization Interface Windows32/64 Solaris32 Linux32/64

This section details the functions associated with the Synchronization Interface.

1.20.5.1 CosTransactions_Synchronization_before_completion Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Synchronization_before_completion

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
CosTransactions_Synchronization_before_completion(
```

```
                        CosTransactions_Synchronization        sync,
                                CORBA_Environment       *env);
```

Description

This function is used to synchronize transactions that have been created.

When the Database Linkage Service receives a commit request, it executes before a commit request to a resource manager.

The Synchronization object specified by the register_synchronization function in the Coordinator interface is specified in sync.

Return Values

For normal termination, processing continues.

For abnormal termination, the Database Linkage Service rolls the transaction back.

1.20.5.2 CosTransactions_Synchronization_after_completion Windows32/64 Solaris32 Linux32/64

Name

CosTransactions_Synchronization_after_completion

Synopsis

```
#include  "orb.h"
#include  "CosTransactions.h"
CosTransactions_Synchronization_after_completion(
                                CosTransactions_Synchronization  sync,
                                CosTransactions_Status  status,
                                CORBA_Environment       *env);
```

Description

This function is used to synchronize transactions that have been created.

When the Database Linkage Service receives a commit or rollback request, it executes the commit or rollback request after a request has been received from a resource, and the transaction has been completed.

The Synchronization object specified by the register_synchronization function in the Coordinator interface is specified in sync.

The transaction status is set in status. The transaction status is the same as that referenced with the get_status method.

Return Values

For normal or abnormal termination, processing continues, and a commit or rollback request is returned to the requesting user application.

1.21 Interstage Directory Service Interface

This section explains the Interstage Directory Service client API function specifications.

The API functions that are supported in the client API library are shown in the following table.

Table 1.5 Interstage Directory Service Client API Functions

Classification	Function	Explanation	Extended function
Interface for opening and closing sessions	ldap_init()	Initializes the session with the repository server.	
	ldapssl_init()	Initializes the session with the repository server using ssl.	

Classification	Function	Explanation	Extended function
	ldap_unbind()	Closes the connection to the repository server, and unbinds the session.	
	ldap_unbind_s()	Used for the same processing as ldap_unbind().	
Interface for session handle option settings/reference	ldap_get_option()	References the value set for the session handle option.	
	ldap_set_option()	Sets the session handle option.	
User authentication interface with the repository server	ldap_simple_bind()	Used for simple authentication with the repository server. (Asynchronous type)	
	ldap_simple_bind_s()	Used for simple authentication with the repository server. (Synchronous type)	
Interface for entry search	ldap_search()	Searches for the specified condition entry. (Asynchronous type)	
	ldap_search_s()	Searches for the specified condition entry. (Synchronous type)	
	ldap_search_st()	Searches for the specified condition entry. (Synchronous type, with time limit)	
	ldap_search_ext()	Searches for the specified condition entry. (Asynchronous type, LDAP V3 extended function)	
	ldap_search_ext_s()	Searches for the specified condition entry. (Synchronous type, LDAP V3 extended function)	
Interface for comparing attribute values	ldap_compare()	Compares the entry attribute value with the specified value. (Asynchronous type)	
	ldap_compare_s()	Compares the entry attribute value with the specified value. (Synchronous type)	
	ldap_compare_ext()	Compares the entry attribute value with the specified value. (Asynchronous type, LDAP V3 extended function)	
	ldap_compare_ext_s()	Compares the entry attribute value with the specified value. (Synchronous type, LDAP V3 extended function)	
Interface for changing entries	ldap_modify()	Updates the specified entry. (Asynchronous type)	
	ldap_modify_s()	Updates the specified entry. (Synchronous type)	
	ldap_modify_ext()	Updates the specified entry. (Asynchronous type, LDAP V3 extended function)	

Classification	Function	Explanation	Extended function
	ldap_modify_ext_s()	Updates the specified entry. (Synchronous type, LDAP V3 extended function)	
Interface for changing entry names	ldap_rename()	Changes the specified entry DN or RDN. (Asynchronous type, LDAP V3 extended function)	
	ldap_rename_s()	Changes the specified entry DN or RDN. (Synchronous type, LDAP V3 extended function)	
Interface for adding entries	ldap_add()	Adds the specified entry. (Asynchronous type)	
	ldap_add_s()	Adds the specified entry. (Synchronous type)	
	ldap_add_ext()	Adds the specified entry. (Asynchronous type, LDAP V3 extended function)	
	ldap_add_ext_s()	Adds the specified entry. (Synchronous type, LDAP V3 extended function)	
Interface for deleting entries	ldap_delete()	Deletes the specified entry. (Asynchronous type)	
	ldap_delete_s()	Deletes the specified entry. (Synchronous type)	
	ldap_delete_ext()	Deletes the specified entry. (Asynchronous type, LDAP V3 extended function)	
	ldap_delete_ext_s()	Deletes the specified entry. (Synchronous type, LDAP V3 extended function)	
Interface for canceling asynchronous processing	ldap_abandon()	Cancels asynchronous processing.	
	ldap_abandon_ext()	Cancels asynchronous processing. (LDAP V3 extended function)	
Interface for receiving/ analyzing processing results	ldap_result()	Receives asynchronous function processing results and search result entries.	
	ldap_msgid()	References the message ID of the specified message.	
	ldap_msgtype()	References the type of the specified message.	
Interface for obtaining error information	ldap_parse_result()	Reads the LDAP error information of the specified "result" information.	
	ldap_err2string()	Converts the LDAP error code to a string.	
	ldapssl_error()	Reads SSL library error codes.	
Interface for processing message lists	ldap_first_message()	Returns the address of the first message.	
	ldap_next_message()	Returns the address of the next message. (The message that follows the message read previously)	

Classification	Function	Explanation	Extended function
	ldap_count_messages()	Counts the number of messages contained in the message list.	
Interface for processing entry lists	ldap_first_entry()	Returns the first entry of the specified message.	
	ldap_next_entry()	Returns the next entry of the specified message. (The entry that follows the entry read previously)	
	ldap_count_entries()	Counts the number of entries in the specified message.	
Interface for reading attributes	ldap_first_attribute()	Returns the first attribute of the specified entry.	
	ldap_next_attribute()	Returns the next attribute of the specified entry. (The attribute that follows the attribute read previously)	
Interface for reading attribute values	ldap_get_values()	Reads the attribute value of the specified attribute. (If the attribute value is string data)	
	ldap_get_values_len()	Reads the attribute value of the specified attribute. (If the attribute value is binary data)	
	ldap_count_values()	Counts the number of read attribute values. (If the attribute value is string data)	
	ldap_count_values_len()	Counts the number of read attribute values. (If the attribute value is binary data)	
Interface for reading/ analyzing the DN	ldap_get_dn()	Reads the DN of the specified entry.	
	ldap_explode_dn()	Splits the specified DN into it's component parts (RDNs).	
	ldap_explode_rdn()	Splits the specified RDN into it's component parts.	
	ldap_dn2ufn()	Converts the specified DN to user-friendly format.	
Interface for releasing dynamic memory	ber_free()	Releases the memory that the specified BerElement structure used.	
	ldap_ber_free()	Releases the memory that the specified BerElement structure used. (This is not recommended.)	
	ldap_memfree()	Releases the memory obtained dynamically by the LDAP API function.	
	ldap_msgfree()	Releases the memory that the message list notified in ldap_result() used.	
	ldap_value_free()	Releases the pointer array of the string data.	
	ldap_value_free_len()	Releases the pointer array of the binary data.	
Interface for referencing version information	ldap_version()	References library version information.	

Note

Interstage Directory Service does not support LDAP V2 protocol.

1.21.1 Interface for Opening and Closing Sessions

Session Open/Default Settings

Function	Explanation
ldap_init()	Initializes the session with the repository server.
ldapssl_init()	Initializes the session with the repository server using SSL.

Session Close

Function	Explanation
ldap_unbind()	Closes the connection to the repository server, and unbinds the session.
ldap_unbind_s()	Used for the same processing as ldap_unbind().

1.21.1.1 ldap_init()

Name

ldap_init

Synopsis

```
#include "idldap.h"
LDAP  *ldap_init(
        const char  *hostname,
        int  portno );
```

Description

This function opens the LDAP session.

For functions called after this, the session handle obtained here should be specified in the parameters. The session handle is valid until the session is closed.

Authentication is executed using the DN and password. For details of simple authentication, refer to 1.21.3 User Authentication Interface With the Repository Server.

Parameters

hostname

Specify the host name of the repository server, or the address of the string that indicates the IP address. If more than one host name or IP address is specified, use a space to separate the names. If NULL is specified in 'hostname', it is as if "localhost" were specified. If more than one host is specified, test the connections in the order in which they will be established. Execute communication with the first host with which successful connection is established.

portno

Specify the port number of the repository server. If the default LDAP port is used, specify "0" as the port number. The default port number is LDAP_PORT "389".

Return Values

This function returns the following values:

- For normal termination: Session handle

  For normal termination, the pointer to the obtained session handle is notified in the session handle.

- For abnormal termination: NULL

  For abnormal termination, the possible causes are:

  - Insufficient system resources for obtaining memory for the session handle
  - Incorrect parameter specification

Notes

- Sharing a session

  It is not possible to share an opened session between multiple threads and execute an LDAP request from each thread. To execute an LDAP request using threads, use the session obtained in ldap_init() in each thread, and execute the LDAP request from each thread.

- Specifying multiple hosts

  This function cannot be used to connect to a repository that is running in the specified host if there are two or more repositories.

  (If the repository in the first host is not running, try connecting to the second host.)

  If more than one host is specified, (and it is not possible to connect to the first host at the socket level), try connecting to the second host. To perform the above type of repository start check, it must be possible to use the API in the application to check the repository start using ldap_simple_bind()/ldap_simple_bind_s().

1.21.1.2 ldapssl_init()

Name

ldapssl_init

Synopsis

```
#include "idldap.h"
LDAP  *ldapssl_init(
        const char  *hostname,
        int  portno,
        SSLENV  *sslenv);
```

Description

This function uses SSL to open the LDAP session. Communication with the server is secure.

For functions called after this, the session handle obtained here should be specified in the parameters. The session handle is valid until the session is closed.

Authentication is executed using the DN and password. For details of simple authentication, refer to 1.21.3 User Authentication Interface With the Repository Server. This function is thread unsafe.

Parameters

hostname

Specify the host name of the repository server, or the address of the string that indicates the IP address. If more than one host name or IP address is specified, use a space to separate the names. If NULL is specified in 'hostname', it is as if "localhost" were specified. If more than one host is specified, test the connections in the order in which they will be established. Execute communication with the first host with which successful connection is established.

portno

Specify the port number of the repository server. If the default LDAP port is used, specify "0" as the port number. The default port number is "636".

sslenv

Specifies the SSLENV structure address. If SSL is used, the SSL operating environment information must be set in the structure specified here, and passed as the parameter. Set the following values for the structure members.

SSLENV structure members

ssl_version

Specifies the SSL protocol version used.

- SSL version 2.0: [2]
- SSL version 3.0: [3]
- TLS version 1.0: [31]

If this is omitted ([0] is specified), the settings are the same as for the SSL version 3.0. If a different value is specified, processing is the same as for when SSL is not used (the same as for ldap_init()).

ssl_verify

Specifies if certificate verification is used.

- Certificate verification is not used: [0]
- Client certificate verification is used: [1]
- Verification of certificates sent from the server is used: [2]

If a different value is specified, the settings are the same as for the following values:

- SSL version 2.0: [1]
- SSL version 3.0 or TLS version 1.0: [2]

crypt (optional)

Select the encryption method used for SSL from those in the tables below, and specify the address for the encryption algorithm string. If more than one encryption method is specified, separate them using a colon (":"). Set the same SSL definition as used in the communication partner Interstage Directory Service server.

- If [2] is specified for the SSL protocol version (ssl_version), specify values from the following table:

Encryption algorithm string	Description
"DES-CBC3-MD5"	168bit triple DES code, MD5 MAC
"RC4-MD5"	128bit RC4 code, MD5 MAC
"RC2-MD5"	128bit RC2 code, MD5 MAC
"DES-CBC-MD5"	56bit DES code, MD5 MAC
"EXP-RC4-MD5"	40bit RC4 code, MD5 MAC
"EXP-RC2-MD5"	40bit RC2 code, MD5 MAC

- If [3] or [31] is specified for the SSL protocol version (ssl_version), specify values from the following table:

Encryption algorithm string	Description
"RSA-SC2000-256-SHA"	256bit SC2000 code, SHA-1 MAC
"RSA-AES-256-SHA"	256bit AES code, SHA-1 MAC
"RSA-SC2000-128-SHA"	128bit SC2000 code, SHA-1 MAC
"RSA-AES-128-SHA"	128bit AES code, SHA-1 MAC
"RSA-3DES-SHA"	168bit triple DES code, SHA-1 MAC
"RSA-RC4-MD5"	128bit RC4 code, MD5 MAC
"RSA-RC4-SHA"	128bit RC4 code, SHA-1 MAC

Encryption algorithm string	Description
"RSA-DES-SHA"	56bit DES code, SHA-1 MAC
"RSA-EXPORT-RC4-MD5"	40bit RC4 code, MD5 MAC
"RSA-EXPORT-RC2-MD5"	40bit RC2 code, MD5 MAC
"RSA-NULL-MD5"	None, MD5 MAC
"RSA-NULL-SHA"	None, SHA-1 MAC

Encryption method types supported in Interstage Application Server are as follows:

- Public key encryption method: RSA
- Private key encryption method: SC2000, AES, DES, 3DES (triple DES), RC4, RC2 (NULL indicates that encryption is not used)
- Private key processing mode: CBC (the numerical value is the block length)
- Hash key: SHA, MD5

MAC is the message authenticator.

If this is omitted (NULL is specified), the settings are shown below for each SSL protocol version (ssl_version) specified. In the following description, a line feed has been inserted after each encryption method.

- If [2] is specified for the SSL protocol version (ssl_version):

  DES-CBC3-MD5:

  DES-CBC-MD5:

  RC4-MD5:

  RC2-MD5:

  EXP-RC4-MD5:

  EXP-RC2-MD5:

- [3] or [31] is specified for the SSL protocol version (ssl_version)

  RSA-SC2000-256-SHA:

  RSA-AES-256-SHA

  RSA-SC2000-128-SHA

  RSA-AES-128-SHA

  RSA-3DES-SHA

  RSA-RC4-MD5

  RSA-RC4-SHA

  RSA-DES-SHA

  RSA-EXPORT-RC4-MD5

  RSA-EXPORT-RC2-MD5

slot_path (essential)

Specifies the slot information directory address (give the full path for the directory name). Specify the slot information directory that was specified when the certificate/key management environment was created.

tkn_lbl (essential)

Specifies the token label address. Specify the token label that was specified when the certificate/key management environment was created.

tkn_pwd (essential)

Specifies the user PIN address. Specify the user PIN address certificate/key that was specified when the management environment was created.

cert_path (essential)

Specifies the application management directory address (give the full path for the directory name). Specify the application management directory that was specified when the certificate/key management environment was created.

user_cert (optional)

Specifies the user certificate nickname. If this is omitted (NULL is specified). The settings are the same as for all your own certificates registered for certificate management. Specify the user certificate nickname that was specified when the user certificate was registered.

ssl_err (essential)

Sets the SSL error code. This must be initialized to [0].

ssl_err_detail (essential)

Sets the SSL error detail code. This must be initialized to [0].

ssl_timer (optional)

Sets the SSL timer (in seconds). If [0] is specified, the timer is [3600] seconds. This is used for the maximum wait time for the Repository server response.

ssl_err_funcinfo (optional)

Sets the SSL library maintenance code.

Note

- For details on creating the certificate and key management environment, refer to 'Creating a Certificate and Key Management Environment (Client)' in the 'Directory Service Operator's Guide'.
- For details on creating user certificates (Site certificates), refer to ' Registering a Certificate and CRL (Client)' in the 'Directory Service Operator's Guide'.

Return Values

This function returns the following values:

- For normal termination: Session handle

  The pointer to the obtained session handle is notified in the session handle.

- For abnormal termination: NULL

  The possible causes are:

  - Insufficient system resources for obtaining memory for the session handle
  - When the trace was obtained, backup of the trace files failed.
  - The SSL library package was not installed.
  - SSL environment settings are incorrect (an essential option was not set).
  - An error occurred in the SSL library.

Notes

- SSLENV structure area

  The SSL library error code output in "ldap_unbind/ldap_unbind_s" is also set in the SSLENV structure specified in the "sslenv" parameter. For this reason, do not release the area specified in the "sslenv" parameter until the session handle release is complete.

- SSL error code

  The SSL library error code output in "ildapssl_init()" cannot be referenced in ldapssl_error(). For this reason, the values for the SSLENV structure members ("ssl_err" and "ssl_err_detail") must be checked.

- Sharing a session

  It is not possible to share an opened session between multiple threads and execute an LDAP request from each thread. To execute an LDAP request using threads, use the session obtained in ldapssl_init() in each thread, and execute the LDAP request from each thread.

- Thread exclusion processing

  ldapssl_init() is a thread unsafe function. Use exclusion processing between each thread on the application side when this function is called from more than one thread at the same time. If exclusion is not used, the behavior cannot be guaranteed.

- Certificate and key management environments

  A running process cannot use more than one certificate and key management environment at the same time.

1.21.1.3 ldap_unbind()

Name

ldap_unbind

Synopsis

```
#include "idldap.h"
int  ldap_unbind( LDAP  *ld );
```

Description

This function unbinds the repository server connection and closes the session. ldap_unbind() and ldap_unbind_s() both execute processing as synchronous types, and are functionally the same.

Parameter

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: A non-LDAP_SUCCESS LDAP error code

Notes

- Unbinding the session handle

  ldap_unbind() unbinds the specified session handle and then returns to the call source. For this reason, the session handle (ld parameter) cannot be used after ldap_unbind() has completed.

1.21.1.4 ldap_unbind_s()

Name

ldap_unbind_s

Synopsis

```
#include "idldap.h"
int  ldap_unbind_s( LDAP  *ld );
```

Description

This function unbinds the repository server connection and closes the session. ldap_unbind() and ldap_unbind_s() both execute processing as synchronous types, and are functionally the same.

Parameter

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: A non-LDAP_SUCCESS LDAP error code

Notes

- Unbinding the session handle

  ldap_unbind_s() unbinds the specified session handle and then returns to the call source. For this reason, the session handle (ld parameter) cannot be used after ldap_unbind_s() has completed.

1.21.2 Interface for Session Handle Option Settings/Reference

Session Handle Option Settings/Reference

Function	Explanation
ldap_get_option()	References the value set for the session handle option.
ldap_set_option()	Sets the session handle option.

Session Handle Options

In the client API library, the operating environment for each session is set/referenced as the session handle option.

The session handle options that can be used are shown in the following table.

Table 1.6 Session Handle Options that can be Used

Option type	Option value	Value for area indicated by optdata (Meaning of the value)	optdata Format	
			set	get
LDAP_OPT_API_INFO	0x00	References the API version information.	Impossible	LDAPAPIInfo *(*1)
LDAP_OPT_DESC	0x01	References the socket identifier being used. This interface is not recommended. "-1" is set before Bind processing.	Impossible	int *
LDAP_OPT_SIZELIMIT (*2)	0x03	This is the maximum number of entries sent by the repository server. "LDAP_NO_LIMIT" means that the number of entries is unlimited.	int *	int *

Option type	Option value	Value for area indicated by optdata (Meaning of the value)	optdata Format	
			set	get
		The default value is "LDAP_NO_LIMIT".		
LDAP_OPT_ TIMELIMIT (*2)	0x04	This is the maximum search time (in seconds) in the repository server. "LDAP_NO_LIMIT" means that the search time is unlimited. The default value is "LDAP_NO_LIMIT".	int *	int *
LDAP_OPT_ RESTART	0x09	This is the continuation behavior following an interruption. The values are shown below. - LDAP_OPT_OFF (default value) Recovery is not executed in the library internally following interruption by a signal. (An error is returned to the call source.) - LDAP_OPT_ON Recovery is executed in the library internally following interruption by a signal.	void *	int *
LDAP_OPT_ PROTOCOL_ VERSION	0x11	This is the LDAP protocol that is used. The values are shown below. - LDAP_VERSION2 (default value) LDAP V2 protocol is used. - LDAP_VERSION3 LDAP V3 protocol is used.	int *	int *
LDAP_OPT_ HOST_NAME	0x30	This is the host name of the repository server. The name specified here is used as the default host name. Specify either of the following formats to reference the repository server: - "host name" - "host name:port number" When an IPv6 address is specified, "[host name]:port number" If ldap_set_option() is used to specify this using the "host name" format, when ldap_get_option() references it's value, the result format is: "hostname:port number". The "port number" is the value which was specified by ldap_init().	char *	char ** (*3)
LDAP_OPT_ RESULT_CODE	0x31	This is the latest LDAP error number.	int *	int *
LDAP_OPT_ ERROR_NUMBER	0x31	This is the latest LDAP error number.	int *	int *

Option type	Option value	Value for area indicated by optdata (Meaning of the value)	optdata Format	
			set	get
		This interface is not recommended, because it is retained to maintain compatibility. Use LDAP_OPT_RESULT_CODE instead.		
LDAP_OPT_ ERROR_STRING	0x32	This is the address of the latest LDAP error message.	char *	char ** (* 3)
LDAP_OPT_ CONNTIME (*4)	0x50	This is the wait time until the timeout for "connect". The specified value is the maximum wait time (in seconds) until the server response to "connect". - -1 (default value) The timeout time is determined by the OS default value. - LDAP_NO_LIMIT(0) There is no wait time. - Positive value The wait time is the specified time. This interface is not recommended, because it is retained to maintain compatibility. Use LDAP_OPT_NETWORK_ TIMEOUT instead.	int *	int *
LDAP_OPT_ NETWORK_ TIMEOUT	0x5005	This is the wait time until the timeout for "connect". The specified value is the maximum wait time (in seconds) until the server response to "connect". - NULL (default value) The timeout time is determined by the OS default value. - struct timeval structure The wait time is the specified time.	struct timeval *	struct timeval ** (*3)
Windows32/64 LDAP_OPT_WSINIT	0x51	This implements Windows Sockets DLL initialization and end processing. (Note 5) LDAP_NO_WSINIT is used so that Windows Sockets DLL initialization and end processing is not implemented in LDAP-API. The initialization settings perform initialization and end processing.	int *	int *

Note

*1 Set "LDAP_API_INFO_VERSION" in "ldapai_info_version" of the LDAPAPIInfo structure beforehand.

The area for ldapai_vendor_name in the LDAPAPIInfo Structure must be unbound using ldap_memfree().

*2 The LDAP_OPT_SIZELIMIT and LDAP_OPT_TIMELIMIT values may be invalid, depending on the value set for the DN accessed from the client (the bind DN) and the maximum searchable entry number and search timeout time in the server. For details, refer to 1.21.4 Interface for Entry Search.

*3 The area must be unbound using ldap_memfree().

*4 This option is shared with LDAP_OPT_NETWORK_TIMEOUT.

*5 Windows32/64

LDAP C API uses Windows Sockets DLL to perform communication using the TCP/IP protocol. There is no need for initialization processing in the user application because Windows Sockets DLL initialization (WSAStartup()) and end processing (WSACleanup()) are performed by LDAP C API.

If Windows Sockets DLL initialization/end processing is performed in the user application, however, the LDAP_OPT_WSINIT session handle option must be set so that initialization/end processing is not performed in LDAP C API.

1.21.2.1 ldap_set_option()

Name

ldap_set_option

Synopsis

```
#include "idldap.h"
int  ldap_set_option(
        LDAP  *ld,
        int  option,
        const void  *optdata );
```

Description

This function sets the session handle option value.

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

option

Specify the option type that is set. For details of the types/values that can be set for options, refer to Session Handle Options.

optdata

Specify the address for the area in which the value set for the option is to be stored.

Return Values

This function returns the following values:

- For normal termination: "0"
- For abnormal termination: "-1"

  For abnormal termination, the possible causes are:

  - An incorrect parameter was specified.
  - There is insufficient usable memory.

1.21.2.2 ldap_get_option()

Name

ldap_get_option

Synopsis

```
#include "idldap.h"
int  ldap_get_option(
        LDAP  *ld,
```

```
        int  option,
        void  *optdata );
```

Description

This function reads the value set for the session handle option.

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

option

Specify the option type that is referenced. For details of the types/values that can be set for options, refer to Session Handle Options.

optdata

Specify the address for the area in which the read option value is to be stored.

Return Values

This function returns the following values:

- For normal termination: "0"

  For normal termination, the value set for the read parameter is stored in the area specified in the "optdata" parameter.

- For abnormal termination: "-1"

  For abnormal termination, the possible causes are:

  - An incorrect parameter was specified.
  - There is insufficient usable memory.

1.21.3 User Authentication Interface With the Repository Server

Function	Function Explanation
ldap_simple_bind()	Used for simple authentication with the repository server. (Asynchronous type)
ldap_simple_bind_s()	Used for simple authentication with the repository server. (Synchronous type)

1.21.3.1 ldap_simple_bind()

Name

ldap_simple_bind

Synopsis

```
#include "idldap.h"
int  ldap_simple_bind(
        LDAP  *ld,
        const char  *dn,
        const char  *passwd );
```

Description

This function makes a request for asynchronous type simple authentication to the repository server.

For anonymous access, specify NULL in the "dn" and "passwd" parameters. If BIND is omitted, all requests to the server are processed anonymously.

To receive "result" information for a BIND operation, use ldap_result() and specify the message ID that is returned as the ldap_simple_bind() return value. For details, refer to Receiving/Determining Processing Results.

If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the address of the DN for BIND. If NULL is specified, the BIND in the repository server is anonymous.

passwd

Specify the address of the password for BIND.

Return Values

- For normal termination, the message ID of the BIND operation is returned as the return value.
- For abnormal termination, "-1" is returned as the return value.

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

1.21.3.2 ldap_simple_bind_s()

Name

ldap_simple_bind_s

Synopsis

```
#include "idldap.h"
int  ldap_simple_bind_s(
        LDAP  *ld,
        const char  *dn,
        const char  *passwd );
```

Description

This function makes a request for synchronous type simple authentication to the repository server.

For anonymous access, specify NULL in the "dn" and "passwd" parameters. If BIND is omitted, all requests to the server are processed anonymously.

If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the address of the DN for BIND. If NULL is specified, the BIND to the server is performed anonymously.

passwd

Specify the address of the password for BIND.

Return Values

This function returns the LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: A non-LDAP_SUCCESS LDAP error code

1.21.4 Interface for Entry Search

Function	Function Explanation
ldap_search()	Searches for the specified condition entry. (Asynchronous type)
ldap_search_s()	Searches for the specified condition entry. (Synchronous type)
ldap_search_st()	Searches for the specified condition entry. (Synchronous type, with time limit)
ldap_search_ext()	Searches for the specified condition entry. (Asynchronous type, LDAP V3 extended function)
ldap_search_ext_s()	Searches for the specified condition entry. (Synchronous type, LDAP V3 extended function)

The maximum searchable entry number and search timeout time values may be invalid, depending on the value set for the DN accessed from the client (the bind DN) and the maximum searchable entry number and search timeout time in the server.

The maximum searchable entry number and search timeout time can be specified in the Directory Service server and client.

The relationship between the values specified in the Directory Service server and client is shown in the following table.

Table 1.7 Relationship between the Values Specified in the Directory Service Server and Client

DN accessed from the client (the bind DN)	Value specified in the client	Relationship between values specified in the server and the client	Which of the values is enabled
Admin DN	Yes	server>client	The value specified in the client
	Yes	server<=client	The value specified in the client
	Unlimited	-	The value specified in the client
Other	Yes	server>client	The value specified in the client
	Yes	server<=client	The value specified in the server
	Unlimited	-	The value specified in the server

1.21.4.1 ldap_search()

Name

ldap_search

Synopsis

```
#include "idldap.h"
int  ldap_search(
        LDAP  *ld,
        const char  *base,
        int  scope,
        const char  *filter,
```

```
        char  **attrs,
        int  attrsonly );
```

Description

This function executes asynchronous type search processing for entries.

To receive the asynchronous type search processing result, use ldap_result(), and specify the message ID that is returned as the ldap_simple_bind() return value. For details, refer to Receiving/Determining Processing Results.

Execution of the search is controlled by the following session handle options:

- LDAP_OPT_SIZELIMIT: Maximum number of entries that can be received
- LDAP_OPT_TIMELIMIT: Maximum search time (in seconds)

For session handle option details, refer to Session Handle Option Settings/Reference. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

base

Specify the address of the DN for the search base. 'Search base' is the start position of the search for the entry.

scope

Specify one of the following values:

- LDAP_SCOPE_BASE: Searches for entries specified in the search base.
- LDAP_SCOPE_ONELEVEL: Searches for entries one level under the entry specified in the search base.
- LDAP_SCOPE_SUBTREE: Searches for entries specified in the search base (and all of the entries under those entries).

filter

Specify the address of the search filter. If NULL is specified, it is as if "(objectClass=*)" was set. For details of search filter grammar, refer to "Search Filters" in the Directory Service Operator's Guide.

Example

```
(cn=User1)
(o=Fujitsu*)
(&(objectClass=person)(|(uid=0001)(cn=User*)))
```

attrs

Specify the address of the pointer array for setting the address of the attribute to be notified. The repository server only notifies the client of the attribute specified in the pointer array. In the example shown below, only "objectClass" and "commonName" are notified.

Example

```
char  *attrs[3] = {
            "objectClass",
            "commonName",
            NULL
            };
```

To notify only the client of the DN, set "LDAP_NO_ATTRS" at the start of the array in the "attrs" parameter.

Example

```
char  *attrs[2] = {
            LDAP_NO_ATTRS,
```

```
            NULL
            };
```

If NULL is specified in this parameter, the client is notified of all of the usable attributes for entries that were found in the search. NULL must be set at the end of the pointer array.

attrsonly

Specify either of the following values:

- To notify the client of the attribute and attribute value: "0"
- To notify the client of only the attribute: Any value except "0"

Return Values

This function returns the following values:

- For normal termination: Message ID
- For abnormal termination: "-1"

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

1.21.4.2 ldap_search_s()

Name

ldap_search_s

Synopsis

```
#include "idldap.h"
int  ldap_search_s(
        LDAP  *ld,
        const char  *base,
        int  scope,
        const char  *filter,
        char  **attrs,
        int  attrsonly,
        LDAPMessage  **res );
```

Description

This function executes synchronous type search processing for entries with no time limit.

Execution of the search is controlled by the following session handle options:

- LDAP_OPT_SIZELIMIT: Maximum number of entries that can be received
- LDAP_OPT_TIMELIMIT: Maximum search time (in seconds)

For session handle option details, refer to Session Handle Option Settings/Reference.

Use the syntax analysis function to analyze the notified search result. For details, refer to 1.21.14 Interface for Processing Entry Lists, 1.21.15 Interface for Reading Attributes and 1.21.16 Interface for Reading Attribute Values. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

base

Specify the address of the DN for the search base. "Search base" is the start position of the search for the entry.

scope

Specify one of the following values:

- LDAP_SCOPE_BASE: Searches for entries specified in the search base.
- LDAP_SCOPE_ONELEVEL: Searches for entries in the first level of the search base.
- LDAP_SCOPE_SUBTREE: Searches for entries specified in the search base, and all of the entries under those entries.

filter

Specify the address of the search filter. If NULL is specified, it is as if "(objectClass=*)" was set. For details of search filter grammar, refer to "Search Filters" in the Directory Service Operator's Guide.

Example

```
(cn=User1)
(o=Fujitsu*)
(&(objectClass=person)(|(uid=0001)(cn=User*)))
```

attrs

Specify the address of the pointer array for setting the address of the attribute to be notified. The repository server only notifies the client of the attribute specified in the pointer array. In the following example, only "objectClass" and "commonName" are notified.

Example

```
char  *attrs[3] = {
            "objectClass",
            "commonName",
            NULL
            };
```

To only notify the client of the DN, set "LDAP_NO_ATTRS" at the start of the array in the "attrs" parameter.

Example

```
char  *attrs[2] = {
            LDAP_NO_ATTRS,
            NULL
            };
```

If NULL is specified in this parameter, the client is notified of all of the usable attributes for entries that were found in the search. NULL must be set at the end of the pointer array.

attrsonly

Specify either of the following values:

- To notify the client of the attribute and the attribute value: "0"
- To only notify the client of the attribute: Any value except "0"

res

Specify the address of the "LDAPMessage *" type pointer variable for storing the "result" information address.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: A non-LDAP_SUCCESS LDAP error code

If the client is notified of "result" information from the server, the address of the "result" information is set in the pointer variable of the specified "res" parameter.

Notes

- Free the notified area

  The "result" information and search result notified in the "res" parameter must be freed using ldap_msgfree() once they are no longer required. For details, refer to 1.21.18 Interface for Releasing Dynamic Memory.

1.21.4.3 ldap_search_st()

Name

ldap_search_st

Synopsis

```
#include "idldap.h"
int  ldap_search_st(
        LDAP  *ld,
        const char  *base,
        int  scope,
        const char  *filter,
        char  **attrs,
        int  attrsonly,
        struct timeval  *timeout,
        LDAPMessage  **res );
```

Description

This function executes entry search processing for synchronous types with a time limit.

Execution of the search is controlled by the following session handles:

- LDAP_OPT_SIZELIMIT: Maximum number of entries that can be received
- LDAP_OPT_TIMELIMIT: Maximum search time (in seconds)

For session handle option details, refer to Session Handle Option Settings/Reference.

Use the syntax analysis function to analyze the notified search result. For details, refer to 1.21.14 Interface for Processing Entry Lists, 1.21.15 Interface for Reading Attributes and 1.21.16 Interface for Reading Attribute Values. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

base

Specify the address of the DN for the search base. "Search base" is the start position of the search for the entry.

scope

Specify one of the following values:

- LDAP_SCOPE_BASE: Searches for entries specified in the search base.
- LDAP_SCOPE_ONELEVEL: Searches for entries in the first level of the search base.
- LDAP_SCOPE_SUBTREE: Searches for entries specified in the search base, and all of the entries under those entries.

filter

Specify the address of the search filter. If NULL is specified, it is as if "(objectClass=*)" was set. For details of search filter grammar, refer to "Search Filters" in the Directory Service Operator's Guide.

Example

```
(cn=User1)
(o=Fujitsu*)
(&(objectClass=person)(|(uid=0001)(cn=User*)))
```

attrs

Specify the address of the pointer array for setting the address of the attribute to be notified. The repository server only notifies the client of the attribute specified in the pointer array. In the following example, only "objectClass" and "commonName" are notified.

Example

```
char  *attrs[3] = {
            "objectClass",
            "commonName",
            NULL
            };
```

To only notify the client of the DN, set "LDAP_NO_ATTRS" at the start of the array in the "attrs" parameter.

Example

```
char  *attrs[2] = {
            LDAP_NO_ATTRS,
            NULL
            };
```

If NULL is specified in this parameter, the client is notified of all the usable attributes for entries that were found in the search. NULL must be set at the end of the pointer array.

attrsonly

Specify either of the following values:

- To notify the client of the attribute and the attribute value: 0
- To only notify the client of the attribute: Any value except "0"

timeout

Specify the address of the timeval Structure for setting the local timeout value for the search.

Set the following value as the timeval structure member:

- tv_sec: Timer value (seconds)
- tv_usec: Timer value (microseconds)

"0" cannot be specified for both tv_sec and tv_usec at the same time in the timeval structure.

This value is used as the maximum wait time for receiving the search result from the repository server in the client. The relationship between the "timeout" parameter and the Session Handle Options (LDAP_OPT_TIMELIMIT) is shown in the following table.

Table 1.8 Relationship between 'timeout' and (LDAP_OPT_TIMELIMIT)

Value specified for the "timeout" parameter	Value specified for the session handle option (LDAP_OPT_TIMELIMIT)	Enabled values
NULL	LDAP_NO_LIMIT (default value)	Value specified for the session handle option (unlimited)
	Timer	Value specified for the session handle option
Timer	LDAP_NO_LIMIT (default value)	Value specified for the "timeout" parameter
	Timer	The lower of the values specified for the session handle option and the "timeout" parameter

res

Specify the address of the "LDAPMessage *" type pointer variable for storing the "result" information address.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: A non-LDAP_SUCCESS LDAP error code

If the client is notified of "result" information from the server, the address of the "result" information is set in the pointer variable of the specified "res" parameter.

Notes

- Free the notified area
- The "result" information and search result notified in the "res" parameter must be unbound using ldap_msgfree() once they are no longer required. For details, refer to 1.21.18 Interface for Releasing Dynamic Memory.

1.21.4.4 ldap_search_ext()

Name

ldap_search_ext

Synopsis

```
#include "idldap.h"
int  ldap_search_ext(
        LDAP  *ld,
        const char  *base,
        int  scope,
        const char  *filter,
        char  **attrs,
        int  attrsonly,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls,
        struct timeval  *timeout,
        int  sizelimit,
        int  *msgidp );
```

Description

This function executes synchronous type search processing for entries.

To receive the asynchronous type search processing result, use ldap_result() and specify the message ID that is returned as the ldap_search_ext() return value. For details, refer to Receiving/Determining Processing Results.

Execution of the search is controlled by the following session handle options:

- LDAP_OPT_SIZELIMIT: Maximum number of entries that can be received
- LDAP_OPT_TIMELIMIT: Maximum search time (in seconds)

For session handle option details, refer to Session Handle Option Settings/Reference.

Use the syntax analysis function to analyze the notified search result. For details, refer to 1.21.14 Interface for Processing Entry Lists, 1.21.15 Interface for Reading Attributes and 1.21.16 Interface for Reading Attribute Values. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

base

Specify the address of the DN for the search base. "Search base" is the start position of the search for the entry.

scope

Specify one of the following values:

- LDAP_SCOPE_BASE: Searches for entries specified in the search base.
- LDAP_SCOPE_ONELEVEL: Searches for entries in the first level of the search base.
- LDAP_SCOPE_SUBTREE: Searches for entries specified in the search base, and all of the entries under those entries.

filter

Specify the address of the search filter. If NULL is specified, it is as if "(objectClass=*)" was set. For details of search filter grammar, refer to "Search Filters" in the Directory Service Operator's Guide.

Example

```
(cn=User1)
(o=Fujitsu*)
(&(objectClass=person)(|(uid=0001)(cn=User*)))
```

attrs

Specify the address of the pointer array for setting the address of the attribute to be notified. The repository server only notifies the client of the attribute specified in the pointer array. In the following example, only "objectClass" and "commonName" are notified.

Example

```
char  *attrs[3] = {
            "objectClass",
            "commonName",
            NULL
            };
```

To only notify the client of the DN, set "LDAP_NO_ATTRS" at the start of the array in the "attrs" parameter.

Example

```
char  *attrs[2] = {
            LDAP_NO_ATTRS,
            NULL
            };
```

If NULL is specified in this parameter, the client is notified of all of the usable attributes for entries that were found in the search. NULL must be set at the end of the pointer array.

attrsonly

Specify either of the following values:

- To notify the client of the attribute and the attribute value: 0
- To only notify the client of the attribute: Any value except "0"

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

timeout

Specify the address of the timeval Structure for setting the local timeout value for the search.

Set the following value as the timeval structure member.

- tv_sec: Timer value (seconds)

- tv_usec: Timer value (microseconds)

"0" cannot be specified for tv_sec and tv_usec in the timeval structure.

If the NULL pointer is specified in the "timeout" parameter, the value set for the Session Handle Options (LDAP_OPT_TIMELIMIT) is used. This value is used as the maximum wait time for receiving the search result from the repository server in the client. The relationship between the "timeout" parameter and the Session Handle Options (LDAP_OPT_TIMELIMIT) is shown in the following table.

Table 1.9 Relationship between 'timeout' (LDAP_OPT_TIMELIMIT)

Value specified for the "timeout" parameter	Value specified for the session handle option (LDAP_OPT_TIMELIMIT)	Enabled values
NULL	LDAP_NO_LIMIT (default value)	Value specified for the session handle option (unlimited)
	Timer	Value specified for the session handle option
Timer	LDAP_NO_LIMIT (default value)	Value specified for the "timeout" parameter
	Timer	Value specified for the "timeout" parameter

sizelimit

Specify the maximum number of entries that can be received. To make the search result unlimited, specify LDAP_NO_LIMIT. If LDAP_DEFAULT_SIZELIMIT is specified, the value set for the Session Handle Options (LDAP_OPT_SIZELIMIT) is used. The relationship between the 'sizelimit' parameter and the session handle option (LDAP_OPT_SIZELIMIT) is shown in the following table.

Table 1.10 Relationship between 'sizelimit' and the Session Handle Option (LDAP_OPT_SIZELIMIT)

Value specified for the "sizelimit" parameter	Value specified for the session handle option (LDAP_OPT_SIZELIMIT)	Enabled values
LDAP_NO_LIMIT	LDAP_NO_LIMIT (default value)	Value specified for the "sizelimit" parameter (unlimited)
	Size	Value specified for the "sizelimit" parameter (unlimited)
Timer	LDAP_NO_LIMIT (default value)	Value specified for the "sizelimit" parameter
	Size	Value specified for the "sizelimit" parameter
LDAP_DEFAULT_SIZELIMIT	LDAP_NO_LIMIT (default value)	Value specified for the session handle option (unlimited)
	Size	Value specified for the session handle option

msgidp

Specify the address of the variable for storing the message ID.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: A non-LDAP_SUCCESS LDAP error code

If the return of the value terminated normally, the message ID is set in the variable shown in "msgidp".

1.21.4.5 ldap_search_ext_s()

Name

ldap_search_ext_s

Synopsis

```
#include "idldap.h"
int  ldap_search_ext_s(
        LDAP  *ld,
        const char  *base,
        int  scope,
        const char  *filter,
        char  **attrs,
        int  attrsonly,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls,
        struct timeval  *timeout,
        int  sizelimit,
        LDAPMessage  **res );
```

Description

This function executes synchronous type search processing for entries.

The execution of the search is controlled by the following session handle options:

- LDAP_OPT_SIZELIMIT: Maximum number of entries that can be received
- LDAP_OPT_TIMELIMIT: Maximum search time (in seconds)

For session handle option details, refer to Session Handle Option Settings/Reference.

Use the syntax analysis function to analyze the notified search result. For details, refer to 1.21.14 Interface for Processing Entry Lists, 1.21.15 Interface for Reading Attributes and 1.21.16 Interface for Reading Attribute Values. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

base

Specify the address of the DN for the search base. "Search base" is the start position of the search for the entry.

scope

Specify one of the following values:

- LDAP_SCOPE_BASE: Searches for entries specified in the search base.
- LDAP_SCOPE_ONELEVEL: Searches for entries in the first level of the search base.
- LDAP_SCOPE_SUBTREE: Searches for entries specified in the search base, and all of the entries under those entries.

filter

Specify the address of the search filter. If NULL is specified, it is as if "(objectClass=*)" was set. For details of search filter grammar, refer to "Search Filters" in the Directory Service Operator's Guide.

Example

```
cn=User1)
o=Fujitsu*)
&(objectClass=person)(|(uid=0001)(cn=User*)))
```

attrs

Specify the address of the pointer array for setting the address of the attribute to be notified. The repository server only notifies the client of the attribute specified in the pointer array. In the following example, only "objectClass" and "commonName" are notified.

Example

```
char  *attrs[3] = {
            "objectClass",
            "commonName",
            NULL
            };
```

To only notify the client of the DN, set "LDAP_NO_ATTRS" at the start of the array in the "attrs" parameter.

Example

```
char  *attrs[2] = {
            LDAP_NO_ATTRS,
            NULL
            };
```

If NULL is specified in this parameter, the client is notified of all of the usable attributes for entries that were found in the search. NULL must be set at the end of the pointer array.

attrsonly

Specify either of the following values:

- To notify the client of the attribute and the attribute value: 0
- To only notify the client of the attribute: Any value except "0"

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

timeout

Specify the address of the timeval Structure for setting the local timeout value for the search. Set the value shown below as the timeval structure member.

- tv_sec: Timer value (seconds)
- tv_usec: Timer value (microseconds)

"0" cannot be specified for tv_sec and tv_usec in the timeval structure.

If the NULL pointer is specified in the "timeout" parameter, the value set for the Session Handle Options (LDAP_OPT_TIMELIMIT) is used. This value is used as the maximum wait time for receiving the search result from the repository server in the client. The relationship between the "timeout" parameter and the Session Handle Options (LDAP_OPT_TIMELIMIT) is shown in the following table.

Table 1.11 Relationship between 'timeout' and (LDAP_OPT_TIMELIMIT)

Value specified for the "timeout" parameter	Value specified for the session handle option (LDAP_OPT_TIMELIMIT)	Enabled values
NULL	LDAP_NO_LIMIT (default value)	Value specified for the session handle option (unlimited)
	Timer	Value specified for the session handle option
Timer	LDAP_NO_LIMIT (default value)	Value specified for the "timeout" parameter
	Timer	Value specified for the "timeout" parameter

sizelimit

Specify the maximum number of entries that can be received. To make the search result unlimited, specify LDAP_NO_LIMIT. If LDAP_DEFAULT_SIZELIMIT is specified, the value set for the Session Handle Options (LDAP_OPT_SIZELIMIT) is used. The relationship between the 'sizelimit' parameter and the session handle option (LDAP_OPT_SIZELIMIT) is shown in the following table.

Table 1.12 Relationship between 'sizelimit' and (LDAP_OPT_TIMELIMIT)

Value specified for the "sizelimit" parameter	Value specified for the session handle option (LDAP_OPT_SIZELIMIT)	Enabled values
LDAP_NO_LIMIT	LDAP_NO_LIMIT (default value)	Value specified for the "sizelimit" parameter (unlimited)
	Size	Value specified for the "sizelimit" parameter (unlimited)
Size	LDAP_NO_LIMIT (default value)	Value specified for the "sizelimit" parameter
	Size	Value specified for the "sizelimit" parameter
LDAP_DEFAULT_SIZELIMIT	LDAP_NO_LIMIT (default value)	Value specified for the session handle option (unlimited)
	Size	Value specified for the session handle option

res

Specify the address of the "LDAPMessage *" type pointer variable for storing the "result" information address.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: A non-LDAP_SUCCESS LDAP error code

If the client is notified of "result" information from the server, the address of the "result" information is set in the pointer variable of the specified "res" parameter.

Notes

- Free the notified area

  The "result" information and search result notified in the "res" parameter must be unbound using ldap_msgfree() once they are no longer required. For details, refer to 1.21.18 Interface for Releasing Dynamic Memory.

1.21.5 Interface for Comparing Attribute Values

Function	Explanation
ldap_compare()	Compares the entry attribute value with the specified value. (Asynchronous type)
ldap_compare_s()	Compares the entry attribute value with the specified value. (Synchronous type)
ldap_compare_ext()	Compares the entry attribute value with the specified value. (Asynchronous type, LDAP V3 extended function)
ldap_compare_ext_s()	Compares the entry attribute value with the specified value. (Synchronous type, LDAP V3 extended function)

1.21.5.1 ldap_compare()

Name

ldap_compare

Synopsis

```
#include "idldap.h"
int  ldap_compare(
        LDAP  *ld,
        const char  *dn,
        const char  *attr,
        const char  *value );
```

Description

This function executes asynchronous type compare processing for attribute values. Binary data cannot be compared.

To receive the asynchronous type search processing result, use ldap_result() and specify the message ID that is returned as the ldap_compare() return value. For details, refer to Receiving/Determining Processing Results. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entries to be compared.

attr

Specify the address of the attributes to be compared.

value

Specify the address for the compared string data.

Return Values

This function returns the following values:

- For normal termination: Message ID
- For abnormal termination: "-1"

  For details of the cause of an error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

1.21.5.2 ldap_compare_s()

Name

ldap_compare_s

Synopsis

```
#include "idldap.h"
int  ldap_compare_s(
        LDAP  *ld,
        const char  *dn,
        const char  *attr,
        const char  *value );
```

Description

This function executes synchronous type compare processing for attribute values. Binary data cannot be compared. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entries to be compared.

attr

Specify the address of the attributes to be compared.

value

Specify the address for the compared string data.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- LDAP_COMPARE_TRUE (For normal termination): Values specified in the "value" parameter are included in the specified attribute.
- LDAP_COMPARE_FALSE (For normal termination): Values specified in the "value" parameter are not included in the specified attribute.
- For abnormal termination: An LDAP error code not mentioned above is notified.

1.21.5.3 ldap_compare_ext()

Name

ldap_compare_ext

Synopsis

```
#include "idldap.h"
int  ldap_compare_ext(
        LDAP  *ld,
        const char  *dn,
        const char  *attr,
        const struct berval  *bvalue,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls,
        int  *msgidp );
```

Description

This function executes asynchronous type compare processing for attribute values. Binary data cannot be compared.

To receive the asynchronous type search processing result, use ldap_result() and specify the message ID that is returned as the ldap_compare_ext() return value. For details, refer to Receiving/Determining Processing Results. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entries to be compared.

attr

Specify the address of the attributes to be compared.

bvalue

Specify the address of the berval Structure for setting the data to be compared.

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

msgidp

Specify the address of the variable for storing the message ID.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

If the return of the value was terminated normally, the message ID is set in the variable specified in the "msgidp" parameter.

1.21.5.4 ldap_compare_ext_s()

Name

ldap_compare_ext_s

Synopsis

```
#include "idldap.h"
int  ldap_compare_ext_s(
        LDAP  *ld,
        const char  *dn,
        const char  *attr,
        const struct berval  *bvalue,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls );
```

Description

This function executes synchronous type compare processing for attribute values. Binary data cannot be compared. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entries to be compared.

attr

Specify the address of the attributes to be compared.

bvalue

Specify the address of the berval Structure for setting the data to be compared.

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- LDAP_COMPARE_TRUE (For normal termination): Values specified in the "bvalue" parameter are included in the specified attribute.
- LDAP_COMPARE_FALSE (For normal termination): Values specified in the "bvalue" parameter are not included in the specified attribute.
- For abnormal termination: An LDAP error code not mentioned above was notified.

1.21.6 Interface for Changing Entries

Function	Explanation
ldap_modify()	Updates the specified entry. (Asynchronous type)
ldap_modify_s()	Updates the specified entry. (Synchronous type)
ldap_modify_ext()	Updates the specified entry. (Asynchronous type, LDAP V3 extended function)
ldap_modify_ext_s()	Updates the specified entry. (Synchronous type, LDAP V3 extended function)

1.21.6.1 ldap_modify()

Name

ldap_modify

Synopsis

```
#include "idldap.h"
int  ldap_modify(
        LDAP  *ld,
        const char  *dn,
        LDAPMod  **mods );
```

Description

This function executes entry change processing for asynchronous types.

To receive the asynchronous type search processing result, use ldap_result() and specify the message ID that is returned as the ldap_modify() return value. For details, refer to Receiving/Determining Processing Results. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entry to be changed.

mods

Specify the address of the LDAPMod Structure pointer array. The pointer array specified here must have NULL set in the last option. Set the information to be changed for each attribute in each LDAPMod structure. The values that can be set for each LDAPMod structure are shown below.

- mod_op

  Specify the attribute value type, and the logical OR operation for the change operation type.

  - Attribute value type
  - If the attribute value is string data: "0"
  - If the attribute value is binary data: LDAP_MOD_BVALUES
  - Change operation type
  - To add an attribute value: LDAP_MOD_ADD
  - To delete an attribute value: LDAP_MOD_DELETE
  - To replace an attribute value: LDAP_MOD_REPLACE

- mod_type

  Specify the address of the attribute to be changed. Some of the attributes that are specified must contain ";binary" in the attribute name.

  - To add an attribute value: Specify the attribute value to be added.
  - To delete an attribute value: Specify the attribute value to be deleted.
  - To replace an attribute value: Specify the attribute value to be replaced.

- mod_vals

  Specify the address of the attribute to be added/deleted/replaced. The attribute value specified here is the shared mod_vals pointer array. This pointer array must have NULL set in the last option.

  - To add an attribute value: Specify the address of the pointer array for the attribute value to be added.
  - To delete an attribute value: Specify the address of the pointer array for the attribute value to be deleted. To delete all the attribute values specified in the "mod_type" parameter, specify NULL.
  - To replace an attribute value: Specify the address of the pointer array for the attribute value to be replaced.

  The shared "mod_vals" member that is used depends on the attribute value type specified in the "mod_op" parameter, as shown below.

  - String data: Specify the address of the pointer array for the attribute value in "mod_values". Each attribute value must have NULL set in the last option.
  - Binary data: Specify the address of the berval Structure pointer array in "mod_bvalues". Specify the data address and length of the attribute value in the berval structure.

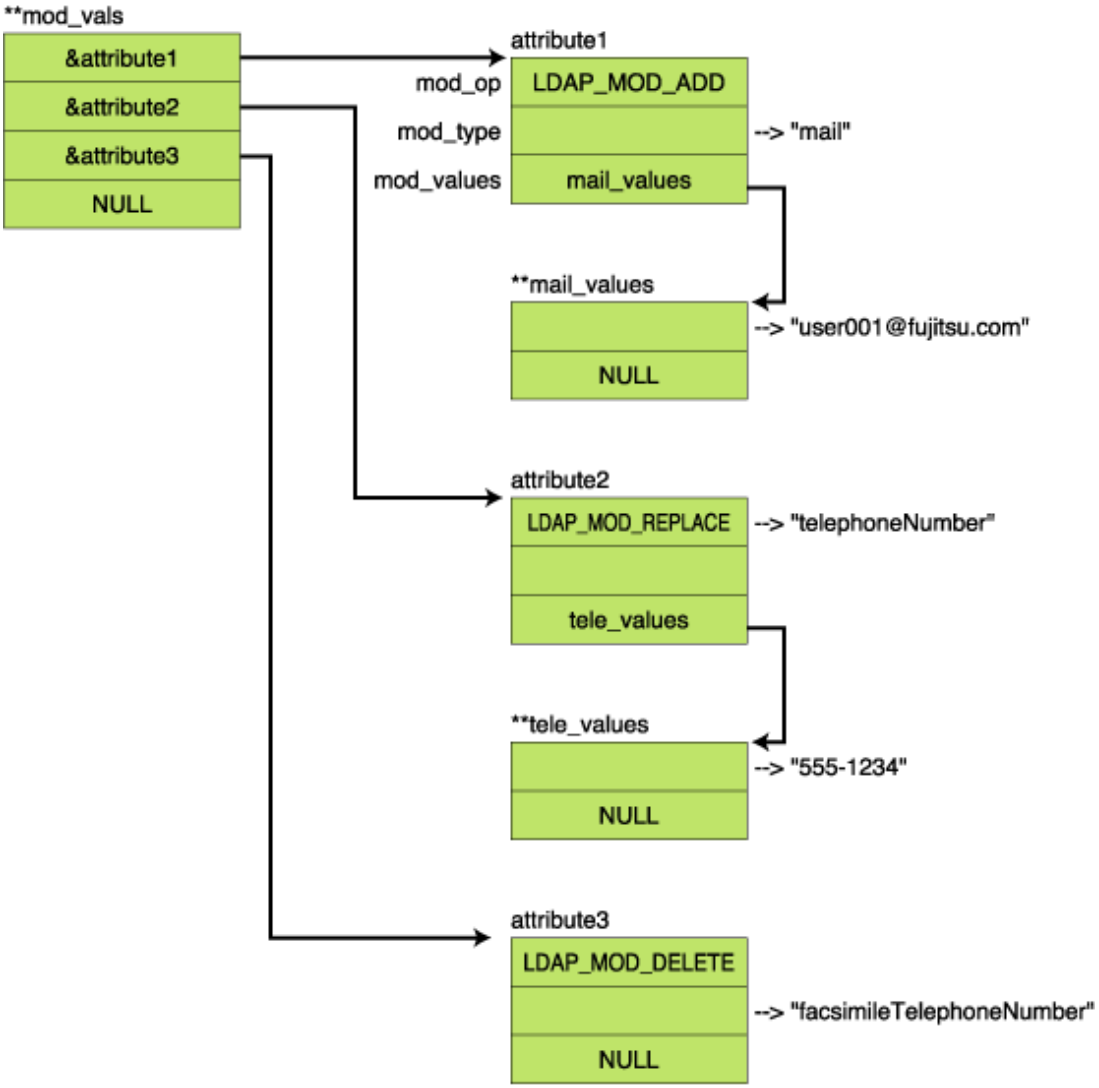
  The pointer array of each attribute value must have NULL set in the last option.

The parameter configuration in "modify" processing

The following example shows the parameters for changing string data in "modify".

- mail --> Add
- telephoneNumber --> Change
- facsimileTelephoneNumber --> Delete

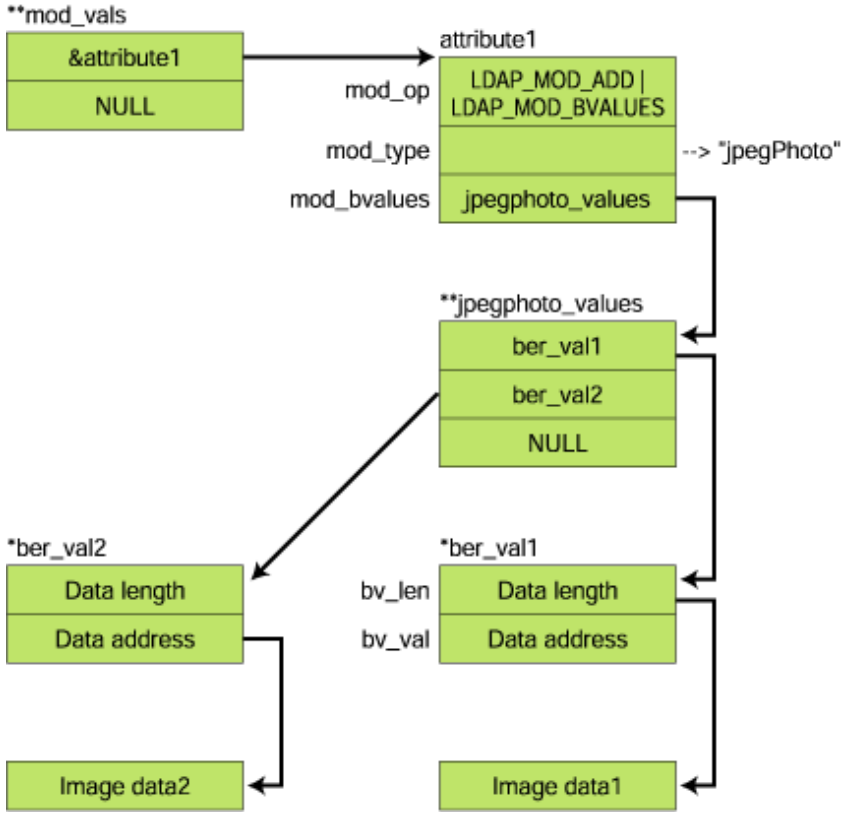
Figure 1.1 Parameters for Changing String Data in 'modify'



The following example shows the parameters for changing binary data in "modify".

- jpegPhoto --> Add

Figure 1.2 Parameters for Changing Binary Data in 'modify'



Return Values

This function returns the following values:

- For normal termination: Message ID
- For abnormal termination: "-1"

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

1.21.6.2 ldap_modify_s()

Name

ldap_modify_s

Synopsis

```
#include "idldap.h"
int  ldap_modify_s(
        LDAP  *ld,        const char  *dn,
        LDAPMod  **mods );
```

Description

This function executes entry change processing for synchronous types.

If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entry to be changed.

mods

Specify the address of the LDAPMod Structure pointer array. The pointer array specified here must have NULL set in the last option. Set the information to be changed for each attribute in each LDAPMod structure. The values that can be set for each LDAPMod structure are shown below.

- mod_op

  Specify the attribute value type, and the logical OR operation for the change operation type.

  - Attribute value type
  - If the attribute value is string data: "0"
  - If the attribute value is binary data: LDAP_MOD_BVALUES
  - Change operation type
  - To add an attribute value: LDAP_MOD_ADD
  - To delete an attribute value: LDAP_MOD_DELETE
  - To replace an attribute value: LDAP_MOD_REPLACE

- mod_type

  Specify the address of the attribute to be changed. Some of the attributes that are specified must contain ";binary" in the attribute name.

  - To add an attribute value: Specify the attribute value to be added.
  - To delete an attribute value: Specify the attribute value to be deleted.
  - To replace an attribute value: Specify the attribute value to be replaced.

- mod_vals

  Specify the address of the attribute to be added/deleted/replaced. The attribute value specified here is the shared mod_vals pointer array. This pointer array must have NULL set in the last option.

  - To add an attribute value: Specify the address of the pointer array for the attribute value to be added.
  - To delete an attribute value: Specify the address of the pointer array for the attribute value to be deleted. To delete all of the attribute values specified in the "mod_type" parameter, specify NULL.
  - To replace an attribute value: Specify the address of the pointer array for the attribute value to be replaced.

  The shared "mod_vals" member that is used depends on the attribute value type specified in the "mod_op" parameter, as shown below.

  - String data: Specify the address of the pointer array for the attribute value in "mod_values".
  - Binary data: Specify the address of the berval Structure pointer array in "mod_bvalues". Specify the attribute value address and size in the berval structure.

  The pointer array of each attribute value must have NULL set in the last option.

For examples of the parameters used for changing data, refer to 'The parameter configuration in "modify" processing'.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

1.21.6.3 ldap_modify_ext()

Name

ldap_modify_ext

Synopsis

```
#include "idldap.h"
int  ldap_modify_ext(
        LDAP  *ld,
        const char  *dn,
        LDAPMod  **mods,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls,
        int  *msgidp );
```

Description

This function executes entry change processing for asynchronous types.

To receive the asynchronous function processing result, use ldap_result() and specify the message ID that is returned as ldap_modify_ext() return value. For details, refer to Receiving/Determining Processing Results. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN of the entry to be changed.

mods

Specify the address of the LDAPMod Structure pointer array. The pointer array specified here must have NULL set in the last option. Set the information to be changed for each attribute in each LDAPMod structure. The values that can be set for each LDAPMod structure are shown below.

- mod_op

  Specify the attribute value type, and the logical OR operation for the change operation type.

  - attribute value type
  - If the attribute value is string data: "0"
  - If the attribute value is binary data: LDAP_MOD_BVALUES
  - change operation type
  - To add an attribute value: LDAP_MOD_ADD
  - To delete an attribute value: LDAP_MOD_DELETE
  - To replace an attribute value: LDAP_MOD_REPLACE

- mod_type

  Specify the address of the attribute to be changed. Some of the attributes that are specified must contain ";binary" in the attribute name.

  - To add an attribute value: Specify the attribute value to be added.

- To delete an attribute value: Specify the attribute value to be deleted.
- To replace an attribute value: Specify the attribute value to be replaced.

- mod_vals

  Specify the address of the attribute to be added/deleted/replaced. The attribute value specified here is the shared mod_vals pointer array. This pointer array must have NULL set in the last option.

  - To add an attribute value: Specify the address of the pointer array for the attribute value to be added.
  - To delete an attribute value: Specify the address of the pointer array for the attribute value to be deleted. To delete all of the attribute values specified in the "mod_type" parameter, specify NULL.
  - To replace an attribute value: Specify the address of the pointer array for the attribute value to be replaced.

  The shared "mod_vals" member that is used depends on the attribute value type specified in the "mod_op" parameter, as shown below.

  - String data: Specify the address of the pointer array for the attribute value in "mod_values". The pointer array of each attribute value must have NULL set in the last option.
  - Binary data: Specify the address of the berval Structure pointer array in "mod_bvalues". Specify the attribute value address and size in the berval structure.

  The pointer array of each attribute value must have NULL set in the last option.

For examples of the parameters used for changing data, refer to 'The parameter configuration in "add" processing'.

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

msgidp

Specify the address of the variable for storing the message ID.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

If the return of the value was terminated normally, the message ID is set in the variable specified in the "msgidp" parameter.

1.21.6.4 ldap_modify_ext_s()

Name

ldap_modify_ext_s

Synopsis

```
#include "idldap.h"
int  ldap_modify_ext_s(
        LDAP  *ld,
        const char  *dn,
        LDAPMod  **mods,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls );
```

Description

This function executes entry change processing for synchronous types.

If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN of the entry to be changed.

mods

Specify the address of the LDAPMod Structure pointer array. The pointer array specified here must have NULL set in the last option. Set the information to be changed for each attribute in each LDAPMod structure. The values that can be set for each LDAPMod structure are shown below.

- mod_op

  Specify the attribute value type, and the logical OR operation for the change operation type.

  - attribute value type
  - If the attribute value is string data: "0"
  - If the attribute value is binary data: LDAP_MOD_BVALUES
  - change operation type
  - To add an attribute value: LDAP_MOD_ADD
  - To delete an attribute value: LDAP_MOD_DELETE
  - To replace an attribute value: LDAP_MOD_REPLACE

- mod_type

  Specify the address of the attribute to be changed. Some of the attributes that are specified must contain ";binary" in the attribute name.

  - To add an attribute value: Specify the attribute value to be added.
  - To delete an attribute value: Specify the attribute value to be deleted.
  - To replace an attribute value: Specify the attribute value to be replaced.

- mod_vals

  Specify the address of the attribute to be added/deleted/replaced. The attribute value specified here is the shared mod_vals pointer array. This pointer array must have NULL set in the last option.

  - To add an attribute value: Specify the address of the pointer array for the attribute value to be added.
  - To delete an attribute value: Specify the address of the pointer array for the attribute value to be deleted. To delete all of the attribute values specified in the "mod_type" parameter, specify NULL.
  - To replace an attribute value: Specify the address of the pointer array for the attribute value to be replaced.

  The shared "mod_vals" member that is used depends on the attribute value type specified in the "mod_op" parameter, as shown below.

  - String data: Specify the address of the pointer array for the attribute value in "mod_values". The pointer array of each attribute value must have NULL set in the last option.
  - Binary data: Specify the address of the berval Structure pointer array in "mod_bvalues". Specify the attribute value address and size in the berval structure.

The pointer array of each attribute value must have NULL set in the last option.

For examples of the parameters used for changing data, refer to 'The parameter configuration in "modify" processing'.

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

1.21.7 Interface for Changing Entry Names

Function	Explanation
ldap_rename()	Changes the specified entry DN or RDN. (Asynchronous type, LDAP V3 extended function)
ldap_rename_s()	Changes the specified entry DN or RDN. (Synchronous type, LDAP V3 extended function)

1.21.7.1 ldap_rename()

Name

ldap_rename

Synopsis

```
#include "idldap.h"
int  ldap_rename(
        LDAP  *ld,
        const char  *dn,
        const char  *newrdn,
        const char  *newparent,
        int  deleteoldrdn,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls,
        int  *msgidp );
```

Description

This function changes entry names asynchronously.

To receive the asynchronous function processing result, use ldap_result() and specify the message ID that is returned as the ldap_rename() return value. For details, refer to Receiving/Determining Processing Results. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entry to be changed.

newrdn

Specify the new RDN address of this entry.

newparent

Specify the DN address of the parent of the new entry in this parameter. To put the new entry in the same location as the original entry, specify NULL in this parameter.

deleteoldrdn

Specify whether to delete the attribute value that corresponds to the original RDN from the entry in this parameter.

- To delete the attribute value that corresponds to the original RDN: Any value except "0"
- To not delete the attribute value that corresponds to the original RDN: "0"

If "0" is specified, the attribute value is left in the entry as a non-RDN attribute value.

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

msgidp

Specify the address of the variable for storing the message ID.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

If the return of the value was terminated normally, the message ID is set in the variable specified in the "msgidp" parameter.

1.21.7.2 ldap_rename_s()

Name

ldap_rename_s

Synopsis

```
#include "idldap.h"
int  ldap_rename_s(
        LDAP  *ld,
        const char  *dn,
        const char  *newrdn,
        const char  *newparent,
        int  deleteoldrdn,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls );
```

Description

This function changes entry names synchronously.

If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entry to be changed.

newrdn

Specify the new RDN address of this entry.

newparent

Specify the DN address of the parent of the new entry in this parameter. To put the new entry in the same location as the original entry, specify NULL in this parameter.

deleteoldrdn

Specify whether to delete the attribute value that corresponds to the original RDN from the entry in this parameter.

- To delete the attribute value that corresponds to the original RDN: Any value except "0"
- To not delete the attribute value that corresponds to the original RDN: "0"

If "0" is specified, the attribute value is left in the entry as a non-RDN attribute value.

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

1.21.8 Interface for Adding Entries

Function name	Function Explanation
ldap_add()	Adds the specified entry. (Asynchronous type)
ldap_add_s()	Adds the specified entry. (Synchronous type)
ldap_add_ext()	Adds the specified entry. (Asynchronous type, LDAP V3 extended function)
ldap_add_ext_s()	Adds the specified entry. (Synchronous type, LDAP V3 extended function)

1.21.8.1 ldap_add()

Name

ldap_add

Synopsis

```
#include "idldap.h"
int  ldap_add(
        LDAP  *ld,
        const char  *dn,
        LDAPMod  **attrs );
```

Description

This function adds entries asynchronously. If there is no parent, the specified entry cannot be added.

To receive the asynchronous function processing result, use ldap_result() and specify the message ID that is returned as the ldap_add() return value. For details, refer to Receiving/Determining Processing Results. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entry to be added.

attrs

Specify the address of the LDAPMod Structure pointer array. Set the attribute and attribute value to be added to the entry in each LDAPMod structure. The values that can be set for each option in the LDAPMod structure are shown below.

- mod_op

  Specify the attribute value type.

  - If the attribute value is string data: "0"
  - If the attribute value is binary data: LDAP_MOD_BVALUES

- mod_type

  Specify the address of the attribute to be changed. Some of the attributes that are specified must contain ";binary" in the attribute name.

- mod_vals

  Specify the address of the attribute value.

  The attribute value specified here is the shared mod_vals pointer array. This pointer array must have NULL set in the last option.

  - String data: Specify the address of the pointer array for the attribute value in "mod_values".
  - Binary data: Specify the address of the berval Structure pointer array in "mod_bvalues". Specify the attribute value address and size in the berval structure.

  The pointer array of each attribute value must have NULL set in the last option.

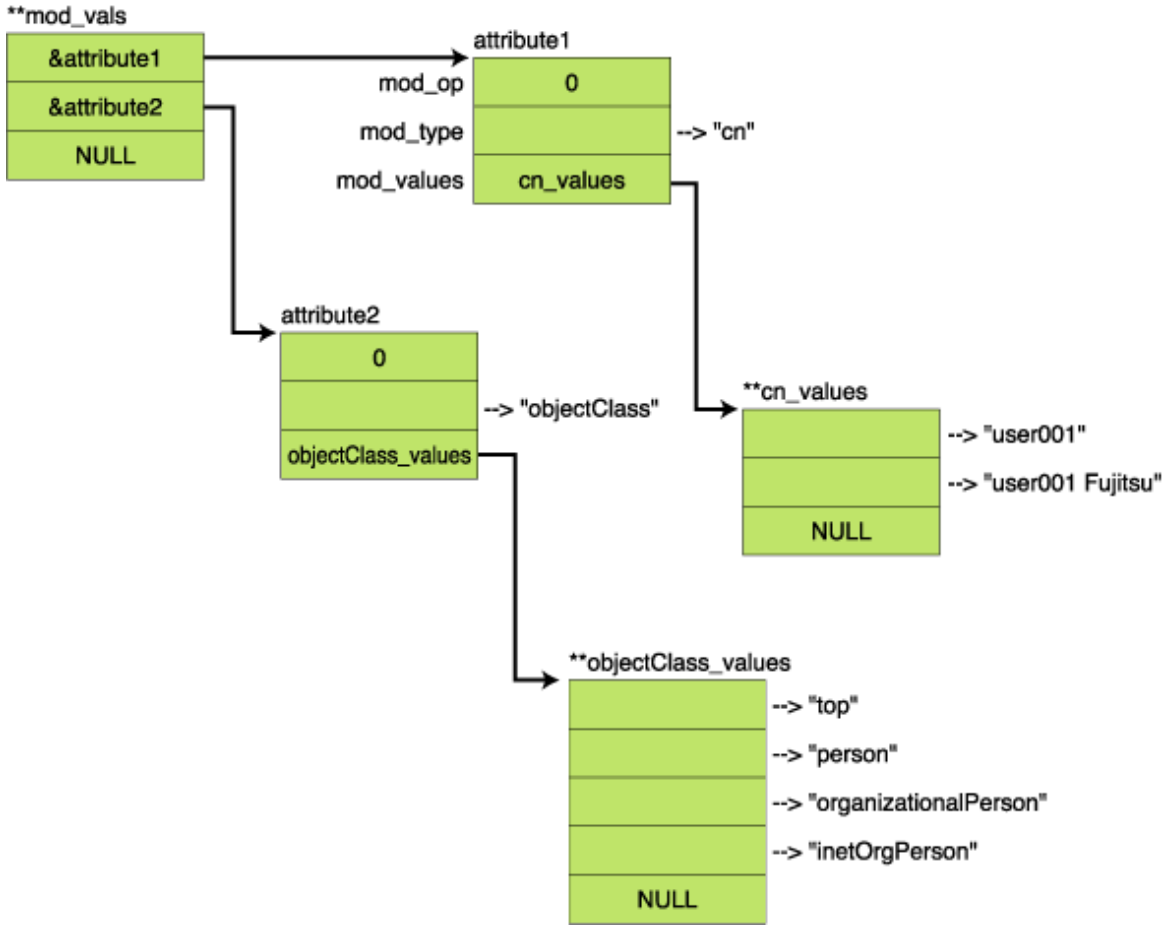
The following example shows the parameters for adding entries to a tree.

Entry configuration

```
cn: user001
cn: user001 Fujitsu
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
```

The parameter configuration in "add" processing

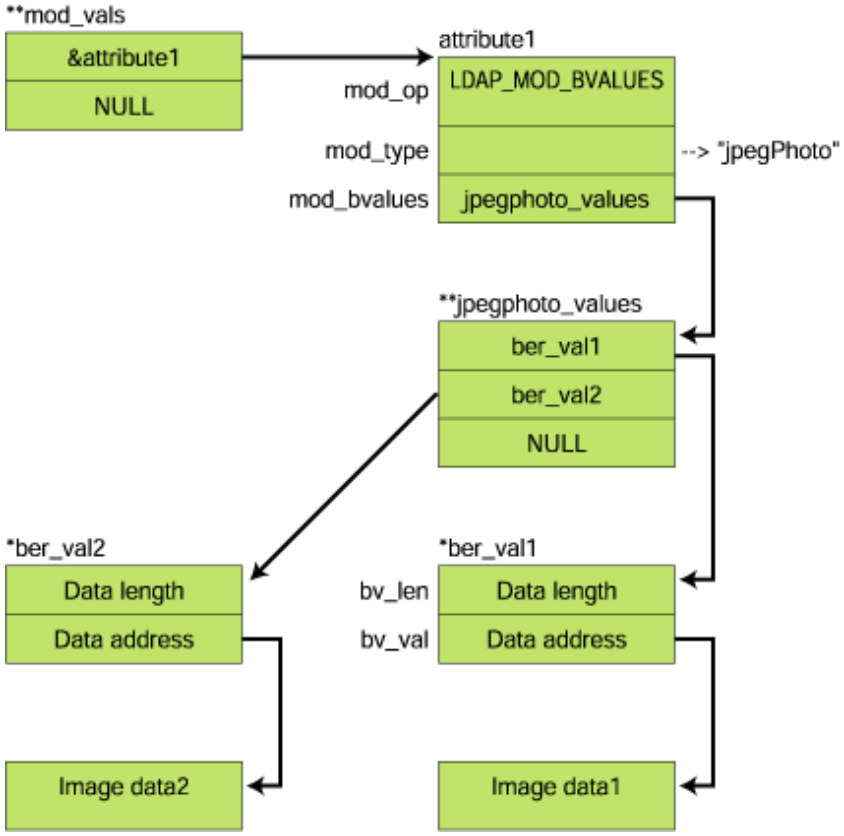
Figure 1.3 Parameters Configuration in 'add' Processing



The following example shows the parameters for adding binary data for attribute values.

- Add --> jpegPhoto

Figure 1.4 Parameters for Adding Binary Data for Attribute Values



Return Values

This function returns the following values:

- For normal termination: Message ID
- For abnormal termination: "-1"

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

1.21.8.2 ldap_add_s()

Name

ldap_add_s

Synopsis

```
#include "idldap.h"
int  ldap_add_s(
        LDAP  *ld,
        const char  *dn,
        LDAPMod  **attrs );
```

Description

This function adds entries synchronously. If there is no parent, the specified entry cannot be added. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entry to be added.

attrs

Specify the address of the LDAPMod Structure pointer array. Set the attribute and attribute value to be added to the entry in each LDAPMod structure. The values that can be set for each option in the LDAPMod structure are shown below.

- mod_op

  Specify the attribute value type.

  - If the attribute value is string data: "0"
  - If the attribute value is binary data: LDAP_MOD_BVALUES

- mod_type

  Specify the address of the attribute to be changed. Some of the attributes that are specified must contain ";binary" in the attribute name.

- mod_vals

  Specify the address of the attribute value.

  The attribute value specified here is the shared mod_vals pointer array. This pointer array must have NULL set in the last option. The shared "mod_vals" member that is used depends on the attribute value type specified in the "mod_op" parameter, as shown below.

  - String data: Specify the address of the pointer array for the attribute value in "mod_values".
  - Binary data: Specify the address of the berval Structure pointer array in "mod_bvalues". Specify the attribute value address and size in the berval structure.

  The pointer array of each attribute value must have NULL set in the last option.

For examples of the parameters used for adding entries, refer to 'The parameter configuration in "add" processing'.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

1.21.8.3 ldap_add_ext()

Name

ldap_add_ext

Synopsis

```
#include "idldap.h"
int  ldap_add_ext(
        LDAP  *ld,
        const char  *dn,
        LDAPMod  **attrs,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls,
        int  *msgidp );
```

Description

This function adds entries asynchronously. If there is no parent, the specified entry cannot be added.

To receive the asynchronous function processing result, use ldap_result() and specify the message ID that is returned as the ldap_add_ext() return value. For details, refer to Receiving/Determining Processing Results. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entry to be added.

attrs

Specify the address of the LDAPMod Structure pointer array. Set the attribute and attribute value to be added to the entry in each LDAPMod structure. The values that can be set for each option in the LDAPMod structure are shown below.

- mod_op

  Specify the attribute value type.

  - If the attribute value is string data: "0"
  - If the attribute value is binary data: LDAP_MOD_BVALUES

- mod_type

  Specify the address of the attribute to be changed. Some of the attributes that are specified must contain ";binary" in the attribute name.

- mod_vals

  Specify the address of the attribute value.

  The attribute value specified here is the shared mod_vals pointer array. This pointer array must have NULL set in the last option. The shared "mod_vals" member that is used depends on the attribute value type specified in the "mod_op" parameter, as shown below.

  - String data: Specify the address of the pointer array for the attribute value in "mod_values".
  - Binary data: Specify the address of the berval Structure pointer array in "mod_bvalues". Specify the attribute value address and size in the berval structure.

  The pointer array of each attribute value must have NULL set in the last option.

For examples of the parameters used for adding entries, refer to 'The parameter configuration in "add" processing'.

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

msgidp

Specify the address of the variable for storing the message ID.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

If the return of the value was terminated normally, the message ID is set in the variable shown in the "msgidp" parameter.

1.21.8.4 ldap_add_ext_s()

Name

ldap_add_ext_s

Synopsis

```
#include "idldap.h"
int  ldap_add_ext_s(
        LDAP  *ld,
        const char  *dn,
        LDAPMod  **attrs,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls );
```

Description

This function adds entries synchronously. If there is no parent, the specified entry cannot be added. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entry to be added.

attrs

Specify the address of the LDAPMod Structure pointer array. Set the attribute and attribute value to be added to the entry in each LDAPMod structure. The values that can be set for each option in the LDAPMod structure are shown below.

- mod_op

  Specify the attribute value type.

  - If the attribute value is string data: "0"
  - If the attribute value is binary data: LDAP_MOD_BVALUES

- mod_type

  Specify the address of the attribute to be changed. Some of the attributes that are specified must contain ";binary" in the attribute name.

- mod_vals

  Specify the address of the attribute value.

  The attribute value specified here is the shared mod_vals pointer array. This pointer array must have NULL set in the last option. The shared "mod_vals" member that is used depends on the attribute value type specified in the "mod_op" parameter, as shown below.

  - String data: Specify the address of the pointer array for the attribute value in "mod_values".
  - Binary data: Specify the address of the berval Structure pointer array in "mod_bvalues". Specify the attribute value address and size in the berval structure.

  The pointer array of each attribute value must have NULL set in the last option.

For examples of the parameters used for adding entries, refer to 'The parameter configuration in "add" processing'.

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

1.21.9 Interface for Deleting Entries

Function	Explanation
ldap_delete()	Deletes the specified entry. (Asynchronous type)
ldap_delete_s()	Deletes the specified entry. (Synchronous type)
ldap_delete_ext()	Deletes the specified entry. (Asynchronous type, LDAP V3 extended function)
ldap_delete_ext_s()	Deletes the specified entry. (Synchronous type, LDAP V3 extended function)

1.21.9.1 ldap_delete()

Name

ldap_delete

Synopsis

```
#include "idldap.h"
int  ldap_delete(
        LDAP  *ld,
        const char  *dn );
```

Description

This function deletes entries asynchronously. Only leaf entries can be deleted.

To receive the asynchronous function processing result, use ldap_result() and specify the message ID that is returned as the ldap_delete() return value. For details, refer to Receiving/Determining Processing Results. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entry to be deleted.

Return Values

This function returns the following values:

- For normal termination: Message ID
- For abnormal termination: "-1"

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

1.21.9.2 ldap_delete_s()

Name

ldap_delete_s

Synopsis

```
#include "idldap.h"
int  ldap_delete_s(
        LDAP  *ld,
        const char  *dn );
```

Description

This function deletes entries synchronously. Only leaf entries can be deleted. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entry to be deleted.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

1.21.9.3 ldap_delete_ext()

Name

ldap_delete_ext

Synopsis

```
#include "idldap.h"
int  ldap_delete_ext(
        LDAP  *ld,
        const char  *dn,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls,
        int  *msgidp );
```

Description

This function deletes entries asynchronously. Only leaf entries can be deleted.

To receive the asynchronous function processing result, use ldap_result() and specify the message ID that is returned as the ldap_delete() return value. For details, refer to Receiving/Determining Processing Results. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entry to be deleted.

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

msgidp

Specify the address of the variable for storing the message ID.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

If the return of the value was terminated normally, the message ID is set in the variable specified in the "msgidp" parameter.

1.21.9.4 ldap_delete_ext_s()

Name

ldap_delete_ext_s

Synopsis

```
#include "idldap.h"
int  ldap_delete_ext_s(
        LDAP  *ld,
        const char  *dn,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls );
```

Description

This function deletes entries synchronously. Only leaf entries can be deleted. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

dn

Specify the DN address of the entry to be deleted.

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

1.21.10 Interface for Canceling Asynchronous Processing

Function	Explanation
ldap_abandon()	Cancels asynchronous processing.
ldap_abandon_ext()	Cancels asynchronous processing. (LDAP V3 extended function)

1.21.10.1 ldap_abandon()

Name

ldap_abandon

Synopsis

```
#include "idldap.h"
int  ldap_abandon(
        LDAP  *ld,
        int  msgid );
```

Description

This function cancels asynchronous type LDAP operations. 'result' information for a request for which processing was canceled is itself canceled in the library internally. For this reason, the call source is not notified of the 'result' information. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

msgid

Specify the message ID of the asynchronous processing to be canceled.

Return Values

This function returns the following values:

- For normal termination: "0"
- For abnormal termination: "-1"

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

1.21.10.2 ldap_abandon_ext()

Name

ldap_abandon_ext

Synopsis

```
#include "idldap.h"
int  ldap_abandon_ext(
        LDAP  *ld,
        int  msgid,
        LDAPControl  **serverctrls,
        LDAPControl  **clientctrls );
```

Description

This function cancels asynchronous type LDAP operations. 'result' information for a request for which processing was canceled is itself canceled in the library internally. For this reason, the call source is not notified of the 'result' information. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

msgid

Specify the message ID of the asynchronous processing to be canceled.

serverctrls

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

clientctrls

Client control is not supported in this library, so specify NULL in this parameter.

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

1.21.11 Interface for Receiving/Analyzing Processing Results

Receiving/Determining Processing Results

Function	Explanation
ldap_result()	Receives asynchronous function processing results and search result entries.
ldap_msgid()	References the message ID of the specified message.
ldap_msgtype()	References the type of the specified message.

1.21.11.1 ldap_result()

Name

ldap_result

Synopsis

```
#include "idldap.h"
int  ldap_result(
        LDAP  *ld,
        int  msgid,
        int  all,
        struct timeval  *timeout,
        LDAPMessage  **resp );
```

Description

This function receives asynchronous function processing results and search result entries.

To determine which asynchronous request the obtained "result" information is for, refer to the message ID of the "result" information in ldap_msgid(). The message ID for the asynchronous request is added to the "result" information. If the session handle obtained using ldapssl_init() is used, the error that occurred in the SSL library can be referenced using ldapssl_error().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

msgid

Specify the message ID that is notified in the asynchronous function. If you do not want to receive a particular processing result, specify LDAP_RES_ANY. If LDAP_RES_ANY is specified, the first message that is received is notified.

all

If the processing result consists of multiple messages, specify one of the following options to process the messages.

- LDAP_MSG_ALL: Receives all of the messages in the processing result for the requested "msgid" parameter and returns to the call source.
- LDAP_MSG_ONE: Receives one message in the processing result for the requested "msgid" parameter and returns to the call source.
- LDAP_MSG_RECEIVED: Searches the messages in the processing result that have been received, reads those messages that match the conditions specified in the "msgid" parameter and returns to the call source.

timeout

Specify the address of the area for setting the receive wait time. Set the value shown below as the timeval Structure member.

- tv_sec: Timer value (seconds)
- tv_usec: Timer value (microseconds)

If the NULL pointer is specified in the "timeout" parameter, the option does not return to the call source until the result is received. If "0" is set in the "tv_sec" and "tv_usec" parameters of the timeval structure, determine whether the specified message has been delivered. If the message has not been delivered, a timeout is returned immediately.

resp

Specify the address of the pointer variable for the LDAPMessage structure.

Return Values

This function returns the following values:

- For normal termination: Message type (The start of the message list)
- Timeout: "0"
- For abnormal termination: "-1"

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

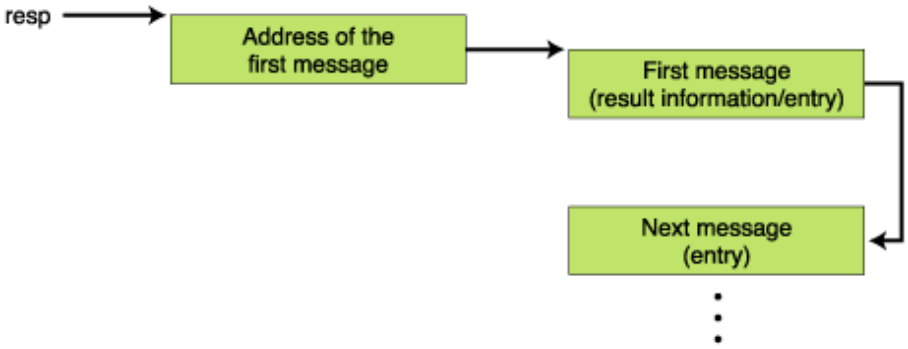
The message types that are notified if the return of the value was terminated normally are shown in the table below.

Table 1.13 Message Types Posted when ldap_result() Terminates Normally

Message type	Value	Explanation
LDAP_RES_BIND	0x61	This is the "result" information for user authentication with the repository server (BIND).
LDAP_RES_SEARCH_ ENTRY	0x64	This shows the entries that were found as a result of the search for entries (SEARCH).
LDAP_RES_SEARCH_ RESULT	0x65	This shows the entry search result report.
LDAP_RES_MODIFY	0x67	This is the "result" information for changing entries (MODIFY).
LDAP_RES_ADD	0x69	This is the "result" information for adding entries (ADD).
LDAP_RES_DELETE	0x6B	This is the "result" information for deleting entries (DELETE).
LDAP_RES_MODDN	0x6D	This is the "result" information for changing entry names (MODDN).
LDAP_RES_COMPARE	0x6F	This is the "result" information for comparing attribute values (COMPARE).

If no messages are received, the start address of the message list is set in the pointer variable indicated in the "resp" parameter. The LDAPMessage notification format is shown below.

Figure 1.5 LDAPMessage Notification Format



The message list notified in ldap_result() is specified as the parameter in message list processing.

Notes

- Free dynamic memory

  The message list notified in ldap_result() must be unbound using ldap_msgfree() once it is no longer required. For details, refer to 1.21.18 Interface for Releasing Dynamic Memory.

1.21.11.2 ldap_msgid()

Name

ldap_msgid

Synopsis

```
#include "idldap.h"
int  ldap_msgid(
        LDAPMessage  *lm );
```

Description

This function notifies the message ID of the specified message.

Parameters

lm

Specify the address of the message.

Return Values

This function returns the following values:

- For normal termination: The message ID of the specified message
- For abnormal termination: "-1"

  If the return of the value was terminated abnormally, it is possible that an incorrect parameter was specified.

1.21.11.3 ldap_msgtype()

Name

ldap_msgtype

Synopsis

```
#include "idldap.h"
int  ldap_msgtype(
        LDAPMessage  *lm );
```

Description

This function notifies the specified message type. For details of the message type values, refer to Return Values in ldap_result().

Parameter

lm

Specify the address of the message.

Return Values

This function returns the following values:

- For normal termination: The message type of the specified message
- For abnormal termination: "-1"

  If the return of the value was terminated abnormally, it is possible that an incorrect parameter was specified.

1.21.12 Interface for Obtaining Error Information

Obtaining Error Information

The follow functions can be used to obtain error information.

Function	Explanation
ldap_parse_result()	Reads the LDAP error information of the specified "result" information.
ldap_err2string()	Converts the LDAP error code to a string.
ldapssl_error()	Reads SSL library error codes.

1.21.12.1 ldap_parse_result()

Name

ldap_parse_result

Synopsis

```
#include "idldap.h"
int  ldap_parse_result(
        LDAP  *ld,
        LDAPMessage  *res,
        int  *errcodep,
        char  **matcheddnp,
        char  **errmsgp,
        char  ***referralsp,
        LDAPControl  ***serverctrlsp,
        int  freeit );
```

Description

This function reads various information in the received "result" information. This function is used for all "result" information except LDAP_RES_SEARCH_ENTRY.

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

res

Specify the address of the "result" information.

errcodep

Specify the address of the variable for storing LDAP error codes contained in the "result" information. If the LDAP error code is not to be read, specify NULL in this parameter.

matcheddnp

Specify the address of the pointer variable for storing the DN address that indicates the matched range. If the DN address that indicates the matched range is not to be read, specify NULL in this parameter.

errmsgp

Specify the address of the variable for storing detailed error messages. If the detailed error messages are not to be read, specify NULL in this parameter.

referralsp

Referral is not supported in Interstage Directory Service, so specify NULL in this parameter.

serverctrlsp

Server control is not supported in Interstage Directory Service, so specify NULL in this parameter.

freeit

Specify whether to unbind the "result" information specified in the "res" parameter when the value is returned to the call source.

- To unbind the "result" information: Specify any value except "0"
- To not unbind the "result" information: Specify "0"

Return Values

This function returns an LDAP error code as the return value. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

- For normal termination: LDAP_SUCCESS
- For abnormal termination: Any LDAP error code except LDAP_SUCCESS

If the return of the value was terminated normally, the information shown below is set in the variable specified in the parameter.

- The variable specified in the "errcodep" parameter

  The LDAP error code is set.

- The variable specified in the "matcheddnp" parameter

  The address of the string that indicates the appropriate DN range is set. This area must be unbound using ldap_memfree() once it is no longer required.

- The variable specified in the "errmsgp" parameter

  The address of the detailed error messages in the specified "result" information is set. This area must be unbound using ldap_memfree() once it is no longer required.

Notes

- Free "result" information

  If any value except "0" is specified in the "freeit" parameter when ldap_parse_result() is called, ldap_msgfree() is used in the library internally to unbind the "result" information. For this reason, the "result" information can no longer be accessed. If "0" is specified in the "freeit" parameter when ldap_parse_result() is called, the "result" information must be unbound using ldap_msgfree() in the call source once it is no longer required.

1.21.12.2 ldap_err2string()

Name

ldap_err2string

Synopsis

```
#include "idldap.h"
char  *ldap_err2string(
        int  err );
```

Description

This function notifies an error message string for the specified LDAP error code.

Parameter

err

Specify the LDAP error code. For details of LDAP error code values, refer to "LDAP Error Codes" in the Messages manual.

Return Values

This function returns the address of the error message for the specified LDAP error code as the function return value. If the specified LDAP error code is unconfigured, an "Unknown error" is notified.

1.21.12.3 ldapssl_error()

Name

ldapssl_error

Synopsis

```
#include "idldap.h"
char  *ldapssl_error(
        LDAP  *ld,
        int  *ssl_err,
        int  *ssl_err_detail );
```

Description

This function obtains error codes detected in the SSL library when SSL is used. For details about SSL error codes, refer to "Error codes notified from Interstage Directory Service" - "SSL error codes" in "Messages".

Parameter

ld

Specifies the session handle notified in ldapssl_init().

ssl_err

Specifies the variable address set for the SSL error code.

ssl_err_detail

Specifies the variable address set for the SSL error detail code.

Return Values

None

1.21.13 Interface for Processing Message Lists

Processing the Message List

Function	Explanation
ldap_first_message()	Returns the address of the first message.
ldap_next_message()	Returns the address of the next message. (The message that follows the one that was read previously)
ldap_count_messages()	Counts the number of messages contained in the message list.

1.21.13.1 ldap_first_message()

Name

ldap_first_message

Synopsis

```
#include "idldap.h"
LDAPMessage  *ldap_first_message(
        LDAP  *ld,
        LDAPMessage  *res );
```

Description

This function notifies the address of the start message in ldap_result() the message list notified in synchronous type search processing.

The address of the message notified in ldap_first_message() can be used as a parameter in the following functions:

- Processing the message list:
  - ldap_next_message()
  - ldap_count_messages()
- Obtaining error information processing (for "result" information)
  - ldap_parse_result()

- Entry list processing
  - ldap_first_entry()
  - ldap_next_entry()
  - ldap_count_entries()

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

res

Specify the address of the message list notified in ldap_result() or synchronous type search processing.

Return Values

This function returns the following values:

- For normal termination: The address of the start message
- For abnormal termination: NULL

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

1.21.13.2 ldap_next_message()

Name

ldap_next_message

Synopsis

```
#include "idldap.h"
LDAPMessage  *ldap_next_message(
        LDAP  *ld,
        LDAPMessage  *msg );
```

Description

This function notifies the address of the message that follows the message notified in ldap_first_message() in ldap_next_message().

The address of the message notified in ldap_next_message() can be used as a parameter in the following functions:

- Processing the message list:
  - ldap_next_message()
  - ldap_count_messages()
- Obtaining error information processing (for "result" information)
  - ldap_parse_result()
- Entry list processing
  - ldap_first_entry()
  - ldap_next_entry()
  - ldap_count_entries()

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

msg

Specify the address of the message notified in ldap_first_message() or ldap_next_message().

Return Values

This function returns the following values:

- For normal termination: The address of the next message
- For abnormal termination: NULL

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

  If there is no next message, NULL is returned as the return value.

1.21.13.3 ldap_count_messages()

Name

ldap_count_messages

Synopsis

```
#include "idldap.h"
int  ldap_count_messages(
        LDAP  *ld,
        LDAPMessage  *res );
```

Description

This function counts the number of messages contained in the message list.

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

res

Specify the address of the message notified in ldap_first_message() or ldap_next_message(). If the address of the message notified in ldap_next_message() is specified, the number of messages from that point on are counted.

Return Values

This function returns the message number as the return value. If NULL is specified in the "res" parameter, "0" is returned.

1.21.14 Interface for Processing Entry Lists

Entry List Processing

Function	Explanation
ldap_first_entry()	Returns the first entry of the specified message.
ldap_next_entry()	Returns the next entry of the specified message. (The entry that follows the one that was read previously)
ldap_count_entries()	Counts the number of entries in the specified message.

1.21.14.1 ldap_first_entry()

Name

ldap_first_entry

Synopsis

```
#include "idldap.h"
LDAPMessage  *ldap_first_entry(
        LDAP  *ld,
        LDAPMessage  *res );
```

Description

This function notifies the address of the start entry that is contained in the specified message.

The address of the entry notified in ldap_first_entry() can be used as a parameter in the following functions:

- Entry list processing
  - ldap_next_entry()
  - ldap_count_entries()
- DN reading/analysis

  ldap_get_dn()
- Attribute reading
  - ldap_first_attribute()
  - ldap_next_attribute()
- Attribute value reading
  - ldap_get_values()
  - ldap_get_values_len()
  - ldap_count_values()
  - ldap_count_values_len()

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

res

Specify the address of the message list. If the address of the entry notified in ldap_first_entry() or ldap_next_entry() is specified, the number of remaining entries (including the specified entries) is counted.

Return Values

This function returns the following values:

- For normal termination: The address of the start entry
- For abnormal termination: NULL

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

1.21.14.2 ldap_next_entry()

Name

ldap_next_entry

Synopsis

```
#include "idldap.h"
LDAPMessage  *ldap_next_entry(
        LDAP  *ld,
        LDAPMessage  *entry );
```

Description

This function notifies the address of the entry that follows the entry notified in ldap_first_entry() or ldap_next_entry().

The address of the entry notified in ldap_next_entry() can be used as a parameter in the following functions:

- Entry list processing
  - ldap_next_entry()
  - ldap_count_entries()
- DN reading/analysis

  ldap_get_dn()
- Attribute reading
  - ldap_first_attribute()
  - ldap_next_attribute()
- Attribute value reading
  - ldap_get_values()
  - ldap_get_values_len()
  - ldap_count_values()
  - ldap_count_values_len()

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

entry

Specify the address of the entry notified in ldap_first_entry() or ldap_next_entry().

Return Values

This function returns the following values:

- For normal termination: The address of the next entry
- For abnormal termination: NULL

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

  If there is no next entry, NULL is returned as the return value.

1.21.14.3 ldap_count_entries()

Name

ldap_count_entries

Synopsis

```
#include "idldap.h"
int  ldap_count_entries(
```

```
        LDAP  *ld,
        LDAPMessage  *res );
```

Description

This function counts the number of entries contained in the specified message.

Parameters

ld

Specify the session handle notified in ldap_init() ot ldapssl_init().

res

Specify the address of the entry notified in ldap_first_entry() or ldap_next_entry().

If the address of the entry notified in ldap_next_entry() is specified, the number of entries from that point onwards are counted.

Return Values

This function returns the entry number as the return value. If there is no entry, "0" is returned as the return value. If NULL is specified in the "res" parameter, "0" is also returned.

1.21.15 Interface for Reading Attributes

Attribute Read Processing

Function	Explanation
ldap_first_attribute()	Returns the first attribute of the specified entry.
ldap_next_attribute()	Returns the next attribute of the specified entry. (The attribute that follows the one that was read previously)

1.21.15.1 ldap_first_attribute()

Name

ldap_first_attribute

Synopsis

```
#include "idldap.h"
char  *ldap_first_attribute(
        LDAP  *ld,
        LDAPMessage  *entry,
        BerElement  **ptr );
```

Description

This function notifies the first attribute name of the specified entry.

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

entry

Specify the address of the entry notified in ldap_first_entry() or ldap_next_entry().

ptr

Specify the address of the pointer variable for storing the obtained BerElement structure address.

Return Values

This function returns the following values:

- For normal termination: The address of the first attribute name

  The address of this attribute name is specified in the attr parameter of ldap_get_values() or ldap_get_values_len(). The obtained BerElement structure address is set in the pointer variable specified in the "ptr" parameter.

- For abnormal termination: NULL

  For details of the cause of the error in the case of abnormal termination, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

Notes

- Free dynamic memory

  The BerElement structure that is notified in ldap_first_attribute() must be unbound using ber_free() once it is no longer required. The attribute name area returned from ldap_first_attribute() as the return value must be unbound using ldap_memfree() once it is no longer required. For details, refer to 1.21.18 Interface for Releasing Dynamic Memory.

1.21.15.2 ldap_next_attribute()

Name

ldap_next_attribute

Synopsis

```
#include "idldap.h"
char  *ldap_next_attribute(
        LDAP  *ld,
        LDAPMessage  *entry,
        BerElement  *ber );
```

Description

This function notifies the attribute name that follows the attribute name notified in ldap_first_attribute() in ldap_next_attribute().

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

entry

Specify the address of the entry as for ldap_first_attribute() and ldap_next_attribute().

ber

Specify the address of the BerElement structure used in ldap_first_attribute() or ldap_next_attribute().

Return Values

This function returns the following values:

- For normal termination: The address of the next attribute

  The address of this attribute name is specified in the attr parameter of ldap_get_values() or ldap_get_values_len(). The contents of the BerElement structure indicated by the "ber" parameter are updated.

- For abnormal termination: NULL

  For details of the cause of the error in the case of abnormal termination, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

If there is no next attribute name, NULL is returned as the return value.

Notes

- Free dynamic memory

  The attribute name area returned from ldap_next_attribute() as the return value must be unbound using ldap_memfree() once it is no longer required. For details, refer to 1.21.18 Interface for Releasing Dynamic Memory.

1.21.16 Interface for Reading Attribute Values

Attribute Value Read Processing

Function	Explanation
ldap_get_values()	Reads the attribute value of the specified attribute. (If the attribute value is string data)
ldap_get_values_len()	Reads the attribute value of the specified attribute. (If the attribute value is binary data)
ldap_count_values()	Counts the number of read attribute values. (If the attribute value is string data)
ldap_count_values_len()	Counts the number of read attribute values. (If the attribute value is binary data)

1.21.16.1 ldap_get_values()

Name

ldap_get_values

Synopsis

```
#include "idldap.h"
char  **ldap_get_values(
        LDAP  *ld,
        LDAPMessage  *entry,
        const char  *attr );
```

Description

This function reads the attribute value of the specified entry as string data.

The attribute value that indicates the notified pointer array member is string data. The address of the notified pointer array is used as the ldap_count_values() parameter.

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

entry

Specify the address of the entry that is notified by ldap_first_entry() or ldap_next_entry().

attr

Specify the address of the attribute name that is included in the search result, an alias cannot be used. The attribute that is notified by ldap_first_attribute() or ldap_next_attribute() can be specified.

Return Values

This function returns the following values:

- For normal termination: The address of the attribute value pointer array
- For abnormal termination: NULL

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

  If there is no next attribute, NULL is returned as the return value.

Notes

- Free dynamic memory

  The pointer array that is notified as the return value in ldap_get_values() must be unbound using ldap_value_free() once it is no longer required.

  For details, refer to 1.21.18 Interface for Releasing Dynamic Memory.

1.21.16.2 ldap_get_values_len()

Name

ldap_get_values_len

Synopsis

```
#include "idldap.h"
struct berval  **ldap_get_values_len(
        LDAP  *ld,
        LDAPMessage  *entry,
        const char  *attr );
```

Description

This function reads the attribute value of the specified entry as binary data.

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

entry

Specify the address of the entry that is notified by ldap_first_entry() or ldap_next_entry().

attr

Specify the address of the attribute name that is included in the search result, an alias cannot be used.

The attribute that is notified by ldap_first_attribute() or ldap_next_attribute() can be specified.

Return Values

This function returns the following values:

- For normal termination: The address of the attribute value pointer array
- For abnormal termination: NULL

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

  If there is no attribute, NULL is returned as the return value.

Notes

- Free dynamic memory

  The pointer array that is notified as the return value in ldap_get_values_len() must be unbound using ldap_value_free_len() once it is no longer required. For details, refer to 1.21.18 Interface for Releasing Dynamic Memory.

1.21.16.3 ldap_count_values()

Name

ldap_count_values

Synopsis

```
#include "idldap.h"
int  ldap_count_values(
        char  **vals );
```

Description

This function counts the number of attribute values notified in ldap_get_values().

Parameter

vals

Specify the address of the pointer array that is notified by ldap_get_values().

Return Values

This function returns the attribute value number as the return value. If NULL is specified in the "vals" parameter, "0" is returned.

1.21.16.4 ldap_count_values_len()

Name

ldap_count_values_len

Synopsis

```
#include "idldap.h"
int  ldap_count_values_len(
        struct berval  **vals );
```

Description

This function counts the number of attribute values notified in ldap_count_values_len().

Parameter

vals

Specify the address of the pointer array that is notified by ldap_get_values_len().

Return Values

This function returns the attribute value number as the return value. If NULL is specified in the "vals" parameter, "0" is returned.

1.21.17 Interface for Reading/Analyzing the DN

DN Reading/Analysis

Function	Explanation
ldap_get_dn()	Reads the DN of the specified entry.
ldap_explode_dn()	Explodes the specified DN into configuration elements.
ldap_explode_rdn()	Explodes the specified RDN into configuration elements.

Function	Explanation
ldap_dn2ufn()	Converts the specified DN to a user-friendly format.

1.21.17.1 ldap_get_dn()

Name

ldap_get_dn

Synopsis

```
#include "idldap.h"
char  *ldap_get_dn(
        LDAP  *ld,
        LDAPMessage  *entry );
```

Description

This function reads the DN of the specified entry.

Parameters

ld

Specify the session handle notified in ldap_init() or ldapssl_init().

entry

Specify the address of the entry that is notified by ldap_first_entry() or ldap_next_entry().

Return Values

This function returns the following values:

- For normal termination: DN address
- For abnormal termination: NULL

  For details of the cause of the error, refer to the LDAP_OPT_RESULT_CODE option of ldap_get_option().

Notes

- Free dynamic memory

  The DN area that is notified in ldap_get_dn() must be unbound using ldap_memfree() once it is no longer required. For details, refer to 1.21.18 Interface for Releasing Dynamic Memory.

1.21.17.2 ldap_explode_dn()

Name

ldap_explode_dn

Synopsis

```
#include "idldap.h"
char  **ldap_explode_dn(
        const char  *dn,
        int  notypes );
```

Description

This function explodes the specified DN into it's component parts.

Parameters

dn

Specify the DN address.

notypes

Specify the following values to determine the format for configuration element output.

- "0": Notifies with the attribute added.
- All values except "0": Notifies without the attribute. (Example: o=fujitsu ---> fujitsu)

Return Values

This function returns the following values:

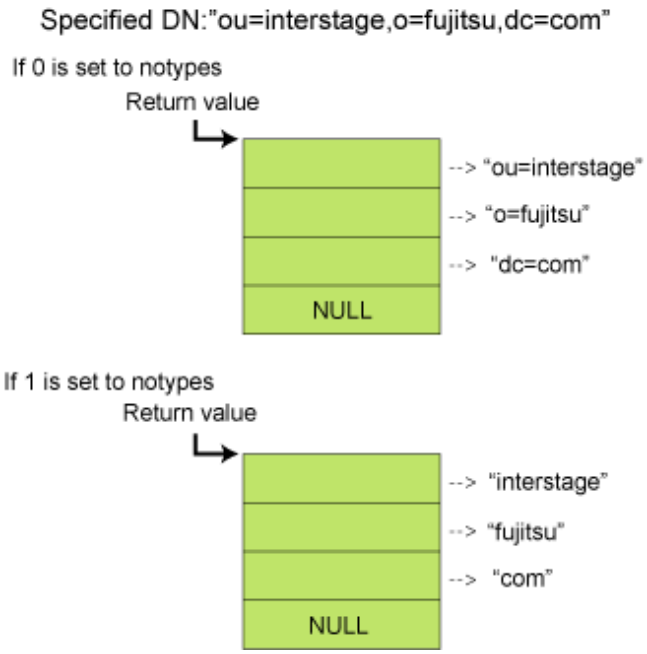
- For normal termination: The address of the pointer array for each configuration element
- For abnormal termination: NULL

  In the case of abnormal termination, possible causes are as follows:

  - An incorrect parameter was specified
  - There is insufficient usable memory

The following example shows the data that is notified in ldap_explode_dn().

Figure 1.6 Data Notified in ldap_explode_dn()



If the DN contains symbols, they are treated as special characters, as shown in the following table. Special characters are converted before being output.

Table 1.14 Conversion of Input Special Characters

Input character	Output character
"""(Double quotation mark)	\22
"#"(Sharp sign)	\23
"+"(Plus)	\2B
","(Comma)	\2C

Input character	Output character
";"(Semicolon)	\3B
"<"(Less-than sign)	\3C
">"(Greater-than sign)	\3E
"\"(Backslash)	\5C

Notes

- Free dynamic memory

  The area that is notified in ldap_explode_dn() must be unbound using ldap_value_free() once it is no longer required. For details, refer to 1.21.18 Interface for Releasing Dynamic Memory.

1.21.17.3 ldap_explode_rdn()

Name

ldap_explode_rdn

Synopsis

```
#include "idldap.h"
char  **ldap_explode_rdn(
        const char  *rdn,
        int  notypes );
```

Description

This function explodes the specified RDN into it's component parts.

Parameters

rdn

Specify the RDN address.

notypes

Specify the following values to determine the format for configuration element output.

- "0": Notifies with the attribute added.
- All values except "0": Notifies without the attribute. (Example: cn=User1 ---> User1)

Return Values

This function returns the following values:

- For normal termination: The address of the pointer array for each configuration element
- For abnormal termination: NULL

  In the case of abnormal termination, possible causes are as follows:

  - An incorrect parameter was specified
  - There is insufficient usable memory

The format for data notified in ldap_explode_rdn() is the same as for ldap_explode_dn().

If the DN contains symbols, they are treated as special characters and converted before being output. For details, refer to Return Values in ldap_explode_dn().

Notes

- Free dynamic memory

  The area that is notified in ldap_explode_rdn() must be unbound using ldap_value_free() once it is no longer required. For details, refer to 1.21.18 Interface for Releasing Dynamic Memory.

1.21.17.4 ldap_dn2ufn()

Name

ldap_dn2ufn

Synopsis

```
#include "idldap.h"
char  *ldap_dn2ufn(
        const char  *dn );
```

Description

This function converts the specified DN to a user-friendly format. Examples of converting the DN to user-friendly format are shown in the following table.

Conversion examples

The specified DN	User-friendly format
"ou=interstage,o=fujitsu,dc=com"	"interstage, fujitsu, com"
"dc=fujitsu,dc=com"	"fujitsu.com"

Parameter

dn

Specify the DN address to be converted.

Return Values

This function returns the following values:

- For normal termination: The address of the string in user-friendly format
- For abnormal termination: NULL

  In the case of abnormal termination, possible causes are as follows:

  - There is insufficient usable memory.
  - An incorrect parameter was specified.

If the DN contains symbols, they are treated as special characters and converted before being output. For details, refer to Return Values in ldap_explode_dn().

Notes

- Free dynamic memory

  The area that is notified in ldap_dn2ufn() must be unbound using ldap_memfree() once it is no longer required. For details, refer to 1.21.18 Interface for Releasing Dynamic Memory.

1.21.18 Interface for Releasing Dynamic Memory

Function	Explanation
ber_free()	Unbinds the specified BerElement structure.
ldap_ber_free()	Unbinds the specified BerElement structure.
ldap_memfree()	Unbinds the area obtained dynamically by the LDAP API function.
ldap_msgfree()	Unbinds the message list notified in ldap_result().
ldap_value_free()	Unbinds the pointer array of the string data.
ldap_value_free_len()	Unbinds the pointer array of the binary data.

1.21.18.1 ber_free()

Name

ber_free

Synopsis

```
#include "idldap.h"
void  ber_free(
        BerElement  *ber,
        int  freebuf );
```

Description

This function unbinds the area of the specified BerElement structure.

Parameters

ber

Specify the address of the BerElement structure to be unbound.

freebuf

- "0": The area retained as a "ber" parameter member is not unbound. Only the area for the address specified in the "ber" parameter is unbound.
- All values except "0": The area retained as a "ber" parameter member and the area for the address specified in the "ber" parameter are unbound.

Return Values

None.

1.21.18.2 ldap_ber_free()

Name

ldap_ber_free

Synopsis

```
#include "idldap.h"
void  ldap_ber_free(
        BerElement  *ber,
        int  freebuf );
```

Description

This function unbinds the area of the specified BerElement structure. This interface is not recommended, because it is retained to maintain compatibility. Use ber_free() instead.

Parameters

ber

Specify the address of the BerElement structure to be unbound.

freebuf

- "0": The area retained as a "ber" parameter member is not unbound. Only the area for the address specified in the "ber" parameter is unbound.
- All values except "0": The area retained as a "ber" parameter member, and the area for the address specified in the "ber" parameter are unbound.

Return Values

None.

1.21.18.3 ldap_memfree()

Name

ldap_memfree

Synopsis

```
#include "idldap.h"
void  ldap_memfree(
        char  *ptr );
```

Description

This function unbinds the area of the specified data.

Parameter

ptr

Specify the address of the area to be unbound.

Return Values

None.

1.21.18.4 ldap_msgfree()

Name

ldap_msgfree

Synopsis

```
#include "idldap.h"
int  ldap_msgfree(
        LDAPMessage  *res );
```

Description

This function unbinds the specified message list.

Parameter

res

Specify the address of the message list to be unbound.

Return Values

This function returns the following values:

- For normal termination: 1 or more
- For abnormal termination: 0

  If the return of the value terminated abnormally, it is possible that an incorrect parameter was specified.

1.21.18.5 ldap_value_free()

Name

ldap_value_free

Synopsis

```
#include "idldap.h"
void  ldap_value_free(
        char  **vals );
```

Description

This function unbinds the pointer array of the specified string.

Parameter

vals

Specify the address of the pointer array for the string to be unbound.

Return Values

None.

1.21.18.6 ldap_value_free_len()

Name

ldap_value_free_len

Synopsis

```
#include "idldap.h"
void  ldap_value_free_len(
        struct berval  **vals );
```

Description

This function unbinds the pointer array of the specified berval Structure.

Parameter

vals

Specify the address of the pointer array for the berval Structure to be unbound.

Return Values

None.

1.21.19 Interface for Referencing Version Information

Version Information Reference

Function	Explanation
ldap_version()	References library version information.

1.21.19.1 ldap_version()

Name

ldap_version

Synopsis

```
#include "idldap.h"
int  ldap_version(
        LDAPVersion  *ver );
```

Description

This function reads the library version information. This interface is not recommended. To read library version information, use the LDAP_OPT_API_INFO option in ldap_get_option().

Parameter

ver

Specify the address of the LDAPVersion Structure. If the LDAPVersion structure information is not required, specify NULL.

Return Values

This function returns the library version as the return value. The following information is set in each member for the area specified in the "ver" option.

- sdk_version

  The value that indicates the library version is set. The value set here is the same as the value returned for the function name return value. The value is 100 times greater than the notified version. (Example: V2.0 > 200)

- protocol_version

  The latest LDAP protocol version that can be used in this library is set. The value is 100 times greater than the notified version. (Example: V3 > 300)

- SSL_version

  The latest SSL protocol version that can be used in this library is set.

- sdk_vendor

  The address of the library source vendor is set.

1.21.20 Configuring the Structure

This section explains the structures that are used for calling the client API function. These are:

- berval Structure
- LDAPAPIInfo Structure

- LDAPMod Structure
- LDAPVersion Structure
- timeval Structure

1.21.20.1 berval Structure

Name

berval

Synopsis

```
struct berval {
        unsigned long bv_len;
        char         *bv_val;
};
```

Usage

This structure is used to specify string and binary data.

Members

bv_len

The size of the string or binary data

bv_val

The address of the string or binary data

1.21.20.2 LDAPAPIInfo Structure

Name

LDAPAPIInfo

Synopsis

```
typedef struct ldapapiinfo {
        int ldapai_info_version;
        int ldapai_api_version;
        int ldapai_protocol_version;
        char **ldapai_extensions;
        char *ldapai_vendor_name;
        int ldapai_vendor_version;
} LDAPAPIInfo;
```

How to Use

This structure is notified when the library version is read in ldap_get_option().

Members

ldapai_info_version

The version information of this structure.

ldapai_api_version

LDAP-API version information.

ldapai_protocol_version

The latest LDAP protocol version that can be used.

ldapai_extensions

Not used (Not notified in this library)

ldapai_vendor_name

The address of the source vendor.

ldapai_vendor_version

The version of this library.

1.21.20.3 LDAPMod Structure

Name

LDAPMod

Synopsis

```
typedef struct ldapmod {
        int  mod_op;
        char  *mod_type;
        union {
                char  **modv_strvals;
                struct  berval  **modv_bvals;
                } mod_vals;
#define mod_values  mod_vals.modv_strvals
#define mod_bvalues  mod_vals.modv_bvals
} LDAPMod;
```

Usage

This structure is used when the collection of attributes and attribute values is specified for the adding/changing of an entry.

Members

mod_op

Operating and attribute value types.

mod_type

The attribute address.

mod_vals

The address of the attribute value pointer array (This is shared).

mod_values

Used when the attribute value is string data.

mod_bvalues

Used when the attribute value is binary data.

1.21.20.4 LDAPVersion Structure

Name

LDAPVersion

Synopsis

```
typedef struct _LDAPVersion {
        int  sdk_version;
        int  protocol_version;
        int  SSL_version;
        int  security_level;
        char  *sdk_vendor;
        int  reserved[3];
} LDAPVersion;
```

Usage

This structure is notified when the library version is read in ldap_version().

Members

sdk_version

The version of this library.

protocol_version

The latest LDAP protocol version that can be used.

SSL_version

The latest SSL protocol version that can be used.

Note: The SSL communication function is not supported by this product.

security_level

The unused area (area that is retained for future use).

sdk_vendor

The address of the source vendor.

reserved

The unused area (area that is retained for future use).

1.21.20.5 SSLENV structure

Name

SSLENV

Synopsis

```
typedef struct sslenv {
        int  ssl_version;
        int  ssl_verify;
        char  *crypt;
        char  *slot_path;
        char  *tkn_lbl;
        char  *tkn_pwd;
        unsigned char  *cert_path;
        char  *crl_path;
        unsigned char  *user_cert;
        int  ssl_err;
        int  ssl_err_detail;
        int  ssl_timer;
        unsigned char  ssl_err_funcinfo;
        char  *reserve[15];
} SSLENV;
```

Usage

This structure is used to specify the SSL operating environment when SSL is used.

Members

ssl_version

SSL protocol version

ssl_verify

Certificate verification method

crypt

Encryption algorithm type

slot_path

Slot information directory address

tkn_lbl

Token label address

tkn_pwd

User PIN address

cert_path

Application management directory address

crl_path

CRL management directory address (not used in this version)

user_cert

User certificate nickname address

ssl_err

SSL error code

ssl_err_detail

SSL error detail code

ssl_timer

SSL timer

ssl_err_funcinfo

SSL library function code

reserve

Unused area (an area reserved for future use)

1.21.20.6 timeval Structure

Name

timeval

Synopsis

```
struct timeval {
        long  tv_sec;
```

```
        long  tv_usec;
};
```

Usage

This structure is used when the Timer value is specified.

Members

tv_sec

tv_sec is the nearest whole number (seconds) down of the specified Timer value.

For example, if the Timer value is specified as 5.6, tv_sec is set to 5.

tv_usec

tv_usec is the decimal part of the Timer value in microseconds.

For example, if the Timer value is specified as 5.6, tv_usec is set to 6,000,000 (0.6sec == 6,000,000 microseconds).

Chapter 2 C++ Interface

This chapter describes the C++ interface.

2.1 Environment Class

This section explains the environment class required to handle error information in the OD.

```
class CORBA {
class Environment
{
public:
void exception(Exception *, Environment&);
Exception *exception( Environment&) const;
void clear( Environment& );
}
}
```

2.1.1 CORBA::Environment::exception() (Setup)

Name

CORBA::Environment::exception

Synopsis

```
#include <orb_cplus.h>
void CORBA::Environment::exception(
     CORBA::Exception     *excp,
     CORBA::Environment&  env);
```

Description

This function sets exception information specified by excp in the environment object.

Parameters

excp

Specify the exception information to be set for the Environment object.

env

A structure that will contain exception information if the function does not terminate normally.

Return Values

None.

2.1.2 CORBA::Environment::exception() (Reference)

Name

CORBA::Environment::exception

Synopsis

```
#include <orb_cplus.h>
CORBA::Exception *CORBA::Environment::exception(
     CORBA::Environment&  env) const;
```

Description

This function reports the exception information set in the environment object.

Parameters

env

A structure that will contain exception information if the function does not terminate normally.

Return Values

For normal termination, an exception object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.1.3 CORBA::Environment::clear()

Name

CORBA::Environment::clear

Synopsis

```
#include <orb_cplus.h>
void CORBA::Environment::clear( );
```

Description

This function clears exception information set in the environment object.

Return Values

None.

2.2 ORB Class

This section explains the ORB (Object Request Broker) class providing the basic interface for initializing the ORB. The ORB function mediates communications between clients and servers.

```
class CORBA
{
public:
typedef string ORBid;
typedef string ObjectID ;
typedef BOA_ptr *BOA;
typedef NVList_ptr *NVList;
typedef OperationDef_ptr *OperationDef;
typedef Context_ptr *Context;
ORB_ptr ORB_init(int&, char**, ORBid);
class ORB : public CORBA
{
public:
BOA_ptr BOA_init (int *, char **, BOAid, CORBA::Environment& );
Object_ptr resolve_initial_references (ObjectID, CORBA::Environment&);
char *object_to_string(Object_ptr, CORBA::Environment&);
Object_ptr string_to_object(const char *, CORBA::Environment& );
Status create_list(Long, NVList_ptr&, CORBA::Environment& );
Status create_operation_list(OperationDef_ptr, NVList&, CORBA::Environment& );
Status get_default_context(Context_ptr&, CORBA::Environment& );
Status create_environment(Environment_ptr&, CORBA::Environment& );
```

```
Status send_multiple_requests_oneway(const RequestSeq&, CORBA::Environment& );
Status send_multiple_requests_deferred(const RequestSeq, CORBA::Environment& );
Status get_next_response(RequestSeq*&, CORBA::Environment& );
boolean poll_next_response( CORBA::Environment& );
};
class RequestSeq
{
public:
RequestSeq();
RequestSeq( ULong max );
RequestSeq(
ULong max,
ULong length,
Request_ptr *value,
Boolean release = FALSE
);
~RequestSeq();
private:
ULong _maximum;
ULong _length;
Request_ptr *_buffer;
};
}
```

2.2.1 CORBA::ORB_init()

Name

CORBA::ORB_init

Synopsis

```
#include <orb_cplus.h>
ORB_ptr CORBA::ORB_init(
     int&   argc,
     char   **argv,
     ORBid  orb_identifier,
CORBA::Environment& env);
```

Description

This function initializes ORB, and returns an ORB object reference. The functions described in the following sections can be used when this function is issued. The object reference returned is used in parameters when other ORB parameters are called.

Parameters

arg_c

A pointer to a copy of the argc to be passed to the main function.

arg_v

A pointer to a copy of the argv to be passed to the main function.

orb_identifier

A name that identifies the ORB. Specify FJ_OM_ORBid.

env

A structure that may contain exception information.

Return Values

For normal termination, the ORB object reference is set.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

Notes

- This function can only be called by one process at a time.
- For shared server and unshared server, set the following values in arg_v:

First parameter	"-ORB_FJ_PROC_ID"	The server application starting parameter
Second parameter	Process number	The OD control parameter
Third parameter	"-ORB_FJ_HOME_DIR"	The definition file specified in the -ax option of OD_impl_inst
Fourth parameter	Setting value of $OD_HOME	The definition file directory container defined at invoke time

- You can use HTTP tunneling on a client application by passing an HTTP tunneling parameter as an argument to CORBA::ORB_init(). For an application that passes main arguments to CORBA::ORB_init() as they are, specify an HTTP tunneling parameter as the startup parameter. For details of HTTP tunneling, refer to "How to start HTTP tunneling" in the Security System Guide.
- When CORBA::RB::resolve_initial_references() fetches individual client initial service object references, the values of argv and argc must be specified according to "Fetching Naming Service Initial Reference" in the Distributed Application Development Guide (CORBA Service Edition).

Example

- Example for HTTP tunneling.

```
char  *argv[] = {
    "",    /* The first argument is ignored */
    "-ORB_FJ_HTTP", "yes",
    "-ORB_FJ_HTTPGW", "http://host.com/od-httpgw",
    NULL };
int  argc = 5;

CORBA::Environment_ptr  env = new CORBA::Environment;
CORBA::ORB_ptr  orb;

orb = CORBA::ORB_init( argc, argv, FJ_OM_ORBid, *env );
```

- Example for initial service.

```
char  *argv[] = {
    "",    /* The first argument is ignored */
    "-ORBInitRef", "NameService=corbaloc::nshost:8002/NameService",
    "-ORBDefaultInitRef", "corbaloc::inithost:8002",
    NULL };
int  argc = 5;

CORBA::Environment_ptr  env = new CORBA::Environment;
CORBA::ORB_ptr  orb;

orb = CORBA::ORB_init( argc, argv, FJ_OM_ORBid, *env );
```

2.2.2 CORBA::ORB::BOA_init()

Name

CORBA::ORB::BOA_init

Synopsis

```
#include <orb_cplus.h>
BOA_ptr CORBA::ORB::BOA_init (
     int    *arg_c,
     char   **arg_v,
     OAid  boa_identifier,
CORBA::Environment& env );
```

Description

This function initializes BOA (Basic Object Adapter) that controls server applications, and returns the object reference of BOA.

Parameters

arg_c

A pointer to a copy of the argc to be passed to the main function.

arg_v

A pointer to a copy of the argv to be passed to the main function.

boa_identifier

Specify CORBA::BOA::OAid.

env

A structure that will contain exception information if the function abnormally terminates.

Return Values

For normal termination, the BOA object reference is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.2.3 CORBA::ORB::resolve_initial_references()

Name

CORBA::ORB::resolve_initial_references

Synopsis

```
#include <orb_cplus.h>
Object_ptr CORBA::ORB::resolve_initial_references (
     ObjectID identifier,
     CORBA::Environment& env );
```

Description

This function returns the object reference of the object specified by identifier.

The object reference is retrieved as described in "Fetching Naming Service Initial Reference" in the Distributed Application Development Guide (CORBA Service Edition).

identifier can specify the following objects:

CORBA_ORB_ObjectID_LightInterfaceRepository

Interface Repository (Static skeleton interface)

CORBA_ORB_ObjectID_InterfaceRepository

Interface Repository (Dynamic skeleton interface)

CORBA_ORB_ObjectID_ImplementationRepository

Implementation Repository

CORBA_ORB_ObjectID_NameService

Naming Service

Parameters

identifier

The initial reference identifier.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference is set.

If the specified object does not exist, NULL is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.2.4 CORBA::ORB::resolve_initial_references_remote()

Name

CORBA::ORB::resolve_initial_references_remote

Synopsis

```
#include <orb_cplus.h>
Object_ptr  CORBA::ORB::resolve_initial_references_remote(
     CORBA::ORB::ObjectId  identifier,
     CORBA::ORB::RemoteModifier& m,
     CORBA::Environment& env);
typedef CORBA::string_var  CORBA_Identifier
typedef CORBA::string_var InitAgentDesignator;
typedef sequence<InitAgentDesignator> RemoteModifier;
```

Description

This function returns the object reference of the object specified by the identifier.

The object reference is retrieved by searching initial_services of the host specified at m in URL format.

identifier can specify the following objects:

CORBA::ORB::ObjectID_InterfaceRepository

Interface Repository (Dynamic skeleton interface)

CORBA::ORB::ObjectID_CosNaming

Naming Service

Multiple URLs can be specified at m. If so, they are searched in the order of specification, and searching is discontinued as soon as the object reference is found.

The URL specification format is as follows:

```
iiop://<address>[:<port>]
```

Either the host name, DNS name, or the IP address can be specified at <address>. The address cannot be omitted. At <port>, specify the ORB port number at the connection destination.

Parameters

identifier

The initial reference identifier.

m

A list of connection URLs.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference is returned. For abnormal termination, the object reference of the SystemException object or the UserException object is set in the exception member of env. The following detailed information is set in UserException_id:

CORBA::ORB::InvalidName

Object specified for identifier not found

For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

Example

```
CORBA::String_var  *URLs;
CORBA::ORB::RemoteModifier *m;

URLs = CORBA::ORB::RemoteModifier::allocbuf(2);

URLs[0] = "iiop://remotehost01:8002";
URLs[1] = "iiop://remotehost02:8002";

m = new CORBA::ORB::RemoteModifier(2,2,URLs,CORBA_TRUE);

CORBA::Object_ptr
    obj = orb->resolve_initial_references_remote(
              CORBA_ORB_ObjectId_NameService, *m, env );
```

2.2.5 CORBA::ORB::list_initial_services()

Name

CORBA::ORB::list_initial_services

Synopsis

```
#include <orb_cplus.h>
ObjectIdList_ptr  CORBA::ORB::list_initial_services(
     CORBA::Environment&  env );
```

Description

This function returns a list of the ObjectIDs of usable objects.

The ObjectIDs are stored in the _buffer of the ObjectIDList_ptr as a string.

```
typedef String ObjectID;
typedef ObjectIDList *ObjectIDList_ptr;
class ObjectIDList{
```

```
private:
ObjectID *_buffer;
}
```

The object reference corresponding to ObjectID is retrieved by specifying the ObjectID returned by this function to the identifier of CORBA::ORB::resolve_initial_references().

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the ObjectID list is set.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.2.6 CORBA::ORB::object_to_string()

Name

CORBA::ORB::object_to_string

Synopsis

```
#include <orb_cplus.h>
char *object_to_string(
     Object_ptr  obj,
     CORBA::Environment  &env );
```

Description

This function converts the object reference of the object specified by obj into a string.

Parameters

obj

The object reference to be converted to a string.

env

A structure that may contain exception information.

Return Values

For normal termination, the string is set.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.2.7 CORBA::ORB::string_to_object()

Name

CORBA::ORB::string_to_object

Synopsis

```
#include <orb_cplus.h>
Object_ptr  CORBA::ORB::string_to_object(
     const char *str,
     CORBA::Environment& env );
```

Description

This function converts the string specified by str into an object reference. This function can convert all of the character string types listed in "URL Schema" in the Distributed Application Development Guide (CORBA Service Edition).

Since this function acquires areas to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

str

The string to be converted to an object reference.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference is set.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. Detailed information is set in SystemException_id. For details on minor codes that are set, refer to CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service. The value and meaning of _id are as follows:

IDL:CORBA/BAD_PARAM:1.0

There is an error in the format of the specified URL schema.

Remarks

For str, you can specify not only a character string beginning with "IOR:" that can be acquired by CORBA::ORB::object_to_string(), but also a character string beginning with "cobaloc" or "corbaname". For details, refer to "URL schemas" in the Distributed Application Development Guide (CORBA Service Edition).

Example

- Example for URL Schema.

```
int main(int argc, char **argv)
{
    CORBA::ORB_ptr  orb;
    CORBA::Object_ptr  obj_ns, obj_sv, obj_sv2;
    CORBA::Environment_ptr  env = new CORBA::Environment;
    int  current_argc = argc;

    orb = CORBA::ORB_init( current_argc, argv, FJ_OM_ORBid, *env );

    CosNaming::NameComponent_ptr
        name_component = new CosNaming::NameComponent;
    name_component->id = (const CORBA::Char *)"test1::intf1";
    name_component->kind = (const CORBA::Char *)"";

    CosNaming::NameComponent_var
        *name_component_var = CosNaming::Name::allocbuf(1);
    name_component_var[0] = name_component;
    CosNaming::Name_ptr
        name = new CosNaming::Name( 1, 1, name_component_var, CORBA_TRUE );
```

```
    obj_ns = orb->string_to_object(
        "corbaloc::nshost:8002/NameService",
        *env );

    CosNaming::NamingContext_ptr
         NamingContext_obj = CosNaming::NamingContext::_narrow( obj_ns );
    obj_sv = NamingContext_obj->resolve( *name, *env );
    test1::intf1_ptr  ap1 = test1::intf1::_narrow( obj_sv );
    test1::intf1_var  av1 = ap1;
    av1->add( 1, 2, *env );

    obj_sv2 = orb->string_to_object(
        "corbaname::nshost:8002/NameService#test1::intf1",
        *env );

    test1::intf1_ptr  ap2 = test1::intf1::_narrow( obj_sv2 );
    test1::intf1_var  av2 = ap2;
    av2->add( 1, 2, *env );

    return 0;
}
```

2.2.8 CORBA::ORB::create_list()

Name

CORBA_ORB_create_list

Synopsis

```
#include <orb_cplus.h>
CORBA::Status  CORBA::ORB::create_list(
     CORBA::Long  count,
     CORBA::NVList_ptr&  new_list,
     CORBA::Environment& env );
```

Description

This function generates a list (NVList) of items in which a number of parameters (specified by count) can be stored.

To add a parameter to the list object created here, use CORBA::NVList::add_item() in the NVList class. To release the list object created here, use CORBA::release().

Parameters

count

The number of items in the list object (NVList) that will contain parameters.

new_list

A pointer to the area that will contain the list object (NVList).

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned, and the list object is set in new_list.

For abnormal termination, CORBA::FAILED is returned.

2.2.9 CORBA::ORB::create_operation_list()

Name

CORBA::ORB::create_operation_list

Synopsis

```
#include <orb_cplus.h>
CORBA::Status  CORBA::ORB::create_operation_list(
     CORBA::OperationDef_ptr  oper,
     CORBA::NVList&  new_list,
     CORBA_Environment&  env );
```

Description

This function generates the list object (NVList) initialized by the parameter information for the operation specified by oper. The list object is returned in operation definition order.

ORB retrieves details of the operation information from the Interface Repository by using information specified by the ORB.

Parameters

oper

The OperationDef object to be used to create list objects.

new_list

A pointer to the area that will contain the list object.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned, and the list object is set in new_list.

For abnormal termination, CORBA::FAILED is returned.

2.2.10 CORBA::ORB::get_default_context()

Name

CORBA::ORB::get_default_context

Synopsis

```
#include <orb_cplus.h>
CORBA::Status  CORBA::ORB::get_default_context(
     CORBA::Context_ptr&  ctx,
     CORBA_Environment&   env );
```

Description

This function returns the default context object reference.

Parameters

ctx

A pointer to the area that will contain the Context object.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_OK is returned, and the default context object reference is set in ctx.

For abnormal termination, CORBA_FAILED is returned.

2.2.11 CORBA::ORB::create_environment()

Name

CORBA::ORB::create_environment

Synopsis

```
#include <orb_cplus.h>
CORBA::Status CORBA::ORB::create_environment(
     Environment_ptr&     envp,
     CORBA::Environment&  env )
```

Description

This function generates an environment object, and returns its object reference.

Parameters

envp

A pointer to the area that will contain the Environment object.

env

A structure that may contain exception information.

Return Values

For normal termination CORBA::OK is returned, and the environment object reference is set in envp.

For abnormal termination, CORBA::FAILED is returned.

2.2.12 CORBA::ORB::send_multiple_requests_oneway()

Name

CORBA::ORB::send_multiple_requests_oneway

Synopsis

```
#include <orb_cplus.h>
CORBA::Status CORBA::ORB::send_multiple_requests_oneway(
     const RequestSeq&    req,
     CORBA::Environment&  env );
```

Description

This function transmits one or more requests in parallel. The function returns to the calling program without waiting for the server application function to complete.

To discover when the server application function has ended, execute CORBA::ORB::get_response() and CORBA::ORB::get_next_response().

Parameters

req

The request object for the request to be sent.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.2.13 CORBA::ORB::send_multiple_requests_deferred()

Name

CORBA::ORB::send_multiple_requests_deferred

Synopsis

```
#include <orb_cplus.h>
CORBA::Status  CORBA::ORB::send_multiple_requests_deferred(
     CORBA::RequestSeq&     req,
     CORBA::Environment&    env );
```

Description

This function transmits one or more requests in parallel. The function returns to the calling program after the server application function terminates.

To discover when the server application function has ended, execute CORBA::ORB::get_next_response().

Parameters

req

The request object for the request to be sent.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.2.14 CORBA::ORB::get_next_response()

Name

CORBA::ORB::get_next_response

Synopsis

```
#include <orb_cplus.h>
CORBA::Status CORBA_Request_get_next_response(
     CORBA::Request_ptr& req,
     CORBA::Environment& env );
```

Description

This function returns the object reference of the request object (sent by CORBA::ORB::send_multiple_requests_oneway() or CORBA::ORB::send_multiple_requests_deferred()) related to the server application function to be terminated. The request termination sequence is not guaranteed.

Parameters

req

A pointer to the area for the request objects related to the server application function that will terminate next.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.3 Object Class

This section explains the Object class that controls object references.

```
class CORBA
{
typedef Identifier string;
typedef Context_ptr *Context;
typedef NamedValue_ptr *NamedValue;
typedef NVList_ptr *NVList;
typedef Request_ptr *Request;
typedef ImplementationDef_ptr *ImplementationDef;
typedef InterfaceDef_ptr *InterfaceDef;
class Object
{
public:
static Object_ptr _nil( CORBA::Environment& );
static Object_ptr _duplicate(Object_ptr obj, CORBA::Environment& );
Status_create_request(
Context_ptr ctx,
const char *operation,
NVList_ptr arg_list,
NamedValue_ptr result,
Request_ptr &request,
Flags req_flags,
CORBA::Environment&
);
Request_ptr _request(Identifier, CORBA::Environment& );
ImplementationDef_ptr _get_implementation( CORBA::Environment& );
InterfaceDef_ptr _get_interface( CORBA::Environment& );
};
};
```

2.3.1 CORBA::Object::_nil()

Name

CORBA::Object::_nil

Synopsis

```
#include <orb_cplus.h>
static Object_ptr CORBA::Object::_nil(
     CORBA::Environment& env );
```

Description

Returns NIL object reference.

Parameters

env

A structure that may contain exception information.

Return Values

NIL object reference.

2.3.2 CORBA::Object::is_nil()

Name

CORBA::Object::is_nil

Synopsis

```
#include <orb_cplus.h>
CORBA::Boolean  CORBA::Object::is_nil(
CORBA::Environment& env );
```

Description

This function checks that the specified object reference obj represents a valid object.

Parameters

env

A structure that may contain exception information.

Return Values

If obj represents a valid object, CORBA_FALSE is returned.

If obj does not represent a valid object, CORBA_TRUE is returned.

2.3.3 CORBA::Object::_duplicate()

Name

CORBA::Object::_duplicate

Synopsis

```
#include <orb_cplus.h>
static Object_ptr  CORBA::Object::_duplicate(
Object_ptr obj,
     CORBA::Environment& env );
```

Description

This function copies an object and returns the object reference of the copy.

Since this function acquires area to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

obj

The object reference to be replicated.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference of the copy is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.3.4 CORBA::Object::_create_request()

Name

CORBA::Object::_create_request

Synopsis

```
#include <orb_cplus.h>
CORBA::Status  CORBA::Object::_create_request(
     Context_ptr ctx,
     const char *operation,
     CORBA::NVList_ptr arg_list,
     CORBA::NamedValue_ptr result,
     CORBA::Request_ptr &request,
     Flags req_flags,
CORBA::Environment& env );
```

Description

This function creates a Request object initialized based on the specified arguments.

Parameters

ctx

Specify the Context object returned by CORBA::Context::create_child() or CORBA_OBJECT_NIL.

operation

The name of the method that will call the server application.

arg_list

The NVList object returned by CORBA::ORB::create_list() or CORBA::ORB::create_operation_list().

result

Specify the NamedValue object (returned by CORBA::NVList::add_item(),CORBA::NVList::add_value() functions) in result. For the NamedValue object TypeCode (any type), specify the TypeCode from the return value of the method called by CORBA::Request::invoke(), CORBA::Request::send_oneway(), or CORBA::Request::send_deffered().

request

An object reference for a request object.

The OD client must specify the request interface object reference when calling CORBA::Request::invoke(), or CORBA::Request::send_oneway(), CORBA::Request::send_deffered().

req_flags

The following flags can be specified in req_flags:

CORBA::OUT_LIST_MEMORY

This function links the list object (NVList) specified in arg_list to the request object. Specify arg_list for this function.

If CORBA::OUT_LIST_MEMORY is not specified, the list object is available until released by the program.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned, and the request object reference is set in request.

For abnormal termination, CORBA::FAILED is returned.

2.3.5 CORBA::Object::_request()

Name

CORBA::Object::_request

Synopsis

```
#include <orb_cplus.h>
CORBA::Request_ptr CORBA::Object::_request(
     CORBA::Identifier operation,
CORBA::Environment& env );
```

Description

This function returns a request object generated by the CORBA::Object::_create_request() function.

Parameters

operation

The operation name.

env

A structure that may contain exception information.

Return Values

For normal termination, the request object reference is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.3.6 CORBA::Object::_get_implementation()

Name

CORBA::Object::_get_implementation

Synopsis

```
#include <orb_cplus.h>
CORBA::ImplementationDef_ptr  CORBA::Object::_get_implementation(
CORBA::Environment& env );
```

Description

This function returns the object reference of the Implementation Repository that manages implementation information.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference of the Implementation Repository is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.3.7 CORBA::Object::_get_interface()

Name

CORBA::Object::_get_interface

Synopsis

```
#include <orb_cplus.h>
CORBA::InterfaceDef_ptr CORBA::Object::_get_interface(
CORBA::Environment& env );
```

Description

This function returns the object reference of the Interface Repository controlling interface information.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference of the Interface Repository is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

Remarks

To acquire interface information, the interface repository service is accessed within this API. Therefore, the interface information must be registered with the interface repository service.

2.4 BOA Class

This section describes the BOA class that provides the interface to control the server application.

```
class CORBA
{
typedef ImplementationDef_ptr *ImplementationDef;
typedef InterfaceDef_ptr *InterfaceDef;
class BOA {
public:
Object_ptr create( const ReferenceData &, InterfaceDef _ptr,
ImplementationDef_ptr, CORBA::Environment& );
void dispose(Object_ptr, CORBA::Environment& );
ReferenceData *get_id(Object _ptr, CORBA::Environment& );
void impl_is_ready(ImplementationDef_ptr, CORBA::Environment& );
void deactivate_impl(ImplementationDef_ptr, CORBA::Environment& );
void obj_is_ready(Object_ptr, ImplementationDef_ptr, CORBA::Environment& );
void deactivate_obj(Object_ptr, CORBA::Environment& );
};
class ReferenceData
{
public:
ReferenceData();
ReferenceData( ULong max );
ReferenceData(
ULong max,
ULong length,
Request_ptr *value,
Boolean release = FALSE
);
~ReferenceData();
private:
ULong _maximum;
ULong _length;
CORBA::octet *_buffer;
};
};
```

2.4.1 CORBA::BOA::create()

Name

CORBA::BOA::create

Synopsis

```
#include <orb_cplus.h>
CORBA::Object_ptr  CORBA::BOA::create(
     CORBA::ReferenceData &ref,
     CORBA::InterfaceDef_ptr intf,
     CORBA::ImplementationDef_ptr impl,
CORBA::Environment& env );
```

Description

This function generates and returns an object reference.

Code type information is appended to the object reference to be generated if the code information is set by the OD_impl_inst or OD_set_env commands.

When creating the object reference in the server application, the data conversion type is the value that would normally be set by the OD_or_adm command -L option (unused).

To dispose of the object reference created here, use CORBA::BOA::dispose(). Alternatively, to release the object reference memory, use CORBA::release().

Parameters

ref

Identification information for the object (to be set by the server application while generating it).

Identification information contains the CORBA::octet type set into the buffer member of the CORBA::ReferenceData class. This value does not change during the object's life.

intf

The object reference of the Interface Repository controlling the interface information of the generating objects.

impl

The object reference of the Implementation Repository controlling the real information of the generating objects.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference is set.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.4.2 CORBA::BOA::dispose()

Name

CORBA::BOA::dispose

Synopsis

```
#include <orb_cplus.h>
void CORBA::BOA::dispose(
     CORBA::Object_ptr obj,
CORBA::Environment& env );
```

Description

This function destroys the object reference specified in obj.

Parameters

obj

The object reference to be disposed.

env

A structure that may contain exception information.

Return Values

None.

2.4.3 CORBA::BOA::get_id()

Name

CORBA::BOA::get_id

Synopsis

```
#include <orb_cplus.h>
CORBA::ReferenceData  *CORBA::BOA::get_id(
     CORBA::Object _ptr obj,
CORBA::Environment& env );
```

Description

This function returns identification information of the object reference specified in obj. This information is the value specified by the CORBA::BOA::create() function, and does not change until the corresponding object is destroyed.

Identification information is set in _buffer of the CORBA_octet_sequence structure.

Parameters

obj

The identification information for the object reference specified here will be returned.

env

A structure that may contain exception information.

Return Values

For normal termination, identification information (octet type) is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.4.4 CORBA::BOA::impl_is_ready()

Name

CORBA::BOA::impl_is_ready

Synopsis

```
#include <orb_cplus.h>
void  CORBA::BOA::impl_is_ready(
     CORBA::ImplementationDef_ptr impl,
CORBA::Environment& env );
```

Description

This function notifies the ORB that the shared server or persistent server specified in impl is ready to receive requests.

For multi-threaded applications where SYNC_END is set in the impl information model, the function does not return until the CORBA::BOA::deactivate_impl() is issued.

Parameters

impl

Implementation information.

env

A structure that may contain exception information.

Return Values

None.

2.4.5 CORBA::BOA::deactivate_impl()

Name

CORBA::BOA::deactivate_impl

Synopsis

```
#include <orb_cplus.h>
void  CORBA::BOA::deactivate_impl(
     CORBA::ImplementationDef_ptr impl,
CORBA::Environment& env );
```

Description

This function notifies ORB that the process for the shared or persistent server specified in impl has stopped.

Parameters

impl

Implementation information.

env

A structure that may contain exception information.

Return Values

None.

2.4.6 CORBA::BOA::obj_is_ready()

Name

CORBA::BOA::obj_is_ready

Synopsis

```
#include <orb_cplus.h>
void CORBA::BOA::obj_is_ready(
CORBA::Object_ptr obj,
CORBA::ImplementationDef_ptr impl,
CORBA::Environment& env );
```

Description

This function notifies ORB that the object specified in obj on the unshared server specified in impl is ready to receive requests.

Parameters

obj

The object that is now ready for request receipt.

impl

Implementation information.

env

A structure that may contain exception information.

Return Values

None.

2.4.7 CORBA::BOA::deactivate_obj()

Name

CORBA::BOA::deactivate_obj

Synopsis

```
#include <orb_cplus.h>
void CORBA::BOA::deactivate_obj(
     CORBA::Object_ptr obj,
CORBA::Environment& env );
```

Description

This function notifies ORB that the object on the shared server or unshared server specified in obj has stopped.

Parameters

obj

The object to be stopped.

env

A structure that may contain exception information.

Return Values

None.

2.4.8 CORBA::BOA::set_impl_dsi()

Name

CORBA::BOA::set_impl_dsi

Synopsis

```
#include <orb_cplus.h>
void  CORBA::BOA::set_impl_dsi(
     CORBA::BOA_ptr boa,
     CORBA::Environment& env,
     CORBA::DynamicImplementationRoutine_cpp  dsi );
```

Description

This function registers the DSI processing function specified in dsi. The ORB calls the DSI function specified in dsi if the server application method is called.

Parameters

boa

The object reference returned by CORBA_ORB_BOA_init().

env

A structure that may contain exception information.

dsi

A pointer to the DSI processing function.

Return Values

For normal termination, the DSI processing function is registered.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.5 NamedValue Class

This section describes the NamedValue class used in the dynamic invocation interface.

```
class CORBA
{
class NamedValue
{
public:
const char *name(CORBA::Environment& ) const;
Any *value(CORBA::Environment& ) const;
                Flags flags(CORBA::Environment& ) const;
        };
};
```

2.5.1 CORBA::NamedValue::name()

Name

CORBA::NamedValue::name

Synopsis

```
#include <orb_cplus.h>
const char *CORBA::NamedValue::name(
CORBA::Environment& env) const;
```

Description

This function returns the name of the parameter information (NamedValue object) added to the NVList object by the CORBA::NVList::add_item() or CORBA::NVList::add_value() functions.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the name of the parameter information (NamedValue object) is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.5.2 CORBA::NamedValue::value()

Name

CORBA::NamedValue::value

Synopsis

```
#include <orb_cplus.h>
CORBA::Any *CORBA::NamedValue::value(
CORBA::Environment& env) const;
```

Description

This function returns the value (Any type) of the parameter information (NamedValue object) specified when adding the NVList object to the CORBA::NVList::add_value() function.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the value (Any type) of the parameter information (NamedValue object) is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.5.3 CORBA::NamedValue::flags()

Name

CORBA::NamedValue::flags

Synopsis

```
#include <orb_cplus.h>
CORBA::Flags CORBA::NamedValue::flags(
CORBA::Environment& env ) const;
```

Description

This function returns the flag of the parameter information (NamedValue object) specified when adding the NVList object to the CORBA::NVList::add(), CORBA::NVList::add_item() or CORBA::NVList::add_value() functions.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the flag of the parameter information (NamedValue object) is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.6 NVList Class

This section describes the NVList class used in the dynamic invocation interface.

```
class CORBA
typedef NamedValue_ptr *NamedValue;
class NVList
```

```
{
public:
NamedValue_ptr add_item(const char*, Flags, CORBA::Environment& );
NamedValue_ptr addvalue(const char*,
const Any&,
Flags,
CORBA::Environment& );
NamedValue_ptr add(Flags, CORBA::Environment& );
NamedValue_ptr item(Long, CORBA::Environment& );
Status remove(Long, CORBA::Environment& );
Status free_out_memory( CORBA::Environment& );
Long count( CORBA::Environment& ) const;
};
};
```

2.6.1 CORBA::NVList::add_item()

Name

CORBA::NVList::add_item

Synopsis

```
#include <orb_cplus.h>
CORBA::NamedValue_ptr  CORBA::NVList::add_item(
     const char*   item_name,
     CORBA::Flags  item_flags,
CORBA::Environment&  env );
```

Description

This function adds parameter information for the server application to be invoked, to a list object.

The parameters are appended one after another.

Parameters

item_name

The parameter name.

item_name

Specify the parameter name in item_name. Specify the following values in item_flags:

CORBA::ARG_IN

Input only parameter.

CORBA::ARG_OUT

Output only parameter.

CORBA::ARG_INOUT

Input output parameter.

CORBA::IN_COPY_VALUE

The value is copied, and the copied value is used.

CORBA::DEPENDENT_LIST

The child list is released when its parent list is released.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.6.2 CORBA::NVList::add_value()

Name

CORBA::NVList::add_value

Synopsis

```
#include <orb_cplus.h>
CORBA::NamedValue_ptr  CORBA::NVList:: add_value(
     const char*   item_name,
     const CORBA::Any&  value,
     CORBA::Flags  item_flags,
CORBA::Environment&  env );
```

Description

This function adds parameter information to a list object for the server application to be invoked.

The parameters are appended one after another.

Parameters

item_name

The parameter name.

value

The Any-type data to be used to set the parameter value.

item_flags

Specify the following values:

CORBA::ARG_IN

Input only parameter.

CORBA::ARG_OUT

Output only parameter.

CORBA::ARG_INOUT

Input output parameter.

CORBA::IN_COPY_VALUE

The value is copied, and the copied value is used.

CORBA::DEPENDENT_LIST

The child list is released when its parent list is released.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference of NamedValue object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.6.3 CORBA::NVList::add()

Name

CORBA::NVList::add

Synopsis

```
#include <orb_cplus.h>
CORBA::NamedValue_ptr  CORBA::NVList::add(
     CORBA::Flags  item_flags,
CORBA::Environment&  env );
```

Description

This function adds parameter information for the server application to be invoked, to a list object.

The parameters are appended one after another.

Parameters

item_flags

Specify the following values:

CORBA::ARG_IN

Input only parameter.

CORBA::ARG_OUT

Output only parameter.

CORBA::ARG_INOUT

Input output parameter.

CORBA::IN_COPY_VALUE

The value is copied, and the copied value is used.

CORBA::DEPENDENT_LIST

The child list is released when its parent list is released.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.6.4 CORBA::NVList::item()

Name

CORBA::NVList::item

Synopsis

```
#include <orb_cplus.h>
CORBA::NamedValue_ptr CORBA::NVList::item(
     Long count,
CORBA::Environment& env );
```

Description

This function fetches the parameter information specified in count of the list object.

Parameters

count

The index of the target parameter.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference of the NamedValue object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.6.5 CORBA::NVList::remove()

Name

CORBA::NVList::remove

Synopsis

```
#include <orb_cplus.h>
CORBA::Status  CORBA::NVList::remove(
     Long   count,
CORBA::Environment&  env );
```

Description

This function deletes the parameter information specified in count of the list object.

Parameters

count

The index of the target parameter.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.6.6 CORBA::NVList::free_out_memory()

Name

CORBA::NVList::free_out_memory

Synopsis

```
#include <orb_cplus.h>
CORBA::Status  CORBA::NVList::free_out_memory(
CORBA::Environment&  env );
```

Description

This function releases memory linked to the list object. The list object itself is not released.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

Remarks

To release not only the memory area associated with the list object, but also the list object itself, use CORBA::release().

2.6.7 CORBA::NVList::count()

Name

CORBA::NVList::count

Synopsis

```
#include <orb_cplus.h>
Long  CORBA::NVList::count(
CORBA::Environment&  env ) const;
```

Description

This function returns the total number of parameters set for the list object.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the total number of parameters is returned.

For abnormal termination, CORBA::FAILED is returned.

2.7 Context Class

This section describes the Context class that controls the context objects.

```
class CORBA
{
```

```
typedef Context_ptr *Context;
typedef NamedValue_ptr *NamedValue;
typedef NVList_ptr *NVList;
class Context
{
public:
const char *context_name(CORBA::Environment &) const;
Context_ptr parent(CORBA::Environment &) const;
Status create_child(const char *, Context _ptr&, CORBA::Environment &);
Status set_one_value(const char *, const Any &, CORBA::Environment &);
Status set_values(NVList_ptr, CORBA::Environment &);
Status get_values(const char *,
Flags,
const char *,
NVList _ptr&,
CORBA::Environment &);
Status delete_values(const char *, CORBA::Environment &);
};
};
```

2.7.1 CORBA::Context::context_name()

Name

CORBA::Context::context_name

Synopsis

```
#include <orb_cplus.h>
const char  *CORBA::Context::context_name(
CORBA::Environment&  env) const;
```

Description

This function returns the name of the Context object.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the name of the Context object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.7.2 CORBA::Context::parent()

Name

CORBA::Context::parent

Synopsis

```
#include <orb_cplus.h>
CORBA::Context_ptr  CORBA::Context::parent(
CORBA::Environment&  env) const;
```

Description

This function returns the parent of the Context object.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the parent Context object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.7.3 CORBA::Context::create_child()

Name

CORBA::Context::create_child

Synopsis

```
#include <orb_cplus.h>
CORBA::Status  CORBA::Context::create_child(
     const char     *ctx_name,
     Context _ptr&  child_ctx,
CORBA::Environment&  env );
```

Description

This function generates a Context object with the name specified in ctx_name.

Parameters

ctx_name

The name of the Context object to be created.

child_ctx

A pointer to the area that will contain the Context object.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned, and the context object reference is set in child_ctx.

For abnormal termination, CORBA::FAILED is returned.

2.7.4 CORBA::Context::set_one_value()

Name

CORBA::Context::set_one_value

Synopsis

```
#include <orb_cplus.h>
CORBA::Status  CORBA::Context::set_one_value(
```

```
     const char  *prop_name,
     const CORBA::Any     &value,
CORBA::Environment&  env );
```

Description

This method allows the application program to set an attribute value in the context object. The context object can be retrieved by calling the CORBA::ORB::get_default_context() or CORBA::Context::create_child() methods. Specify the attribute value in value. Specify the attribute name in prop_name.

Parameters

prop_name

The name of the attribute value to be set.

value

The Any-type data used to specify the attribute value to be set.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.7.5 CORBA::Context::set_values()

Name

CORBA::Context::set_values

Synopsis

```
#include <orb_cplus.h>
CORBA::Status  CORBA::Context::set_values(
     CORBA::NVList_ptr value,
CORBA::Environment& env );
```

Description

This function allows an application to set several attribute values in the context object. The context object can be retrieved by calling the CORBA::ORB::get_default_context() or CORBA::Context::create_child() methods. Multiple attribute values can be specified in value.

Parameters

value

The NVList object that contains specifications of the attribute values to be set.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.7.6 CORBA::Context::get_values()

Name

CORBA::Context::get_values

Synopsis

```
#include <orb_cplus.h>
CORBA::Status  CORBA::Context::get_values(
     const char    *start_scope,
     CORBA::Flags  op_flags,
     const char    *prop_name,
     CORBA::NVList_ptr&   value,
CORBA::Environment&  env );
```

Description

This function allows the specified attribute values to be retrieved from the context object. The context object can be retrieved by calling the CORBA::ORB::get_default_context() or CORBA::Context::create_child() methods.

Parameters

start_scope

The Context object level from which a search for the specified attribute begins.

If CORBA::CTX_RESTRICT_SCOPE is not set in op_flags, and the attribute cannot be found in the specified scope, the context tree is traced back to continue the search.

If the scope name is omitted (NULL), the search begins at the specified Context object. If the specified scope name is not found, an exception is returned.

op_flags

The following flag can be set:

CORBA::CTX_RESTRICT_SCOPE

The search is limited to the search scope specified in start_scope, or the Context object.

prop_name

Specify the Context attribute value to be acquired. If a wild card character ("*") is specified, all matching attribute names and their values are returned.

value

A pointer to the NVList object that will contain attribute values.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned, and attributes are set in value.

For abnormal termination, CORBA::FAILED is returned.

2.7.7 CORBA::Context::delete_values()

Name

CORBA::Context::delete_values

Synopsis

```
#include <orb_cplus.h>
CORBA::Status  CORBA::Context::delete_values(
```

```
    const char  *prop_name,
CORBA::Environment&  env );
```

Description

This function deletes the attribute specified at prop_name from the Context object.

Parameters

prop_name

Specify the attribute value to be deleted. If a wild card character (*) is specified, all matching attribute names and their values will be deleted.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.8 Request Class

This section describes the Request class used in the dynamic invocation interface.

```
class CORBA
{
class Request
{
public:
Object_ptr target(CORBA::Environment &) const;
const char *operation(CORBA::Environment &) const;
NVList_ptr arguments(CORBA::Environment &);
NamedValue_ptr result(CORBA::Environment &);
Environment_ptr env(CORBA::Environment &);
void ctx(Context_ptr, CORBA::Environment &);
Context_ptr ctx(CORBA::Environment &) const;
Status invoke(CORBA::Environment &);
Status send_oneway(CORBA::Environment &);
Status send_deferred(CORBA::Environment &);
Status get_response(CORBA::Environment &);
Boolean poll_response(CORBA::Environment &);
};
};
```

2.8.1 CORBA::Request::target()

Name

CORBA::Request::target

Synopsis

```
#include <orb_cplus.h>
CORBA::Object_ptr  CORBA::Request::target(
CORBA::Environment&  env ) const;
```

Description

This function returns the object reference of a request object.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the request object reference is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.8.2 CORBA::Request::operation()

Name

CORBA::Request::operation

Synopsis

```
#include <orb_cplus.h>
const char  *CORBA::Request::operation(
CORBA::Environment&  env ) const;
```

Description

This function fetches an operation name from the request object generated by the CORBA::Object::_create_request() function.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the operation name is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.8.3 CORBA::Request::arguments()

Name

CORBA::Request::arguments

Synopsis

```
#include <orb_cplus.h>
CORBA::NVList_ptr  CORBA::Request::arguments(
CORBA::Environment&  env );
```

Description

This function fetches the NVList object (the object holding parameter information when a method is invoked) from the request object generated by the CORBA::Object::_create_request() function.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the NVList object reference is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.8.4 CORBA::Request::result()

Name

CORBA::Request::result

Synopsis

```
#include <orb_cplus.h>
CORBA::NamedValue_ptr  CORBA::Request::result(
CORBA::Environment&  env );
```

Description

This function fetches the NamedValue object (the object holding a method call result) set in the request object generated by the CORBA::Object::_create_request() function.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the NVList object reference is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.8.5 CORBA::Request::env()

Name

CORBA::Request::env

Synopsis

```
#include <orb_cplus.h>
CORBA::Environment_ptr  CORBA::Request::env(
CORBA::Environment&  env );
```

Description

This function fetches the environment object set in the request object generated by the CORBA::Object::_create_request() function.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the environment object reference is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.8.6 CORBA::Request::ctx()(reference)

Name

CORBA::Request::ctx

Synopsis

```
#include <orb_cplus.h>
CORBA::Context_ptr CORBA::Request::ctx(
CORBA::Environment& env ) const;
```

Description

This function fetches the Context object set in the request object generated by the CORBA::Object::_create_request() function.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the Context object reference is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.8.7 CORBA::Request::ctx()(setup)

Name

CORBA::Request::ctx

Synopsis

```
#include <orb_cplus.h>
void  CORBA::Request::ctx(
     CORBA::Context_ptr ctx,
CORBA::Environment& env );
```

Description

This function sets the Context object specified in ctx in the request object generated by the CORBA::Object::_create_request() function.

Parameters

ctx

The Context object to be set.

env

A structure that may contain exception information.

Return Values

None.

2.8.8 CORBA::Request::invoke()

Name

CORBA::Request::invoke

Synopsis

```
#include <orb_cplus.h>
CORBA::Status CORBA::Request::invoke(
CORBA::Environment& env );
```

Description

This function calls the server application function specified when generating the request object. The result is set in the result argument specified by the CORBA::Object::_create_request() function.

The function waits until the server application program finishes.

When using the method in which user exception is defined, register the contents of the IDL file in the Interface Repository. Refer to the *IDLc* command in the Reference Manual (Command Edition).

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.8.9 CORBA::Request::send_oneway()

Name

CORBA::Request::send_oneway

Synopsis

```
#include <orb_cplus.h>
CORBA::Status CORBA::Request::send_oneway(
CORBA::Environment& env );
```

Description

This function calls the server application function specified when the request object is generated.

Unlike the CORBA::Request::invoke() function, this function returns control to the calling program, without waiting for the server application to finish.

The operation result is not returned. This function is invoked when the operation is defined as one way.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.8.10 CORBA::Request::send_deferred()

Name

CORBA::Request::send_deferred

Synopsis

```
#include <orb_cplus.h>
CORBA::Status CORBA::Request::send_deferred(
CORBA::Environment& env );
```

Description

This function calls the server application function specified when the request object is generated. The result is set in the result argument specified by the CORBA::Object::_create_request() function.

Unlike the CORBA::Request::invoke() function, this function returns control to the calling program, without waiting for the server application to finish.

The operation result is returned by the CORBA::Request::get_response() function.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.8.11 CORBA::Request::get_response()

Name

CORBA::Request::get_response

Synopsis

```
#include <orb_cplus.h>
CORBA::Status CORBA::Request::get_response(
CORBA::Environment& env );
```

Description

This function checks whether the server application function (specified when the request object is generated) has finished.

The function is invoked after the CORBA::Request::send_deferred() function.

The output parameters related to the request object and the function return value become valid when the server application function finishes.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.8.12 CORBA::Request::poll_response()

Name

CORBA::Request::poll_response

Synopsis

```
#include <orb_cplus.h>
CORBA::Status CORBA::Request::poll_response(
CORBA::Environment& env );
```

Description

This function checks whether the server application function (specified when the request object is generated) has finished.

This function is invoked after the CORBA::Request::send_deferred()function.

The output parameters related to the request object and the function return value become valid when the server application function finishes.

Unlike the CORBA::Request::get_response() function, this function returns even if the server application function is still executing.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::OK is returned.

For abnormal termination, CORBA::FAILED is returned.

2.9 ServerRequest Class

This is not valid for Linux (64 bit).

This section describes the ServerRequest class used in the dynamic skeleton interface.

```
class CORBA
{
class ServerRequest
{
public:
CORBA::Identifier      op_name(
CORBA::Environment & )
```

```
throw(CORBA::SystemException) = 0;
CORBA::Context_ptr      ctx(
CORBA::Environment & )
throw(CORBA::SystemException) = 0;
void params(
CORBA::NVList_ptr,
CORBA::Environment & )
throw(CORBA::SystemException) = 0;
void result(
CORBA::Any*,
CORBA::Environment & )
throw(CORBA::SystemException) = 0;
void exception(
CORBA::Any*,
CORBA::Environment & )
throw(CORBA::SystemException) = 0;
  }
}
```

2.9.1 CORBA::ServerRequest::op_name()

Name

CORBA::ServerRequest::op_name

Synopsis

```
#include <orb_cplus.h>
CORBA::Identifier CORBA::ServerRequest::op_name(
CORBA::Environment &env);
```

Description

This function fetches a function name from the ServerRequest object.

The object reference to be returned must be specified as the first parameter of the DSI processing function in the ServerRequest object.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the function name is returned.

For abnormal termination, the object reference of the SystemException object is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.9.2 CORBA::ServerRequest::params()

Name

CORBA::ServerRequest::params

Synopsis

```
#include <orb_cplus.h>
void CORBA::ServerRequest::params(
```

```
     CORBA::NVList_ptr  arg_list,
CORBA::Environment &env);
```

Description

This function fetches parameter information from the ServerRequest object, and sets the parameter information in the NVList object specified in arg_list.

The object reference to be returned must be specified as the first parameter of the DSI processing function in the ServerRequest object.

Parameters

arg_list

Specify the object reference created by CORBA::ORB::create_list(). You must allocate as many parameter information areas as the number of parameters using CORBA::NVList::add_item().

env

A structure that may contain exception information.

Return Values

For normal termination, the parameter information is set in the NVList object specified in arg_list.

For abnormal termination, the object reference of the SystemException object is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.9.3 CORBA::ServerRequest::result()

Name

CORBA::ServerRequest::result

Synopsis

```
#include <orb_cplus.h>
void CORBA::ServerRequest::result(
     CORBA::Any  *any_value,
CORBA::Environment &env);
```

Description

This function sets the return values of the server application method for the ServerRequest object.

Parameters

any_value

Set the any-type variable assigned the return value.

env

A structure that may contain exception information.

Return Values

For abnormal termination, the object reference of the SystemException object is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.9.4 CORBA::ServerRequest::exception()

Name

CORBA::ServerRequest::exception

Synopsis

```
#include <orb_cplus.h>
void CORBA::ServerRequest::exception(
     CORBA::Any  *any_value,
     CORBA::Environment &env);
```

Description

This function sets the server application user exceptions for the ServerRequest object.

Parameters

any_value

Specify the any-type variable assigned to the user exception.

env

A structure that may contain exception information.

Return Values

For abnormal termination, the object reference of the SystemException object is set in the exception member of the env strucure.

For details on the exception information and minor codes that are set, refer to information on Exception Information and Minor Codes Posted from the CORBA Service in the Messages manual.

2.9.5 CORBA::ServerRequest::ctx()

Name

CORBA::ServerRequest::ctx

Synopsis

```
#include <orb_cplus.h>
CORBA::Context_ptr CORBA::ServerRequest::ctx(
CORBA::Environment &env);
```

Description

This function fetches the context object corresponding to the ServerRequest object.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the context object reference is returned.

For abnormal termination, the object reference of the SystemException object is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.10 TypeCode Class

This section describes the TypeCode class used to navigate TypeCode.

```
class CORBA
{
class TypeCode
{
public:
CORBA::TCKind  kind(
CORBA::Environment & )
throw(CORBA::SystemException);
CORBA::Boolean  equal(
CORBA::TypeCode_ptr,
CORBA::Environment & )
throw(CORBA::SystemException);
}
}
```

2.10.1 CORBA::TypeCode::equal()

Name

CORBA::TypeCode::equal

Synopsis

```
#include <orb_cplus.h>
CORBA::boolean CORBA::TypeCode::equal(
     CORBA::TypeCode_ptr tc,
CORBA::Environment &env);
```

Description

This function compares the TypeCode object to the TypeCode object specified in tc.

Parameters

tc

The TypeCode object that the target TypeCode object is to be compared with.

env

A structure that may contain exception information.

Return Values

If the TypeCode objects matches, CORBA::TRUE is returned.

If the TypeCode objects do not match, CORBA::FALSE is returned.

2.10.2 CORBA::TypeCode::kind()

Name

CORBA::TypeCode::kind

Synopsis

```
#include <orb_cplus.h>
CORBA::TCKind CORBA::TypeCode::kind(
CORBA::Environment &env);
```

Description

This function returns the TypeCode object attribute information.

Attribute information contains the following:

```
CORBA::tk_null          0L
CORBA::tk_void          1L
CORBA::tk_short         2L
CORBA::tk_long          3L
CORBA::tk_ushort        4L
CORBA::tk_ulong         5L
CORBA::tk_float         6L
CORBA::tk_double        7L
CORBA::tk_boolean       8L
CORBA::tk_char          9L
CORBA::tk_octet         10L
CORBA::tk_any           11L
CORBA::tk_TypeCode      12L
CORBA::tk_Principal     13L
CORBA::tk_objref        14L
CORBA::tk_struct        15L
CORBA::tk_union         16L
CORBA::tk_enum          17L
CORBA::tk_string        18L
CORBA::tk_sequence      19L
CORBA::tk_array         20L
CORBA::tk_alias         21L
CORBA::tk_except        22L
CORBA::tk_longlong      23L
CORBA::tk_longdouble    25L
CORBA::tk_wchar         26L
CORBA::tk_wstring       27L
```

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the attribute information is returned.

For abnormal termination, CORBA::tk_null is returned.

2.11 SystemException Class

This section describes the SystemException class used in processing exceptions.

```
class CORBA
{
       class Exception{
       }

       class SystemException : public Exception
       {
       public:
               ULong minor() const;
               void minor(ULong);
       }
}
```

2.11.1 CORBA::SystemException::minor (Setup)

Name

CORBA::SystemException::minor

Synopsis

```
#include <orb_cplus.h>
void CORBA::SystemException::minor(
        CORBA::ULong minor_code)
```

Description

Sets a value in a minor member.

Parameters

minor_code

Specify the value to be set.

Return Values

None.

2.11.2 CORBA::SystemException::minor (Reference)

Name

CORBA::SystemException::minor

Synopsis

```
#include <orb_cplus.h>
CORBA::ULong CORBA::SystemException::minor()
```

Description

Obtains a value from a minor member.

Return Values

The value set in the minor member of a system exception is returned.

2.12 Naming Service Class

This section describes the functions provided by the Naming Service.

2.12.1 Type Definitions

Synopsis

```
class CosNaming {
struct NameComponent {
CORBA::String_var id;
CORBA::String_var kind;
};
class Name
{
public:
```

```
Name();
Name( ULong max );
Name(
ULong max,
ULong length,
NameComponent *value,
Boolean release = FALSE
);
Name();
private:
ULong _maximum;
ULong _length;
NameComponent *_buffer;
};
enum BindingType { nobject, ncontext } ;
struct Binding {
Name  binding_name;
BindingType binding_type;
} ;
class BindingList
 {
 public:
 BindingList();
 BindingList( ULong max );
 BindingList(
 ULong max,
 ULong length,
 Request_ptr *value,
 Boolean release = FALSE
 );
 BindingList();
 private:
 ULong _maximum;
 ULong _length;
 Binding *_buffer;
 };
};
```

2.12.2 Naming Context Interface

This section describes the Naming Context interface.

2.12.2.1 CosNaming::NamingContext::bind()

Name

CosNaming::NamingContext::bind

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
void CosNaming::NamingContext::bind(
     CosNaming::Name n,
     CORBA::Object_ptr obj,
CORBA::Environment& env );
```

Description

This function generates binding between the name specified in n, and the object reference specified in obj, and registers the binding to the naming context.

If n is a compound name, then the binding is registered in the naming context specified at the end of the compound name.

Parameters

n

The name.

obj

The object reference which binds the name specified for n.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the object reference of the UserException object is set in the exception member of the env structure. Detailed information is set in UserException_id.

The values of _id are as follows:

CosNaming::NamingContext::NotFound

The naming context specified in n could not be found.

CosNaming::NamingContext::CannotProceed

The naming context does not exist.

CosNaming::NamingContext::InvalidName

The specified Name is invalid.

CosNaming::NamingContext::AlreadyBound

The specified name is already bound.

2.12.2.2 CosNaming::NamingContext::rebind()

Name

CosNaming::NamingContext::rebind

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
void CosNaming::NamingContext::rebind(
     CosNaming::Name n,
     CORBA::Object_ptr obj,
CORBA::Environment& env );
```

Description

This function generates binding between the Name specified in n, and the naming context object reference specified in obj, and registers the binding in the naming context.

No error occurs even if the specified name is already bound. If n is a compound name, the binding is registered in the naming context specified at the end of the compound name.

Parameters

n

The name.

obj

The object reference which binds the name specified for n.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the object reference of the UserException object is set in the exception member of the env structure. Detailed information is set in UserException_id.

The values of _id are as follows:

CosNaming::NamingContext::NotFound

The naming context specified in n could not be found.

CosNaming::NamingContext::CannotProceed

The naming context does not exist.

CosNaming::NamingContext::InvalidName

The specified Name is invalid.

2.12.2.3 CosNaming::NamingContext::bind_context()

Name

CosNaming::NamingContext::bind_context

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
void CosNaming::NamingContext::bind_context(
     CosNaming::Name n,
     CosNaming::NamingContext_ptr nc,
CORBA::Environment& env );
```

Description

This function generates binding between the Name specified in n, and the naming context object reference specified in nc2, and registers the binding in the naming context. If n is a compound name, the binding is registered in the naming context specified at the end of the compound name.

Parameters

n

The name.

nc

The object reference which binds the name specified for n.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the object reference of the UserException object is set in the exception member of the env structure. Detailed information is set in UserException_id.

The values of _id are as follows:

CosNaming::NamingContext::NotFound

The naming context specified in n could not be found.

CosNaming::NamingContext::CannotProceed

The naming context does not exist.

CosNaming::NamingContext::InvalidName

The specified Name is invalid.

CosNaming::NamingContext::AlreadyBound

The specified name is already bound.

2.12.2.4 CosNaming::NamingContext::rebind_context()

Name

CosNaming::NamingContext::rebind_context

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
void CosNaming::NamingContext::rebind_context(
     CosNaming::Name n,
     CosNaming::NamingContext_ptr nc,
CORBA_Environment& env );
```

Description

This function generates binding between the Name specified in n, and the naming context object reference specified in nc2, and registers the binding in the naming context. No error occurs even if the specified name is already bound. If n is a compound name, the binding is registered in the naming context specified at the end of the compound name.

Parameters

n

The name.

nc

The object reference which binds the name specified for n.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the object reference of the UserException object is set in the exception member of the env structure. Detailed information is set in UserException_id.

The values of _id are as follows:

CosNaming::NamingContext::NotFound

The naming context specified in n could not be found.

CosNaming::NamingContext::CannotProceed

The naming context does not exist.

CosNaming::NamingContext::InvalidName

The specified Name is invalid.

2.12.2.5 CosNaming::NamingContext::resolve()

Name

CosNaming::NamingContext::resolve

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
CORBA::Object_ptr CosNaming::NamingContext::resolve(
     CosNaming::Name n,
CORBA::Environment& env );
```

Description

This function returns the object reference name specified in n that is bound in the naming context.

Since this function acquires area to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

n

The name.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the object reference of the UserException object is set in the exception member of the env structure. Detailed information is set in UserException_id.

The values of _id are as follows:

CosNaming::NamingContext::NotFound

The naming context specified in n could not be found.

CosNaming::NamingContext::CannotProceed

The naming context does not exist.

CosNaming::NamingContext::InvalidName

The specified Name is invalid.

2.12.2.6 CosNaming::NamingContext::unbind()

Name

CosNaming::NamingContext::unbind

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
void CosNaming::NamingContext::unbind(
```

```
    CosNaming::Name n,
CORBA::Environment& env );
```

Description

This function deletes the binding of the object reference Name specified in n, from the naming context.

Parameters

n

The name which deletes the binding of the naming context.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the object reference of the UserException object is set in the exception member of the env structure. Detailed information is set in UserException_id.

The values of_id are as follows:

CosNaming::NamingContext::NotFound

The naming context specified in n could not be found.

CosNaming::NamingContext::CannotProceed

The naming context does not exist.

CosNaming::NamingContext::InvalidName

The specified Name is invalid.

2.12.2.7 CosNaming::NamingContext::new_context()

Name

CosNaming::NamingContext::new_context

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
CosNaming::NamingContext_ptr CosNaming::NamingContext::new_context(
CORBA::Environment& env);
```

Description

This function generates a new naming context in the naming server that manages naming contexts, and returns the object reference of the naming context.

Since this function acquires area to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the naming context object reference is returned, and a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the UserException object reference is set in the exception member of the env structure.

Notes

- Fujitsu recommends using the CosNaming::NamingContext::bind_new_context() method to create and register a naming context object.
- When you create a naming context object using this method, create the binding, then register it in the naming context using the CosNaming::NamingContext::bind_context() method.

2.12.2.8 CosNaming::NamingContext::bind_new_context()

Name

CosNaming::NamingContext::bind_new_context

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
CosNaming::NamingContext_ptr CosNaming::NamingContext::bind_new_context(
     CosNaming_Name n,
     CORBA_Environment *env);
```

Description

This function generates a new naming context and binding between the object reference of the generated context, and the name specified at n, and registers them in the existing naming context. The function returns the object reference of the new naming context.

If n is a compound name, the binding is registered in the naming context specified at the end of the compound name.

The naming context where Name is registered is generated in the naming server that manages naming contexts.

Since this function acquires area to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

n

The name which binds the object reference of a new naming context.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference of a naming context is returned, and a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the UserException object reference is set in the exception member of the env structure. Detailed information is set in UserException_id.

The values of _id are as follows:

CosNaming::NamingContext::NotFound

The naming context specified in n could not be found.

CosNaming::NamingContext::CannotProceed

The naming context does not exist.

CosNaming::NamingContext::InvalidName

The specified Name is invalid.

CosNaming::NamingContext::AlreadyBound

The specified name is already bound.

2.12.2.9 CosNaming::NamingContext::destroy()

Name

CosNaming::NamingContext::destroy

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
void CosNaming::NamingContext::destroy(
CORBA::Environment& env );
```

Description

This function deletes the naming context.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the UserException object reference is set in the exception member of the env structure. Detailed information is set in UserException_id.

The value of _id is as follows:

CosNaming::NamingContext::NotEmpty

The binding exists in the naming context.

2.12.2.10 CosNaming::NamingContext::list()

Name

CosNaming::NamingContext::list

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
void CosNaming::NamingContext::list(
     CORBA_ULong how_many,
     CosNaming::BindingList *&bl,
     CosNaming::BindingIterator_ptr bi,
     CORBA::Environment& env );
```

Description

This function returns a list of bindings from a NamingContext object according to the number specified in how_many. If the value specified in how_many is greater than the number of bindings specified in the bl_how_many parameter of the nsconfig file (refer to the *nsconfig* command), the Naming Service returns the maximum number of individual bindings specified in the bl_how_many parameter. If 0 is specified in how_many, the client returns bi for access to bindings and a sequence bl of 0 length.

The list (CosNaming::BindingList) is returned in bl.

Since this function acquires area to store the binding iterator, use CosNaming::BindingIterator::destroy() to release the area as soon as it is no longer needed.

Parameters

how_many

The number of obtained bindings lists.

bl

The set area of the bindings list.

bi

The set area of the bindings iterator.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the UserException object reference is set in the exception member of the env structure.

2.12.2.11 CosNaming::NamingContext::_narrow()

Name

CosNaming::NamingContext::_narrow

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
CosNaming::NamingContext_ptr
CosNaming::NamingContext::_narrow( CORBA::Object_ptr obj );
```

Description

The object reference acquired from the Naming Service is converted into the NamingContext class.

Parameters

obj

The object reference.

Return Values

For normal termination, the object reference is returned, and CORBA_NO_EXCEPTION is set in _major of the env structure.

For abnormal termination, CORBA_SYSTEM_EXCEPTION or CORBA_USER_EXCEPTION is set in _major of the env structure, and the detailed information is set in _id of the env structure.

For details on exception information and minor codes that are set, refer to 'Exception Information Minor Codes to be Reported from the CORBA Service' in the Messages manual.

2.12.3 Binding Iterator Interface

This section describes the binding iterator interface.

2.12.3.1 CosNaming::BindingIterator::next_one()

Name

CosNaming::BindingIterator::next_one

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
CORBA::boolean CosNaming::BindingIterator::next_one(
     CosNaming::Binding& *b,
CORBA::Environment& env );
```

Description

This function fetches the next binding from the current position in the naming context displayed by the binding iterator, and stores it in b.

Since this function acquires area to store the binding iterator, use CosNaming::BindingIterator::destroy() to release the area as soon as it is no longer needed.

Parameters

b

The set area of the bindings.

env

A structure that may contain exception information.

Return Values

For normal termination with more than one valid binding returned, the function returns CORBA::TRUE. If non-listed bindings exist, the function returns CORBA::FALSE.

For abnormal termination, SystemException is set in the exception method of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.12.3.2 CosNaming::BindingIterator::next_n()

Name

CosNaming::BindingIterator::next_n

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
CORBA::boolean CosNaming::BindingIterator::next_n(
     CORBA::ULong how_many,
     CosNaming::BindingList *&bl,
CORBA::Environment& env );
```

Description

This function retrieves bindings from the current position of the naming context indicated by the binding iterator up to the number specified in how_many, and stores them in bi. If the value specified in how_many is greater than the number of bindings specified in the bl_how_many parameter of the nsconfig file (refer to the *nsconfig* command), the Naming Service returns the maximum number of individual bindings specified in the bl_how_many parameter. If 0 is specified in how_many, a BAD_PARAM system exception occurs.

Since this function acquires area to store the binding iterator, use CosNaming::BindingIterator::destroy() to release the area as soon as it is no longer needed.

Parameters

how_many

The number of obtained bindings lists.

bl

The set area of the bindings list.

env

A structure that may contain exception information.

Return Values

For normal termination with more than one valid binding returned, the function returns CORBA::TRUE. If non listed bindings exist, the function returns CORBA::FALSE.

For abnormal termination, SystemException is set in the exception method of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.12.3.3 CosNaming::BindingIterator::destroy()

Name

CosNaming::BindingIterator::destroy

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
void CosNaming::BindingIterator::destroy(
CORBA::Environment& env );
```

Description

This function destroys the binding iterator.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the UserException object reference is set in the exception member of the env structure. Detailed information is set in UserException_d.

2.12.4 Naming Context Extended Interface

This section details the functions provided by the Naming Context Extended interface.

2.12.4.1 CosNaming::NamingContextExt::to_string()

Name

CosNaming::NamingContextExt::to_string

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
CosNaming::NamingContextExt::StringName CosNaming::NamingContextExt::to_string(
        CosNaming::Name n,
        CORBA::Environment& env );
```

Description

This function converts the structural type binding name specified in n into a character string binding name.

Parameters

n

The structural type binding name.

env

A structure that may contain exception information.

Return Values

For normal termination, sets a NULL object reference in the env exception member.

For abnormal termination, sets a UserException object reference in the env exception member. Detailed information is set in UserException_id.

The value and meaning of _id are as follows:

CosNaming::NamingContext::InvalidName

Error in name specification.

2.12.4.2 CosNaming::NamingContextExt::to_name()

Name

CosNaming::NamingContextExt::to_name

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
CosNaming::Name *CosNaming::NamingContextExt::to_name(
        CosNaming::NamingContextExt::StringName sn,
        CORBA::Environment& env );
```

Description

This function converts the character string type binding name specified in sn into a structural type binding name.

Parameters

sn

The character string type binding name.

env

A structure that may contain exception information.

Return Values

For normal termination, sets a NULL object reference in the env exception member.

For abnormal termination, sets a UserException object reference in the env exception member. Detailed information is set in UserException_id.

The value and meaning of _id are as follows:

CosNaming::NamingContext::InvalidName

Error in name specification.

2.12.4.3 CosNaming::NamingContextExt::to_url()

Name

CosNaming::NamingContextExt::to_url

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
CosNaming::NamingContextExt::URLString CosNaming::NamingContextExt::to_url(
        CosNaming::NamingContextExt::Address addrkey,
        CosNaming::NamingContextExt::StringName sn,
        CORBA::Environment& env );
```

Description

This function creates a URL schema from the address specified in addrkey and the character string binding name specified in sn.

Parameters

addrkey

The address of the naming context.

sn

The character string type binding name.

env

A structure that may contain exception information.

Return Values

For normal termination, sets a NULL object reference in the env exception member.

For abnormal termination, sets a UserException object reference in the env exception member. Detailed information is set in UserException_id.

The value and meaning of _id are as follows:

CosNaming::NamingContext::InvalidAddress

Error in address specification.

CosNaming::NamingContext::InvalidName

Error in name specification.

2.12.4.4 CosNaming::NamingContextExt::resolve_str()

Name

CosNaming::NamingContextExt::resolve_str

Synopsis

```
#include <orb_cplus.h>
#include <CosNaming_cplus.h>
CORBA::Object CosNaming::NamingContextExt::resolve_str(
        CosNaming::NamingContextExt::StringName sn,
        CORBA::Environment& env );
```

Description

This function returns to the naming context specified in nce the object reference bound to the character string binding name specified in sn.

Parameters

sn

The character string type binding name.

env

A structure that may contain exception information.

Return Values

For normal termination, sets a NULL object reference in the env exception member.

For abnormal termination, sets a UserException object reference in the env exception member. Detailed information is set in UserException_id.

The value and meaning of _id are as follows:

CosNaming::NamingContext::NotFound

The name specified in sn could not be found.

CosNaming::NamingContext::CannotProceed

The naming context does not exist.

CosNaming::NamingContext::InvalidName

Error in name specification.

2.13 Interface Repository Class

This section describes the functions provided by the Interface Repository.

2.13.1 Type Definitions

Synopsis

```
class CORBA
{
typedef CORBA::String_var  Identifier;          /* Identifier */
typedef CORBA::String_var  ScopedName;          /* Scope name */
typedef CORBA::String_var  RepositoryId;        /* Global name */
enum  DefinitionKind {                          /* Object kind */
dk_none, dk _all,
dk_Attribute, dk_Constant, dk_Exception, dk_Interface,
dk_Module, dk_Operation, dk_Typedef,
dk_Alias, dk_Struct, dk_Union, dk_Enum,
dk_Primitive, dk_String, dk_Sequence, dk_Array,
dk_Repository, dk_Wstring
} ;
typedef CORBA::Stringvar  VersionSpec;          /* Version information */
```

```
typedef CORBA::Object_ptr  IRObject;            /* Object ref of IRObject */
typedef CORBA::Object_ptr  Contained;           /*Contained object ref */
typedef CORBA::Object_ptr  Container;           /* Container object ref */
typedef CORBA::Object_ptr  IDLType;             /* IDLType object ref */
typedef CORBA::Object_ptr  Repository;          /* Repository object ref */
typedef CORBA::Object_ptr  ModuleDef;           /* ModuleDef object ref */
typedef CORBA::Object_ptr  ConstantDef;         /* ConstantDef object ref */
typedef CORBA::Object_ptr  TypedefDef;          /* TypedefDef object ref */
typedef CORBA::Object_ptr  StructDef;           /* StructDef  object ref */
typedef CORBA::Object_ptr  UnionDef;            /* UnionDef object ref */
typedef CORBA::Object_ptr  EnumDef;             /* EnumDef  object ref */
typedef CORBA::Object_ptr  AliasDef;            /* AliasDef object ref */
typedef CORBA::Object_ptr  PrimitiveDef;        /* PrimitiveDef object ref */
typedef CORBA::Object_ptr  StringDef;           /* StringDef object ref */
typedef CORBA::Object_ptr  SequenceDef;         /* SequenceDef object ref */
typedef CORBA::Object_ptr  ArrayDef;            /* ArrayDef object ref */
typedef CORBA::Object_ptr  ExceptionDef;        /* ExceptionDef object ref */
typedef CORBA::Object_ptr  AttributeDef;        /* AttributeDef object ref */
typedef CORBA::Object_ptr  OperationDef;        /* OperationDef object ref
 */
typedef CORBA::Object_ptr  InterfaceDef;        /* InterfaceDef object ref
*/
typedef CORBA::Object_ptr  WstringDef;          /* WstringDef object ref */
typedef struct {                                /* Object information structure */
CORBA::DefinitionKind  kind;                    /* Object kind */
CORBA::Any  value;                              /* Information classified by object */
} Description;
typedef struct {
ULong  _maximum;
ULong  _length;
CORBA::Description  *_buffer;
} DescriptionSeq;                               /* Description structure
 sequence */
typedef struct {
    ULong  _maximum;
    ULong  _length;
    CORBA::Description  *_buffer;
  } DescriptionSeq;                 /* Description structure sequence */
class ContainedSeq
{
public:
ContainedSeq();
ContainedSeq( ULong max );
ContainedSeq(
ULong  max,
ULong  length,
CORBA::Contained_ptr  *value,
Boolean  release = FALSE
);
~ContainedSeq();
private:
ULong  _maximum;
ULong  _length;
CORBA::Contained_ptr  *_buffer;
};                                          /* Contained object ref
sequence */
typedef struct {                            /* Structure member structure
*/
CORBA::Identifer  name;                     /* Identifier */
CORBA::TypeCode_ptr  type;                  /* Type code */
CORBA::IDLType_ptr  type_def;               /* Member object ref */
} StructMember;
class  StructMemberSeq
```

```
{
public:
StructMemberSeq();
StructMemberSeq( ULong max );
StructMemberSeq(
ULong  max,
ULong  length,
CORBA::StructMember  *value,
Boolean release = FALSE
);
~StructMemberSeq();
private:
ULong _maximum;
ULong _length;
CORBA::StructMember *_buffer;
};                                          /* Structure member structure sequence*/
typedef struct {                            /* union member structure */
CORBA::Identifer  name;                     /* Identifier */
CORBA::Any  label;                          /* Discriminative value */
CORBA::TypeCode_ptr  type;                  /* Type code */
CORBA::IDLType_ptr  type_def;               /* Member object reference */
} UnionMember;
class  UnionMemberSeq
{
public:
UnionMemberSeq();
UnionMemberSeq( ULong max );
UnionMemberSeq(
ULong  max,
ULong  length,
CORBA::UnionMember  *value,
Boolean release = FALSE
);
~UnionMemberSeq();
private:
ULong _maximum;
ULong _length;
CORBA::UnionMember *_buffer;
};                                          /* union member structure sequence*/
class  EnumMemberSeq
{
public:
EnumMemberSeq();
EnumMemberSeq( ULong max );
EnumMemberSeq(
ULong  max,
ULong  length,
CORBA::Identifier  *value,
Boolean release = FALSE
);
~EnumMemberSeq();
private:
ULong  _maximum;
ULong  _length;
CORBA::Identifier  *_buffer;
} ;                                         /* Enum member sequence */
 enum PrimitiveKind {
pk_null, pk_void, pk_short, pk_long, pk_ushort, pk_ulong,
pk_float, pk_double, pk_boolean, pk_char, pk_octet,
pk_any, pk_TypeCode, pk_Principal, pk_string, pk_objref,
pk_longlong, pk_ulonglong, pk_longdouble, pk_wchar,
pk_wstring
};
```

```
enum  AttributeMode { ATTR_NORMAL, ATTR _READONLY } ;    /* Attribute type */
typedef  struct {                              /* AttributeDef information structure */
CORBA::Identifer  name;                        /* Identifier */
CORBA::RepositoryId  id;                       /* Global name */
CORBA::RepositoryId  defined_in;               /* Global name of parent object */
CORBA::VersionSpec  version;                   /* Version information */
CORBA::TypeCode_ptr  type;                     /* Type code */
CORBA::AttributeMode  mode;                    /* Attribute */
} AttributeDescription;
enum  OperationMode { OP_NORMAL, OP _ONEWAY };
/* Operation attribute kind */
enum  ParameterMode { PARAM_IN, PARAM _OUT, PARAM _INOUT};
/* Parameter attribute kind */
typedef  struct {                              /* Parameter information structure */
CORBA::Identifer  name;                        /* Identifier */
CORBA::TypeCode_ptr  type;                     /* Type code */
CORBA::ParameterMode  mode;                    /* Attribute */
} ParameterDescription;
class  ParDescriptionSeq
{
public:
ParDescriptionSeq();
ParDescriptionSeq( ULong max );
ParDescriptionSeq(
ULong  max,
ULong  length,
CORBA::ParameterDescription  *value,
Boolean release = FALSE
);
~ParDescriptionSeq();
private:
ULong  _maximum;
ULong  _length;
CORBA::ParameterDescription  *_buffer;
};                                      /* Parameter information structure sequence */
typedef CORBA::String_var  ContextIdentifier;
class  ContextIdSeq
{
public:
ContextIdSeq();
ContextIdSeq(
ULong  max,
ULong  length,
CORBA::ContextIdentifier  *value,
Boolean release = FALSE
);
ContextIdSeq();
private:
ULong  _maximum;
ULong  _length;
CORBA::ContextIdentifier  *_buffer;
};                                        /* ContextIdentifier sequence */
class ExceptionDefSeq
{
public:
ExceptionDefSeq();
ExceptionDefSeq( ULong max );
ExceptionDefSeq(
ULong  max,
ULong  length,
CORBA::ExceptionDef_ptr  *value,
Boolean release = FALSE
);
```

```
~ExceptionDefSeq();
private:
ULong  _maximum;
ULong  _length;
CORBA::ExceptionDef_ptr  *_buffer;
};                                      /* ExceptionDef object reference sequence */
class ExcDescriptionSeq
{
public:
ExcDescriptionSeq();
ExcDescriptionSeq( ULong max );
ExcDescriptionSeq(
ULong  max,
ULong  length,
CORBA::ExceptionDescription  *value,
Boolean  release = FALSE
);
~ExcDescriptionSeq();
private:
ULong  _maximum;
ULong  _length;
CORBA::ExceptionDescription  *_buffer;
};                                    /* ExceptionDef information structure sequence  */
typedef struct {                        /* ExceptionDef information structure  */
CORBA::Identifer  name;                 /* identifier name */
CORBA::RepositoryId  id;                /* global name */
CORBA::RepositoryId  defined_in;        /* new object global name*/
CORBA::VersionSpec version;             /* version information */
CORBA::TypeCode_ptr  type;              /* type code */
CORBA::OperationMode  mode;             /* attribute */
CORBA::ContextIdSeq  contexts;          /* context */
CORBA::ParDescriptionSeq  parameters;   /* parameter information*/
CORBA::ExcDescriptionSeq  exceptions;   /* Exception information */
} OperationDescription;
class OpDescriptionSeq
{
public:
OpDescriptionSeq();
OpDescriptionSeq( ULong max );
OpDescriptionSeq(
ULong  max,
ULong  length,
CORBA::OperationDescription  *value,
Boolean  release = FALSE
);
~OpDescriptionSeq();
private:
ULong  _maximum;
ULong  _length;
CORBA::OperationDescription  *_buffer;
};                                      /* OperationDef info structure sequence */
class AttrDescriptionSeq
{
public:
AttrDescriptionSeq();
AttrDescriptionSeq( ULong max );
AttrDescriptionSeq(
ULong  max,
ULong  length,
CORBA::AttributeDescription  *value,
Boolean  release = FALSE
);
~AttrDescriptionSeq();
```

```
private:
ULong  _maximum;
ULong  _length;
CORBA::AttributeDescription  *_buffer;
};
typedef struct {
CORBA_long  _maximum;
CORBA_long  _length;
CORBA_Description  *_buffer;
} CORBA_AttrDescriptionSeq;              /* AttributeDef info structure sequence */
typedef struct {                         /* Interface information structure */
CORBA::Identifer  name;                  /* Identifier */
CORBA::RepositoryId  id;                 /* Global name */
CORBA::RepositoryId  defined_in;         /* Global name of parent object */
CORBA::VersionSpec  version;             /* Version information */
CORBA::OpDescriptionSeq  operations;     /* Operation information */
CORBA::AttrDescriptionSeq  attributes;   /* Attribute information */
CORBA::RepositoryIdSeq base_interfaces;  /* Inherited information */
CORBA::TypeCode_ptr  type;               /* Type code */
} FullInterfaceDescription;
```

2.13.2 IRObject Common Interface

This section explains the functions that each Interface Repository interface object can inherit and use as its own functions.

2.13.2.1 CORBA::IRObject::def_kind()

Name

CORBA::IRObject::def_kind

Synopsis

```
#include <orb_cplus.h>
CORBA::DefinitionKind  CORBA::IRObject::def_kind(
     CORBA::Environment& env );
```

Description

This function returns the interface type of an interface object.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the interface type of an interface object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.3 Contained Common Interface

This section explains the functions that can be inherited by interface objects that can be contained in other interface objects, and used as functions of those interface objects. Interface objects contained within other interface objects are called Contained interface objects.

2.13.3.1 CORBA::Contained::id()

Name

CORBA::Contained::id

Synopsis

```
#include <orb_cplus.h>
CORBA::RepositoryId  CORBA::Contained::id(
     CORBA::Environment& env );
```

Description

This function returns the repository ID of the interface object.

Since this function acquires area to store the repository ID, use CORBA::string_free() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the repository ID of the interface object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.3.2 CORBA::Contained::name()

Name

CORBA_Contained::name

Synopsis

```
#include <orb_cplus.h>
CORBA::Identifier  CORBA::Contained::name(
     CORBA::Environment& env ) ;
```

Description

This function returns the Name of an interface object.

Since this function acquires area to store the object name, use CORBA::string_free() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the interface object name (CORBA::Identifier string type) is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.3.3 CORBA::Contained::defined_in()

Name

CORBA::Contained::defined_in

Synopsis

```
#include <orb_cplus.h>
CORBA::Container_ptr  CORBA::Contained::defined_in(
     CORBA::Environment& env ) ;
```

Description

This function returns the object reference of an object (Container) containing the interface object.

Since this function acquires area to store the object reference, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.3.4 CORBA::Contained::describe()

Name

CORBA::Contained::describe

Synopsis

```
#include <orb_cplus.h>
CORBA::Contained::Description  *describe(
CORBA::Environment& env );
```

Description

This function returns CORBA::Description showing the object interface information.

This information differs with each interface object. The unique information of each interface object is held at the value member of CORBA::Description. Value is a CORBA::Any type. The following structure addresses are set at the _value member, depending on the interface object type. Refer to 2.13.1 Type Definitions for details of each structure.

CORBA::ModuleDef object

ModuleDescription structure

CORBA_ConstantDef object

ConstantDescription structure

CORBA_StructDef object

TypeDescription structure

CORBA_UnionDef object

TypeDescription structure

CORBA_EnumDef object

TypeDescription structure

CORBA_AliasDef object

TypeDescription structure

CORBA_ExceptionDef object

ExceptioDescription structure

CORBA_AttributeDef object

AttributDescription structure

CORBA_OperationDef object

OperationDescription structure

CORBA_InterfaceDef object

InterfacDescription structure

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA::Description is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.3.5 Inherited Usable Functions

(1) CORBA::Contained::def_kind()

Refer to 2.13.2 IRObject Common Interface for details about (1).

2.13.4 Container Common Interface

This section explains the functions that can be inherited by those interface objects that can be contained within other interface objects, and used as functions of those interface objects. Interface objects contained within other interface objects are called Contained interface objects.

2.13.4.1 CORBA::Container::lookup()

Name

CORBA::Container::lookup

Synopsis

```
#include <orb_cplus.h>
CORBA::Contained_ptr  CORBA::Container::lookup(
     CORBA::ScopedName search_name,
     CORBA::Environment& env ) ;
```

Description

This function searches for the interface objects contained within an interface object of a specific name, and returns its object reference.

If the object to be returned, cannot be found, object reference of NIL (null) is returned and operation ends normally.

Since this function acquires area to store the object reference, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

search_name

The name (ScopedName) to be used as the retrieval key.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference of the specified object is returned.

For abnormal termination, a NIL (empty) object reference is returned. The SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.4.2 CORBA::Container::contents()

Name

CORBA::Container::contents

Synopsis

```
#include <orb_cplus.h>
CORBA::ContainedSeq  *CORBA::Container::contents(
     CORBA::DefinitionKind limit_type,
     CORBA::boolean exclude_inherited,
     CORBA:Environment& env );
```

Description

This function returns an object reference list of the interface objects contained within the specified interface object directly, or by inheritance.

If the object to be returned cannot be found, 0 is set to the length of the return list and the object reference becomes undefined.

Parameters

limit_type

The included objects of the interface type.

If the limit_type is dk_all and exclude_inherited is FALSE, the list includes all the objects contained or inherited.

exclude_inherited

TRUE:

The inherited object is not targeted in the retrieval.

FALSE:

The inherited object is targeted in the retrieval.

env

A structure that may contain exception information.

Return Values

For normal termination, an object reference list of the retrieved objects is returned.

For abnormal termination, a NIL (empty) object reference list is returned. The SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.4.3 CORBA::Container::lookup_name()

Name

CORBA::Container::lookup_name

Synopsis

```
#include <orb_cplus.h>
CORBA::ContainedSeq  *CORBA::Container::lookup _name(
     CORBA::Identifier search_name,
     CORBA::Long levels_to_search,
     CORBA::DefinitionKind limit_type,
     CORBA::Boolean exclude_inherited,
     CORBA::Environment& env ) ;
```

Description

This function returns an object reference list of the interface objects contained within the specified interface object directly, or by inheritance.

If the object to be returned cannot be found, 0 is set to the length of the return list and the object reference becomes undefined.

Parameters

search_name

The retrieval key name (identifier).

levels_to_search

The depth of the retrieval hierarchy to search at levels. If -1 is specified, all hierarchical levels are searched. If 1 is specified, only the objects below the specified hierarchical level are searched.

limit_type

The list includes the contained objects of the interface type.

If the limit_type is dk_all and exclude_inherited is FALSE, the list includes all the objects contained or inherited.

exclude_inherited

TRUE:

The inherited object is not targeted in the retrieval.

FALSE:

The inherited object is targeted in the retrieval.

env

A structure that may contain exception information.

Return Values

For normal termination, an object reference list of the retrieved objects is returned.

For abnormal termination, a NIL (empty) object reference list is returned. The SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.4.4 CORBA::Container::describe_contents()

Name

CORBA::Container::describe_contents

Synopsis

```
#include <orb_cplus.h>
CORBA::DescriptionSeq  *CORBA::Container::describe _contents(
     CORBA::DefinitionKind limit_type,
     CORBA::Boolean exclude_inherited,
     CORBA::Long max_returned_objs,
     CORBA::Environment& env ) ;
```

Description

This function returns the definition information of objects contained in a specified interface object directly, or by inheritance.

The information is returned in Description list format (refer to the 2.13.1 Type Definitions).

If the object to be returned cannot be found, 0 is set to the length of the return list and the object reference becomes undefined.

Parameters

limit_type

The list includes the contained objects of the interface type

If limit_type is dk_all and exclude_inherited is FALSE, the list includes all the objects contained or inherited. Definition information is returned for the number of objects specified in max_returned_objs. Specify -1 in max_returned_objs to acquire definition information for all registered objects.

exclude_inherited

TRUE:

The inherited object is not targeted in the retrieval.

FALSE:

The inherited object is targeted in the retrieval.

max_returned_objs

The number of objects whose definition information is to be returned.

env

A structure that may contain exception information.

Return Values

For normal termination, an object definition information list is returned.

For abnormal termination, a NIL (empty) definition information list is returned. The SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.4.5 Functions Usable when Inherited

(1) CORBA::Container::def_kind

Refer to 2.13.2 IRObject Common Interface for details about (1).

2.13.5 IDLType Common Interface

This section describes the IDL type common interface.

2.13.5.1 CORBA::IDLType::type()

Name

CORBA::IDLType::type

Synopsis

```
#include <orb_cplus.h>
CORBA::TypeCode_ptr CORBA::IDLType::type (
     CORBA::Object obj,
     CORBA::Environment& env ) ;
```

Description

This function returns the type code of an IDLType object.

Since this function acquires area to store the type code, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

obj

The IDLType object.

env

A structure that may contain exception information.

Return Values

For normal termination, the type code of the object is set.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.5.2 Functions Usable when Inherited

(1) CORBA::IDLType::def_kind

Refer to 2.13.2 IRObject Common Interface for details about (1).

2.13.6 Repository Interface

This section describes the Interface Repository interface.

2.13.6.1 CORBA::Repository::lookup_id()

Name

CORBA::Repository::lookup_id

Synopsis

```
#include <orb_cplus.h>
CORBA:Contained_ptr  CORBA::Repository::lookup_id(
     CORBA::RepositoryId search_id,
     CORBA::Environment& env ) ;
```

Description

This function searches an object with the Repository ID specified in search_id, and returns its object reference.

If the object to be returned, cannot be found, object reference of NIL (null) is returned and operation ends normally.

Since this function acquires area to store the type code, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

search_id

The object with retrieved Repository ID.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference is returned.

For abnormal termination, a NIL (empty) object reference is returned. The SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.6.2 Functions Usable when Inherited

(1) CORBA::Repository::def_kind

(2) CORBA::Repository::lookup

(3) CORBA::Repository::contents

(4) CORBA::Repository::lookup_name

(5) CORBA::Repository::describe_contents

Refer to 2.13.2 IRObject Common Interface for details about (1).

Refer to 2.13.4 Container Common Interface for details about (2) to (5).

2.13.7 ModuleDef Interface

This section describes the ModuleDef interface.

2.13.7.1 Functions Usable when Inherited

(1) CORBA::ModuleDef::def_kind

(2) CORBA::ModuleDef::id

(3) CORBA::ModuleDef::name

(4) CORBA::ModuleDef::defined_in

(5) CORBA::ModuleDef::describe

(6) CORBA::ModuleDef::lookup

(7) CORBA::ModuleDef::contents

(8) CORBA::ModuleDef::lookup_name

(9) CORBA::ModuleDef::describe_contents

Refer to 2.13.2 IRObject Common Interface for details about (1).

Refer to 2.13.3 Contained Common Interface for details about (2) to (5).

Refer to 2.13.4 Container Common Interface for details about (6) to (9).

2.13.8 ConstantDef Interface

This section describes the ConstantDef interface.

2.13.8.1 CORBA::ConstantDef::type()

Name

CORBA::ConstantDef::type

Synopsis

```
#include <orb_cplus.h>
CORBA::TypeCode_ptr  CORBA::ConstantDef::type(
     CORBA::Environment& env ) ;
```

Description

This function returns the type code of a constant definition object.

Since this function acquires area to store the type code, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the type code of the constant definition object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.8.2 CORBA::ConstantDef::value()

Name

CORBA_ConstantDef::value

Synopsis

```
#include <orb_cplus.h>
CORBA::Any*  CORBA::ConstantDef::value(
     CORBA::Environment& env ) ;
```

Description

This function returns the constant value of the ConstantDef object in a (CORBA::Any) structure.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a constant value (any structure) is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.8.3 Functions Usable when Inherited

(1) CORBA::ConstantDef::def_kind

(2) CORBA::ConstantDef::id

(3) CORBA::ConstantDef::name

(4) CORBA::ConstantDef::defined_in

(5) CORBA::ConstantDef::describe

Refer to 2.13.2 IRObject Common Interface for details about (1).

Refer to 2.13.3 Contained Common Interface for details about (2) to (5).

2.13.9 StructDef Interface

This section describes the StructDef interface.

2.13.9.1 CORBA::StructDef::members()

Name

CORBA::StructDef::members

Synopsis

```
#include <orb_cplus.h>
CORBA::StructMemberSeq  *CORBA::StructDef::members(
     CORBA::Environment& env );
```

Description

This function returns the member information of a StructDef object in a StructMember list format (refer to 2.13.1 Type Definitions).

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the member information (StructMember) of a StructDef object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.9.2 Functions Usable when Inherited

(1) CORBA::StructDef::def_kind

(2) CORBA::StructDef::id

(3) CORBA::StructDef::name

(4) CORBA::StructDef::defined_in

(5) CORBA::StructDef::describe

(6) CORBA::StructDef::type

Refer to 2.13.2 IRObject Common Interface for details about (1).

Refer to 2.13.3 Contained Common Interface for details about (2) to (5)

Refer to 2.13.2 IRObject Common Interface for details about (6).

2.13.10 UnionDef Interface

This section describes the UnionDef interface.

2.13.10.1 CORBA::UnionDef::discriminator_type()

Name

CORBA::UnionDef::discriminator_type

Synopsis

```
#include <orb_cplus.h>
CORBA::TypeCode_ptr  CORBA::UnionDef::discriminator_type(
     CORBA::Environment& env ) ;
```

Description

This function returns the TypeCode of a UnionDef object discriminator definition.

Since this function acquires area to store the type code, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the type code of the discriminator definition is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.10.2 CORBA::UnionDef::members()

Name

CORBA::UnionDef::members

Synopsis

```
#include <orb_cplus.h>
CORBA::UnionMemberSeq  *CORBA::UnionDef::members(
     CORBA::Environment& env ) ;
```

Description

This function returns the member information of a UnionDef object in UnionMember list format (refer to 2.13.1 Type Definitions).

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the member information (UnionMember) of the UnionDef object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.10.3 Functions Usable when Inherited

(1) CORBA::UnionDef::def_kind

(2) CORBA::UnionDef::id

(3) CORBA::UnionDef::name

(4) CORBA::UnionDef::defined_in

(5) CORBA::UnionDef::describe

Refer to 2.13.2 IRObject Common Interface for details about (1).

Refer to 2.13.3 Contained Common Interface for details about (2) to (5).

2.13.11 EnumDef Interface

This section describes the EnumDef interface.

2.13.11.1 CORBA::EnumDef::members()

Name

CORBA::EnumDef::members

Synopsis

```
#include <orb_cplus.h>
CORBA::EnumMemberSeq  *CORBA::EnumDef::members(
     CORBA::Environment& env ) ;
```

Description

This function returns the member information list of an EnumDef object (refer to 2.13.1 Type Definitions).

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the member information list (UnionMember) of the EnumDef object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.11.2 Functions Usable when Inherited

(1) CORBA::EnumDef::def_kind

(2) CORBA::EnumDef::id

(3) CORBA::EnumDef::name

(4) CORBA::EnumDef::defined_in

(5) CORBA::EnumDef::describe

Refer to 2.13.2 IRObject Common Interface for details about (1).

Refer to 2.13.3 Contained Common Interface for details about (2) to (5).

2.13.12 AliasDef Interface

This section describes the AliasDef interface.

2.13.12.1 CORBA::AliasDef::original_type_def()

Name

CORBA::AliasDef::original_type_def

Synopsis

```
#include <orb_cplus.h>
#include "InterfaceRep_cplus.h"
CORBA::IDLType_ptr  original_type_def(
     CORBA::Environment & env );
```

Description

This function returns the object reference of an AliasDef object original data type.

Since this function acquires area to store the object reference, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference of the AliasDef object original data type is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.12.2 Functions Usable when Inherited

(1) CORBA::AliasDef::def_kind

(2) CORBA::AliasDef::id

(3) CORBA::AliasDef::name

(4) CORBA::AliasDef::defined_in

(5) CORBA::AliasDef::describe

Refer to 2.13.2 IRObject Common Interface for details about (1).

Refer to 2.13.3 Contained Common Interface for details about (2) to (5).

2.13.13 StringDef Interface

This section describes the StringDef interface.

2.13.13.1 CORBA::StringDef::bound()

Name

CORBA::StringDef::bound

Synopsis

```
#include <orb_cplus.h>
CORBA::ULong  CORBA::StringDef::bound(
     CORBA::Environment& env ) ;
```

Description

This function returns the maximum number of characters in a StringDef object.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the maximum number of characters is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.13.2 Functions Usable when Inherited

(1) CORBA::StringDef::def_kind

Refer to 2.13.2 IRObject Common Interface for details about (1).

2.13.14 SequenceDef Interface

This section describes the SequenceDef interface.

2.13.14.1 CORBA::SequenceDef::bound()

Name

CORBA::SequenceDef::bound

Synopsis

```
#include <orb_cplus.h>
CORBA::ULong  CORBA::SequenceDef::bound(
     CORBA::Environment& env );
```

Description

This function returns the maximum number of sequence elements defined in a SequenceDef object.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the maximum number of sequence elements is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.14.2 CORBA::SequenceDef::element_type()

Name

CORBA::SequenceDef::element_type

Synopsis

```
#include <orb_cplus.h>
CORBA::TypeCode_ptr  CORBA::SequenceDef::element_type(
     CORBA::Environment& env) ;
```

Description

This function returns a type code indicating the type of sequence elements defined in a SequenceDef object.

Since this function acquires area to store the type code, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a type code indicating the type of sequence elements is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.14.3 Functions Usable when Inherited

(1) CORBA::SequenceDef::def_kind

Refer to 2.13.2 IRObject Common Interface for details about (1).

2.13.15 ArrayDef Interface

This section describes the ArrayDef interface.

2.13.15.1 CORBA::ArrayDef::length()

Name

CORBA::ArrayDef::length

Synopsis

```
#include <orb_cplus.h>
CORBA:Ulong  CORBA::ArrayDef::length(
     CORBA::Environment& env );
```

Description

This function returns the number of array elements defined in an ArrayDef object.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the number of array elements is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.15.2 CORBA::ArrayDef::element_type()

Name

CORBA::ArrayDef::element_type

Synopsis

```
#include <orb_cplus.h>
CORBA::TypeCode_ptr  CORBA::ArrayDef::element_type(
     CORBA::Environment& env ) ;
```

Description

This function returns a type code indicating the type of array elements defined in an ArrayDef object.

Since this function acquires area to store the type code, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a type code indicating the type of array elements is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.15.3 Functions Usable when Inherited

(1) CORBA::ArrayDef::def_kind

Refer to 2.13.2 IRObject Common Interface for details about (1).

2.13.16 WstringDef Interface

This section describes the WstringDef interface.

2.13.16.1 CORBA::WstringDef::bound()

Name

CORBA::WstringDef::bound

Synopsis

```
#include <orb_cplus.h>
CORBA::ULong  CORBA::WstringDef::bound(
CORBA::Environment& env );
```

Description

This function returns the maximum number of characters of the WstringDef object.

Parameters

env

A structure that may contain exception information.

Return Values

Normal termination, the maximum number of characters is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.16.2 Functions Usable when Inherited

(1) CORBA::WstringDef::def_kind

Refer to 2.13.2 IRObject Common Interface for details about (1).

2.13.17 InterfaceDef Interface

This section describes the InterfaceDef interface.

2.13.17.1 CORBA::InterfaceDef::describe_interface()

Name

CORBA::InterfaceDef::describe_interface

Synopsis

```
#include <orb_cplus.h>
CORBA::FullinterfaceDescription *CORBA::InterfaceDef::describe_interface(
     CORBA::Environment& env );
```

Description

This function returns the definition information of an InterfaceDef object, including inheritance information, from FullinterfaceDescription.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the definition information of an InterfaceDef (FullinterfaceDescription) is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.17.2 Functions Usable when Inherited

(1) CORBA::InterfaceDef::def_kind

(2) CORBA:InterfaceDef::id

(3) CORBA::InterfaceDef::name

(4) CORBA::InterfaceDef::defined_in

(5) CORBA::InterfaceDef::describe

(6) CORBA::InterfaceDef::lookup

(7) CORBA::InterfaceDef::contents

(8) CORBA::InterfaceDef::lookup_name

(9) CORBA::InterfaceDef::describe_contents

Refer to 2.13.2 IRObject Common Interface for details about (1).

Refer to 2.13.3 Contained Common Interface for details about (2) to (5).

Refer to 2.13.4 Container Common Interface for details about (6) to (9).

2.13.18 OperationDef Interface

This section describes the OperationDef interface.

2.13.18.1 CORBA::OperationDef::result()

Name

CORBA::OperationDef::result

Synopsis

```
#include <orb_cplus.h>
CORBA::TypeCode_ptr  CORBA::OperationDef::result(
     CORBA::Environment& env );
```

Description

This function returns a type code indicating the type of operation return values defined in an OperationDef object.

Since this function acquires area to store the type code, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a type code indicating the type of operation return values is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.18.2 CORBA::OperationDef::params()

Name

CORBA::OperationDef::params

Synopsis

```
#include <orb_cplus.h>
CORBA::ParDescriptionSeq  *CORBA::OperationDef::params(
CORBA::Environment& env );
```

Description

This function returns operation parameter information defined in an OperationDef object, in CORBA::ParDescriptionSeq list format (refer to 2.13.1 Type Definitions).

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the operation parameter information (CORBA::ParDescriptionSeq) is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.18.3 CORBA::OperationDef::contexts()

Name

CORBA::OperationDef::contexts

Synopsis

```
#include <orb_cplus.h>
CORBA::ContextIdSeq  *CORBA::OperationDef::contexts(
     CORBA::Environment& env );
```

Description

This function returns a list of the operation context IDs defined in an OperationDef object (refer to 2.13.1 Type Definitions).

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a context ID list (ContextIdSeq) is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.18.4 CORBA::OperationDef::exceptions()

Name

CORBA::OperationDef::exceptions

Synopsis

```
#include <orb_cplus.h>
CORBA::ExceptionDefSeq  *CORBA::OperationDef::exceptions(
     CORBA::Environment& env );
```

Description

This function returns a list of operation exception information defined in an OperationDef object (refer to 2.13.1 Type Definitions).

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a list of defined operation exception information is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.18.5 Functions Usable when Inherited

(1) CORBA::OperationDef::def_kind

(2) CORBA::OperationDef::id

(3) CORBA::OperationDef::name

(4) CORBA::OperationDef::defined_in

(5) CORBA::OperationDef::describe

Refer to 2.13.2 IRObject Common Interface for details about (1).

Refer to 2.13.3 Contained Common Interface for details about (2) to (5).

2.13.19 AttributeDef Interface

This section describes the AttributeDef interface.

2.13.19.1 CORBA::AttributeDef::type()

Name

CORBA::AttributeDef::type

Synopsis

```
#include <orb_cplus.h>
CORBA::TypeCode_ptr  CORBA::AttributeDef::type(
     CORBA::Environment& env ) ;
```

Description

This function returns the type code of an AttributeDef object.

Since this function acquires area to store the type code, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the type code of an AttributeDef object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.19.2 Functions Usable when Inherited

(1) CORBA::AttributeDef::def_kind

(2) CORBA::AttributeDef::id

(3) CORBA::AttributeDef::name

(4) CORBA::AttributeDef::defined_in

(5) CORBA::AttributeDef::describe

Refer to 2.13.2 IRObject Common Interface for details about (1).

Refer to 2.13.3 Contained Common Interface for details about (2) to (5).

2.13.20 ExceptionDef Interface

This section describes the ExceptionDef interface.

2.13.20.1 CORBA::ExceptionDef::type()

Name

CORBA::ExceptionDef::type

Synopsis

```
#include <orb_cplus.h>
CORBA::TypeCode_ptr  CORBA::ExceptionDef::type(
     CORBA::Environment& env );
```

Description

This function returns the type code of an ExceptionDef object.

Since this function acquires area to store the type code, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the type code of an ExceptionDef object is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.20.2 CORBA::ExceptionDef::members()

Name

CORBA::ExceptionDef::members

Synopsis

```
#include <orb_cplus.h>
CORBA::StructMemberSeq  *CORBA::ExceptionDef::members(
     CORBA::Environment& env );
```

Description

This function returns the definition information list (CORBA::StructMemberSeq) of exception events defined in an ExceptionDef object.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the definition information list (CORBA::StructMemberSeq) of exception events is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.13.20.3 Functions Usable when Inherited

(1) CORBA::ExceptionDef::def_kind

(2) CORBA::ExceptionDef::id

(3) CORBA::ExceptionDef::name

(4) CORBA::ExceptionDef::defined_in

(5) CORBA::ExceptionDef::describe

Refer to 2.13.2 IRObject Common Interface for details about (1).

Refer to 2.13.3 Contained Common Interface for details about (2) to (5).

2.14 Other Functions

This section describes the CORBA functions not included in the previous interfaces.

2.14.1 CORBA::string_alloc()

Name

CORBA::string_alloc

Synopsis

```
#include <orb_cplus.h>
static CORBA::Char *   CORBA::string_alloc(
     CORBA::ULong len );
```

Description

This function allocates a character string area equal to the number of characters specified at len +1, and returns a pointer to that string.

Parameters

len

The length of the area to be allocated.

Return Value

When this function terminates normally, it returns a pointer to the beginning of the allocated string area.

When this function does not terminate normally, it returns NULL.

2.14.2 CORBA::string_free()

Name

CORBA::string_free

Releases a character string area.

Synopsis

```
#include <orb_cplus.h>
void    CORBA::string_free(
     CORBA::Char *str );
```

Description

This function releases a character string area that has been specified with str.

Parameters

str

The string area to be freed.

Return Value

None.

2.14.3 CORBA::string_dup()

Name

CORBA::string_dup

Synopsis

```
#include <orb_cplus.h>
CORBA::Char *   CORBA::string_dup(
     const CORBA::Char *str );
```

Description

This function duplicates the string area specified at str, and returns a pointer to the duplicated string.

Parameters

str

The string area to be replicated.

Return Value

When this function terminates normally, it returns a pointer to the beginning of the replicated string area.

When this function does not terminate normally, it returns NULL.

2.14.4 CORBA::wstring_alloc()

Name

CORBA::wstring_alloc

Synopsis

```
#include <orb_cplus.h>
CORBA::WChar *  CORBA::wstring_alloc(
     CORBA::ULong len );
```

Description

This function allocates a character string area equal to the number of characters specified at len +1, and returns a pointer to that string.

Parameters

len

The length of the area to be allocated.

Return Value

When this function terminates normally, it returns a pointer to the beginning of the allocated string area.

When this function does not terminate normally, it returns NULL.

2.14.5 CORBA::wstring_free()

Name

CORBA::wstring_free

Synopsis

```
#include <orb_cplus.h>
void    CORBA::wstring_free(
     CORBA::WChar *str );
```

Description

This function releases the string area specified at str.

Parameters

str

The string area to be freed.

Return Value

None.

2.14.6 CORBA::wstring_dup()

Name

CORBA::wstring_dup

Synopsis

```
#include <orb_cplus.h>
CORBA::Wchar *  CORBA::wstring_dup(
     const CORBA::WChar *wstr );
```

Description

This function duplicates the string area specified at wstr, and returns a pointer to the duplicated string.

Parameters

wstr

The string area to be replicated.

Return Value

When this function normally terminates, it returns a pointer to the beginning of the replicated string area.

When this function abnormally terminates, it returns NULL.

2.14.7 CORBA::release()

Name

CORBA::release

Synopsis

```
#include <orb_cplus.h>
void    CORBA::release( CORBA::Object_ptr obj ) ;
void    CORBA::release( CORBA::Environment_ptr obj ) ;
void    CORBA::release( CORBA::NamedValue_ptr obj ) ;
void    CORBA::release( CORBA::NVList_ptr obj ) ;
void    CORBA::release( CORBA::Request_ptr obj ) ;
void    CORBA::release( CORBA::Context_ptr obj ) ;
void    CORBA::release( CORBA::Principal_ptr obj ) ;
void    CORBA::release( CORBA::TypeCode_ptr obj ) ;
void    CORBA::release( CORBA::BOA_ptr obj ) ;
void    CORBA::release( CORBA::ORB_ptr obj ) ;
void    CORBA::release( CORBA::ServerRequest_ptr obj ) ;
```

Description

This function releases a pointer area specified with obj.

Parameters

obj

The memory area to be freed.

Return Value

None.

2.14.8 CORBA::ORB::net_disconnect()

Name

CORBA::ORB::net_disconnect

Releases communication resource

Synopsis

```
void
CORBA::ORB::net_disconnect(
        CORBA::Object_ptr         obj,
        CORBA::Environment&       env )
```

Description

This function releases the communication resource used by the object specified in obj.

Parameters

obj

The object reference to be disconnected.

env

A structure that may contain exception information.

Return Values

For normal termination, CORBA_NO_EXCEPTION is returned.

For abnormal termination, a CORBA_SYSTEM_EXCEPTION is returned as follows:

IDL:CORBA/StExcep/INV_OBJREF:1.0

An invalid Object was specified.

IDL:CORBA/StExcep/COMM_FAILURE:1.0

Communication failure.

Notes

- Depending on the client application status, error messages may be issued when this function is invoked.
  - CORBA_COMM_FAILURE is issued if the application is using multithreading and a thread other than the one invoking this function is communicating with the server.
  - Asynchronous request results cannot be received when an asynchronous request has been issued.
  - If this function is issued for the same obj, CORBA_INV_OBJREF is reported.
  - If the connection information for the server machine does not exist, CORBA_INV_OBJREF or CORBA_COMM_FAILURE is posted.
- This operation also releases instance information held by other objects sharing the server which hosts the object to be released (obj), specified in the parameters of the invoking client process.

2.14.9 CORBA::ORB::set_client_timer()

Name

CORBA::ORB::set_client_timer

Sets server method wait time.

Synopsis

```
#include <orb_cplus.h>
void
CORBA::ORB::set_client_timer(
        CORBA::Object_ptr       obj,
        CORBA::Long             time,
        CORBA::Environment      &env );
```

Description

Sets the wait time until the server method for the server object is returned in client applications. The wait time specified is valid for all server objects operating on the host specified in obj.

Parameters

obj

The object reference to the server object whose response time is to be set.

time

Response time until the server method returns (seconds). The response time can be specified by a value from 0 to 100000000. If 0 is specified, the response time until the server method returns will not be monitored.

env

A structure that may contain exception information.

Return Values

For normal termination, there is no return value.

For abnormal termination, the following SystemExceptions occur:

IDL:CORBA/StExcep/INV_OBJREF:1.0

An invalid object was specified in obj.

IDL:CORBA/StExcep/BAD_PARAM:1.0

An invalid value was specified in time.

IDL:CORBA/StExcep/NO_MEMORY:1.0

There is insufficient memory.

For details minor codes that are set, refer to CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

Notes

- This method is used when the standby time of the client processing must be modified from the standby time set in the config file (period_receive_timeout x 5 seconds).
- Wait time is specified in process units by issuing this method after the server object object reference is fetched and before requests are sent to the server.
- To modify the wait time while a task is running, release the connection information for the relevant Object with CORBA::ORB::net_disconnect(), then issue this method.
- However, connection information must be released with CORBA::ORB::net_disconnect() before this method is issued, if requests are issued to the same server machine before server object object references are fetched or while they are being fetched. This would apply, for example, if CosNaming::NamingContext::resolve() was used when server object object references were fetched, and the Naming Service and server applications were running on the same machine.

2.14.10 CORBA::ORB::set_client_request_timer()

Name

CORBA::ORB::set_client_request_timer

Set the server method response time for the thread.

Synopsis

```
#include <orb_cplus.h>
void
CORBA::ORB::set_client_request_timer(
        CORBA::Long             time,
        CORBA::Environment      &env );
```

Description

This function sets the response time during which the client application will wait until the server method returns. This time is actually specified by the parameter time. The response time set here is effective on all the requests that are issued by the thread that invokes this function. Once this function is invoked, CORBA::ORB::clear_client_request_timer() must also be invoked before the thread terminates.

Parameters

time

Response time until the server method returns (seconds). The response time can be specified by a value from 0 to 100000000. If 0 is specified, the response time until the server method returns will not be monitored.

env

A structure that may contain exception information.

Return Values

For normal termination, there is no return value.

For abnormal termination, the following SystemExceptions occur:

IDL:CORBA/StExcep/BAD_PARAM:1.0

An invalid value was specified in time.

IDL:CORBA/StExcep/NO_MEMORY:1.0

There is insufficient memory.

For details of the minor codes that are set, refer to CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

Notes

- Use this method if you need to change the response time during which the client process waits.
- When invoked before requests are sent to the server, this method sets the response time for the thread.
- If this method is invoked, CORBA::ORB::clear_client_request_timer() must also be invoked before the thread terminates; otherwise, a memory leak will occur.
- To change the timeout value once this method is invoked, re-invoke CORBA::ORB::set_client_request_timer(). When doing so, you do not need to invoke CORBA::ORB::clear_client_request_timer() beforehand.
- The priorities of response time settings are indicated below, in descending order.
    1. Setting made by CORBA::ORB::set_client_request_timer().
    2. Setting made by CORBA::ORB::set_client_timer().
    3. Setting made by the period_receive_timeout parameter in the CORBA service environment setup file (config).

2.14.11 CORBA::ORB::get_client_request_timer()

Name

CORBA::ORB::get_client_request_timer

Acquire the response time of a server method.

Synopsis

```
#include <orb_cplus.h>
CORBA::Long
CORBA::ORB::get_client_request_timer(
        CORBA:: Object_ptr      obj,
        CORBA::Environment      &env );
```

Description

This function acquires the server method response time (in seconds).

The priorities of response time settings are indicated below, in descending order. If CORBA_OBJECT_NIL is assigned to obj, this method will acquire the setting 1 or 3.

1. Setting made by CORBA::ORB::set_client_request_timer()
2. Setting made by CORBA::ORB::set_client_timer()
3. Setting made by the period_receive_timeout parameter in the CORBA service environment setup file (config)

Use this function when you want to see the response time of a server method in a client application.

Parameters

obj

The object reference to the server object whose response time is to be acquired.

env

A structure that may contain exception information.

Return Values

The server method response time returns.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.14.12 CORBA::ORB::clear_client_request_timer()

Name

CORBA::ORB::clear_client_request_timer

Reset the server method response time for the thread.

Synopsis

```
#include <orb_cplus.h>
void
CORBA::ORB::clear_client_request_timer(
        CORBA::Environment      &env );
```

Description

This function resets the server method response time set by CORBA::ORB::set_client_request_timer(). After this function has been executed, the server method response time (during which requests issued by the thread that invoked this function will wait) returns to the system default value.

If the server method response time was set by CORBA::ORB::set_client_request_timer(), you need to issue this function before terminating the thread. This function has no effect if it is issued by a thread that has not set any server method response time by CORBA::ORB::set_client_request_timer().

Parameters

env

A structure that may contain exception information.

Return Values

When this function terminates normally, no value is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. For details on the exception information and minor codes that are set, refer to CORBA Service Exception Information, and CORBA Service Minor Codes, in Exception Information Minor Codes to be Reported from the CORBA Service.

2.14.13 CORBA::ORB::bind_object()

Name

CORBA::ORB::bind_object

Register the object and process bind relationship.

Synopsis

```
#include <orb_cplus.h>
void
CORBA::ORB::bind_object(
        CORBA::Object_ptr       obj,
        CORBA::Environment      &env );
```

Description

This function registers the bind relationship for the argument object (obj) and the process that issued this API in the CORBA Service. To unbind the relationship, issue CORBA::ORB::unbind_object().

Parameters

obj

The target object that registers the bind relationship.

env

The structure that contains the exception information.

Return Values

When this function returns normally, no value is returned.

If the function terminates abnormally, the SystemException object reference is set in the exception member of the env structure. The meaning of the exception information is shown below. For details on the meanings of the minor codes, refer to "Exception Information Minor Codes Reported from the CORBA Service" in "Messages".

IDL:CORBA/StExcep/BAD_OPERATION:1.0

The POA(Portable Object Adapter) library may be linked.

If POA is not used, remove the POA library from the link.

IDL:CORBA/StExcep/BAD_PARAM:1.0

The object that you tried to bind has already been bound.

Implementation of the object that you tried to bind is different to the implementation of the process.

IDL:CORBA/StExcep/NO_RESOURCE:1.0

The upper limit for binding was exceeded.

IDL:CORBA/StExcep/NO_IMPLEMENT:1.0

The client library (ODWINCPP.LIB) was used.

Notes

- The number of objects that can be registered in the bind relationship depends on max_bind_instances in the CORBA Service settings file (config).
- If this function is used, set "object" for the "iswitch" parameter when registering the implementation repository.
- If POA(Portable Object Adapter) is used, this function must not be issued.
- This function cannot be used in the client library (ODWINCPP.LIB).

2.14.14 CORBA::ORB::unbind_object()

Name

CORBA::ORB::unbind_object

Unbind the object and process bind relationship.

Synopsis

```
#include <orb_cplus.h>
void
CORBA::ORB::unbind_object(
        CORBA::Object_ptr       obj,
        CORBA::Environment      &env );
```

Description

This function unbinds the relationship for the argument object (obj) and process registered in CORBA::ORB::bind_object().

Parameters

obj

The target object that unbinds the relationship.

env

The structure that contains the exception information.

Return Values

When this function terminates normally, no value is returned.

If the function terminates abnormally, the SystemException object reference is set in the exception member of the env structure. The meaning of the exception information is shown below. For details on the meanings of the minor codes, refer to Exception Information Minor Codes Reported from the CORBA Service" in "Messages".

IDL:CORBA/StExcep/BAD_OPERATION:1.0

The POA(Portable Object Adapter) library may be linked.

If POA is not used, remove the POA library from the link.

IDL:CORBA/StExcep/BAD_PARAM:1.0

The object has not been registered in the CORBA Service bind relationship information.

IDL:CORBA/StExcep/NO_IMPLEMENT:1.0

The client library (ODWINCPP.LIB) was used.

Notes

- If POA(Portable Object Adapter) is used, this function must not be issued.
- This function cannot be used in the client library (ODWINCPP.LIB).

2.14.15 CORBA::ORB:: set_unbinded_object_rejecting()

Name

CORBA::ORB::set_unbinded_object_rejecting

Register the interface that notifies the exception for an object for which a bind relationship has not been registered.

Synopsis

```
#include <orb_cplus.h>
void
CORBA::ORB::set_unbinded_object_rejecting(
        CORBA::Char             *IntfId,
        CORBA::Environment      &env );
```

Description

This function registers the interface repository ID (IntfId). When a request is received for the object that holds the registered interface, it notifies the client to a system exception (INV_OBJREF) if the object bind relationship has not been registered in the CORBA Service.

If any of the following cases apply, it means that the bind relationship is not registered.

- CORBA::ORB::bind_object() was not issued
- The relationship was unbound after CORBA::ORB::unbind_object() was already issued
- A bind relationship for which the session time was detected was deleted.

Issue this function for each server process. To register more than one interface, issue this function for each interface registered.

Parameters

IntfId

The interface repository ID that notifies the exception if the bind relationship has not been registered.

env

The structure that contains the exception information.

Return Values

When the function terminates normally, no value is returned.

If the function terminates abnormally, the SystemException object reference is set in the exception member of the env structure. The meaning of the exception information is shown below. For details on the meanings of the minor codes, refer to "Exception Information Minor Codes Reported from the CORBA Service" in "Messages".

IDL:CORBA/StExcep/BAD_INV_ORDER:1.0

This function was issued after the activation function (CORBA::BOA::impl_is_ready) was issued.

IDL:CORBA/StExcep/BAD_PARAM:1.0

NULL was specified for the argument interface repository ID (IntfId).

IDL:CORBA/StExcep/NO_IMPLEMENT:1.0

The client library (ODWINCPP.LIB) was used.

IDL:CORBA/StExcep/NO_RESOURCE:1.0

The upper limit that can be registered was exceeded.

Notes

- This function must be issued before the activation function (CORBA::BOA::impl_is_ready).
- The number of interfaces that can be registered depends on the memory capacity that can be used in the process.
- If POA(Portable Object Adapter) is used, this function must not be issued.
- This function cannot be used in the client library (ODWINCPP.LIB).

2.14.16 CORBA::Object::check_ssn_timeout()

Name

CORBA::Object::check_ssn_timeout

Check whether a session timeout has occurred.

Synopsis

```
#include <orb_cplus.h>
void
CORBA::Object::check_ssn_timeout(
        CORBA::Environment      &env );
```

Description

This function determines whether a session timeout has occurred in the object.

Parameters

env

The structure that contains the exception information.

Return Values

When this function returns normally, it returns CORBA_TRUE when a session timeout has occurred. If there was no session timeout, it returns CORBA_FALSE.

If the function terminates abnormally, the SystemException object reference is set in the exception member of the env structure. The meaning of the exception information is shown below. For details on the meanings of the minor codes, refer to"Exception Information Minor Codes Reported from the CORBA Service" in "Messages".

IDL:CORBA/StExcep/NO_IMPLEMENT:1.0

The client library (ODWINCPP.LIB) was used.

Notes

- This function must be issued using the destructor extension.
- This function cannot be used in the client library (ODWINCPP.LIB).

2.14.17 CORBA::ORB::register_reply_interceptor()

Name

CORBA::ORB::register_reply_interceptor

Registers the exit function.

Synopsis

```
#include <orb_cplus.h>
void
CORBA::Object::check_ssn_timeout(
        void                    (*funcptr)(void),
        CORBA::Environment      &env );
```

Description

The function specified in the function pointer (funcptr) of the argument is registered as the exit function. The exit function that is registered is executed after the server application interface implementation function completes processing and sends a reply to the client.

The exit function is executed after interface implementation function processing is complete, even if no reply is sent to the client, for example when 'oneway' is specified in the interface implementation function.

Parameters

funcptr

This is the function pointer of the exit function.

env

The structure that contains the exception information.

Return Values

When this function returns normally, it returns CORBA_TRUE when a session timeout has occurred. If there was no session timeout, it returns CORBA_FALSE.

If the function terminates abnormally, the SystemException object reference is set in the exception member of the env structure. The meaning of the exception information is shown below. For details about the meanings of the minor codes, refer to"Exception Information Minor Codes Reported from the CORBA Service" in "Messages".

IDL:CORBA/StExcep/BAD_INV_ORDER:1.0

This function was issued after the activation function (CORBA::BOA::impl_is_ready) was issued.

IDL:CORBA/StExcep/BAD_PARAM:1.0

NULL was specified in the function pointer (funcptr) of the exit function.

IDL:CORBA/StExcep/NO_IMPLEMENT:1.0

The client library (ODWINCPP.LIB) was used.

IDL:CORBA/StExcep/NO_RESOURCE:1.0

The exit function has already been registered.

Notes

- The exit function format must be as follows:

```
void exit function name()
```

- If a member function is used as the exit function, it must be a static member function in 'public'.
- This function must be issued before the activation function (CORBA::BOA::impl_is_ready) is issued.

- If the exit function processing registered in this function is not completed within the monitoring time (set in 'reply_interceptor_timeout' of the implementation repository definition; the default is 300 seconds), the application is stopped by force. For this reason, it is recommended that the application is run on a WorkUnit that contains the automatic application restart functionality when using this function.
- This function cannot be used in the client library (ODWINCPP.LIB).

Example

```
class MyClass
{
public:
    static void exitfunc()
    {
        /* exit function processing */
    }
};

int
main( int argc, char *argv[] )
{
    CORBA::ORB_ptr  orb;
    CORBA::Environment_ptr  env = new CORBA::Environment;

    orb = CORBA::ORB_init( argc, argv, FJ_OM_ORBid, *env );

    orb->register_reply_interceptor( MyClass::exitfunc, *env );

    ...
}
```

2.14.18 FJ::ImplementationRep::lookup_id()

Name

FJ::ImplementationRep::lookup_id

Synopsis

```
#include <orb_cplus.h>
CORBA::Object_ptr  FJ::ImplementationRep::lookup_id(
        FJ::RepositoryId        searchid,
        CORBA::Environment& env );
```

Description

This function retrieves the object with the object ID specified in searchid and returns the object reference.

Parameters

searchid

The implementation repository ID.

env

The structure that contains the target exception information.

Return Values

For normal termination, returns the retrieved object reference.

For abnormal termination, sets the following user definition exception or system exception in env (thrown when try-catch is used):

FJ::NameDoesn'tExist

No object is registered with the repository ID information specified in the Implementation Repository.

Refer to the CORBA Service Exception Information for details of system exceptions.

2.15 Server Application Interface Windows32 Solaris32 Linux32

This is not valid for Linux (64 bit).

This section describes memory management application interfaces (API) required by server applications to transfer data to and from skeletons.

The following APIs are available:

String type memory Acquisition API (TD::string_alloc)

Memory release API (TD::free)

SMO name Acquisition API (TD::get_smo_name)

Client identifier Acquisition API (TD::get_client_id)

User identifier Acquisition API (TD::get_user_information)

Session ID notice API (TD::getsessionid)

Session continuation declaration API (TD::setcontcvt)

Session ID reference API (TD::refsessionid)

Each API is described below.

2.15.1 TD::string_alloc Windows32 Solaris32 Linux32

Name

TD::string_alloc

Synopsis

```
#include "td_cplus.h"
static void *TD::string_alloc( unsigned long length );
```

Description

This API corresponds to the CORBA::string_alloc function, which allocates a character string area equal to the number of characters specified at length +1, and returns a pointer to that string.

Return Values

For normal termination, a pointer to the acquired string is returned.

For abnormal termination, a NULL pointer is returned.

2.15.2 TD::wstring_alloc Windows32 Solaris32 Linux32

Name

TD::wstring_alloc

Synopsis

```
#include "td_cplus.h"
void *TD::wstring_alloc( unsigned long length );
```

Description

This is equivalent to CORBA::wstring_alloc. It acquires a dynamic wstring area of the number of characters specified in length.

Return Values

In the case of normal termination, a pointer to the acquired data is returned.
In the case of abnormal termination, a NULL pointer is returned.

2.15.3 TD::free Windows32 Solaris32 Linux32

Name

TD::free

Synopsis

```
#include "td_cplus.h"
static void TD::free( void* );
```

Description

This API corresponds to the CORBA::string_free function, which releases the string area specified by the pointer.

Return Values

None.

2.15.4 TD::wstring_free Windows32 Solaris32 Linux32

Name

TD::wstring_free

Synopsis

```
#include "td_cplus.h"
void TD::wstring_free( void* );
```

Description

This is equivalent to CORBA::wstring_free. It releases the specified pointer area.

Return Values

None.

2.15.5 TD::get_smo_name Windows32 Solaris32 Linux32

Name

TD::get_smo_name

Synopsis

```
#include "td_cplus.h"
static long TD::get_smo_name ( char *smoname );
```

Description

This API stores the SMO name in the area specified in smoname. Allocate 256 bytes of memory for the SMO name.

Return Values

For normal termination, 0 is returned.

For abnormal termination, one of the following is returned:

1: Parameter error

2: Protocol error

3: Sequence error

99: System error

2.15.6 TD::get_client_id Windows32 Solaris32 Linux32

Name

TD::get_client_id

Synopsis

```
#include "td_cplus.h"
#include "ISTD_smocpp.h"
static long TD::get_client_id( char *bufaddr, long buflength, long length );
```

Description

This API sets a client ID in the address specified by bufaddr, and the length of a sequence data area using the length specified. The length of bufaddr (> 48 bytes) is specified in buflength.

Return Values

For normal termination, NULL is returned.

For abnormal termination, one of the following is returned:

1: Parameter error

2: Protocol error

3: Sequence error

99: System error

2.15.7 TD::get_user_information Windows32 Solaris32 Linux32

Name

TD::get_user_information

Synopsis

```
#include "td_cplus.h"
static long TD::get_user_information ( TD_USERINFO *addr );
```

Description

This API sets the length of the User Identification Information to data_length of the TD_USERINFO area specified by addr, and the User Identifier to the data.

/* TD_USER_INFORMATION area */

typedef struct TD_user_information_t{

long data_length;

char data[TD_USERID_LENGTH];

} TD_USERINFO;

Return Values

For normal termination, NULL is returned.

For abnormal termination, one of the following is returned:

1: Parameter error

2: Protocol error

3: Sequence error

99: System error

2.15.8 TD::getsessionid Windows32 Solaris32 Linux32

Name

TD::getsessionid

Synopsis

```
#include  "td_cplus.h"
long  TD::getsessionid(TD_SESSION_ID *sessionid_p)
```

Description

This API acquires and transmits the keys (session ID) required to implement the following processes:

- Process Binding Function
- Session information management
- Windows32 Solaris32
  AIM linkage session inheritance

In sessionid_p, set the pointer to the session ID structure.

Return Values

If the operation terminates normally, the return value is 0 and the session ID is stored in the area specified by the parameter. In the event of abnormal termination, one of the following values is returned:

1 : Parameter error

2 : Protocol error (double execution of TD::getsessionid)

99 : System error

2.15.9 TD::setcontcvt Windows32 Solaris32 Linux32

Name

TD::setcontcvt()

Synopsis

```
#include  "td_cplus.h"
long  TD::setcontcvt()
```

Description

This API declares a session continuation to the system.

Return Values

For normal termination, the return value is 0. In the event of abnormal termination, one of the following values is returned.

1 : Parameter error

2 : Protocol error (Invoked by non-resident application, meaning that either process binding definition was not made, or session ID was not issued.)

99 : System error

2.15.10 TD::refsessionid Windows32 Solaris32 Linux32

Name

TD::refsessionid

Synopsis

```
#include  "td_cplus.h"
long  TD::refsessionid(TD_SESSION_ID *sessionid_p)
```

Description

This operation is used when the user application refers to a session ID which has already been acquired.

The session information storage area must be acquired before this API is issued.

In sessionid_p, specify the area address for locating the session continuation information.

Return Values

If the operation terminates normally, the return value is 0 and the session ID is stored in the area specified by sessionid_p. In the event of abnormal termination, one of the following values is returned:

1 : Parameter error

99 : System error

2.16 Current Class Windows32 Solaris32 Linux32

This section describes the current class.

2.16.1 CosTransactions::Current::begin Windows32 Solaris32 Linux32

Name

CosTransactions::Current::begin

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
void  CosTransactions::Current::begin(
CORBA::Environment& env)throw(CORBA::Exception);
```

Description

This function creates a new transaction.

Use the CORBA::ORB::resolve_initial_references method (CORBA::ORB::ObjectID_TransactionCurrent) to obtain the object reference of the Current interface.

If timeout is not set using the CosTransactions::Current::set_timeout method, the timeout period specified for TRAN_TIME_OUT in the operating environment file is assumed.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the SystemException or UserException object reference is set in the exception member of the env structure.

For UserException, the following exception class member is returned:

CosTransactions::SubtransactionsUnavailable:

An attempt was made to create a nested transaction.

For SystemException, one of the following exception class members is returned:

CORBA::StExcep::NO_IMPLEMENT:

The Database Linkage Service system is not started.

CORBA::StExcep::COMM_FAILURE:

The possible causes are:

- A communication error has occurred.
- Host information relating to the Database Linkage Service system is not defined in the OD environment file. The attribute initial_hosts (inithost on the PC) has not been defined correctly.

CORBA::StExcep::NO_RESOURCES:

A resource became insufficient. The maximum number of transactions may have been exceeded.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

2.16.2 CosTransactions::Current::commit Windows32 Solaris32 Linux32

Name

CosTransactions::Current::commit

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
void  CosTransactions::Current::commit(
CORBA::boolean          report_heuristics,
CORBA::Environment& env)throw(CORBA::Exception);
```

Description

This function commits a transaction.

Use the CORBA::ORB::resolve_initial_references method (CORBA::ORB::ObjectID_TransactionCurrent) to obtain the object reference of the Current interface. If CORBA_TRUE is specified for report_heuristics, a heuristic error is reported.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the SystemException or UserException object reference is set in the exception member of the env structure.

For UserException, one of the following exception class members is returned:

CosTransactions::NoTransaction:

No transaction has been created.

CosTransactions::HeuristicMixed:

Some transactions are in the commit completion state, and others are in the rollback completion state.

CosTransactions::HeuristicHazard:

The transaction completion status is unknown.

For SystemException, one of the following exception class members is returned:

CORBA::StExcep::NO_PERMISSION:

Permission to commit transactions is denied.

CORBA::StExcep::TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

CORBA::StExcep::NO_IMPLEMENT:

The Database Linkage Service system or Resource Manager is not started.

CORBA::StExcep::COMM_FAILURE:

A communication error has occurred.

CORBA::StExcep::NO_RESOURCES:

A resource became insufficient. The maximum number of transactions may have been exceeded.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

2.16.3 CosTransactions::Current::rollback Windows32 Solaris32 Linux32

Name

CosTransactions::Current::rollback

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
void  CosTransactions::Current::rollback(
     CORBA::Environment& env)throw(CORBA::Exception);
```

Description

This function rolls back a transaction.

Use the CORBA::ORB::resolve_initial_references method (CORBA::ORB::ObjectID_TransactionCurrent) to obtain the object reference of the Current interface.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the SystemException or UserException object reference is set in the exception member of the env structure.

For UserException, the following exception class member is returned:

CosTransactions::NoTransaction:

No transaction has been created.

For a SystemException, the following exception class members can be thrown:

CORBA::StExcep::NO_PERMISSION:

Permission to roll back transactions is denied.

CORBA::StExcep::TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

CORBA::StExcep::NO_IMPLEMENT:

The Database Linkage Service system or Resource Manager is not started.

CORBA::StExcep::COMM_FAILURE:

A communication error has occurred.

CORBA::StExcep::NO_RESOURCES:

A resource became insufficient. The maximum number of transactions may have been exceeded.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

2.16.4 CosTransactions::Current::rollback_only Windows32 Solaris32 Linux32

Name

CosTransactions::Current::rollback_only

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
void  CosTransactions::Current::rollback_only(
     CORBA::Environment& env)throw(CORBA::Exception);
```

Description

This function enables the transaction rollback operation only.

Use the CORBA::ORB::resolve_initial_references method (CORBA::ORB::ObjectID_TransactionCurrent) to obtain the object reference of the Current interface.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the SystemException or UserException object reference is set in the exception member of the env structure.

For UserException, the following exception class member is returned:

CosTransactions::NoTransaction:

No transaction has been created.

For SystemException, one of the following exception class members is returned:

CORBA::StExcep::TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

CORBA::StExcep::NO_IMPLEMENT:

The Database Linkage Service system is not started.

CORBA::StExcep::COMM_FAILURE:

A communication error has occurred.

CORBA::StExcep::NO_RESOURCES:

A resource became insufficient. The maximum number of transactions may have been exceeded.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

2.16.5 CosTransactions::Current::get_status Windows32 Solaris32 Linux32

Name

CosTransactions::Current::get_status

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
CosTransactions::Status  CosTransactions::Current::get_status(
     CORBA::Environment&      env)throw(CORBA::Exception);
```

Description

This function returns the status of a transaction.

Use the CORBA::ORB::resolve_initial_references method (CORBA::ORB::ObjectID_TransactionCurrent) to obtain the object reference of the Current interface.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure, and one of the following values is returned:

CosTransactions::StatusActive:

The transaction is being executed.

CosTransactions::StatusMarkedRollback:

The transaction is marked rollback.

CosTransactions::StatusPrepared:

The transaction is prepared.

CosTransactions::StatusCommited:

The transaction has been committed.

CosTransactions::StatusRolledBack:

The transaction has been rolled back.

CosTransactions::StatusUnknown:

The transaction status is unknown.

CosTransactions::StatusNoTransaction:

No transaction is being executed.

CosTransactions::StatusPreparing:

Prepare processing is being executed.

CosTransactions::StatusCommitting:

Commit processing is being executed.

CosTransactions::StatusRollingBack:

Rollback processing is being executed.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. One of the following exception class members is returned:

CORBA::StExcep::TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

CORBA::StExcep::NO_IMPLEMENT:

The Database Linkage Service system is not started.

CORBA::StExcep::COMM_FAILURE:

A communication error has occurred.

CORBA::StExcep::NO_RESOURCES:

A resource became insufficient. The maximum number of transactions may have been exceeded.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

2.16.6 CosTransactions::Current::get_transaction_name Windows32 Solaris32 Linux32

Name

CosTransactions::Current::get_transaction_name

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
CORBA::string CosTransactions::Current::get_transaction_name(
     CORBA::Environment&      env)throw(CORBA::Exception);
```

Description

This function returns a character string for identifying a transaction.

Use the CORBA::ORB::resolve_initial_references method (CORBA::ORB::ObjectID_TransactionCurrent) to obtain the object reference of the Current interface.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure, and the character string for identifying the transaction is returned. If no transaction has been created, a NULL character is returned.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure. One of the following exception class members is returned:

CORBA::StExcep::TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

CORBA::StExcep::NO_IMPLEMENT:

The Database Linkage Service system is not started.

CORBA::StExcep::COMM_FAILURE:

A communication error has occurred.

CORBA::StExcep::NO_RESOURCES:

A resource became insufficient. The maximum number of transactions may have been exceeded.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

2.16.7 CosTransactions::Current::set_timeout Windows32 Solaris32 Linux32

Name

CosTransactions::Current::set_timeout

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
void  CosTransactions::Current::set_timeout(
CORBA::ULong          seconds,
CORBA::Environment&          env)throw(CORBA::Exception);
```

Description

This function sets the value specified for seconds as the transaction timeout observation time.

If processing is not completed within the time set for seconds, the transaction is rolled back. If 0 is specified for seconds, no timeout is observed.

If CosTransactions::Current::set_timeout is called while a transaction is being processed, the specified timeout value is not valid until the next transaction starts.

Use the CORBA::ORB::resolve_initial_references method (CORBA::ORB::ObjectID_TransactionCurrent) to obtain the object reference of the Current interface.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the SystemException object reference is set in the exception member of the env structure.

If a system exception occurs, the following exception classes are posted:

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

CORBA::StExcep::INTERNAL:

CORBA Service error.

2.16.8 CosTransactions::Current::get_control 

Name

CosTransactions::Current::get_control

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
CosTransactions::Control  CosTransactions::Current::get_control(
                          CORBA::Environment&      env);
```

Description

This function returns the Control object that manages transaction text.

This Control object can be used to complete transactions in server applications.

Specify in current the Current interface object reference fetched by the CORBA::ORB::resolve_initial_references method (CORBA::ORB::ObjectId_TransactionCurrent).

Return Values

For normal termination, a NULL object reference is set in the exception member in env.

For abnormal termination, a SystemException object reference is set in the exception member in env.

When a system exception occurs, the following information is set:

CORBA::StExcep_NO_IMPLEMENT:

The Database Linkage Service system did not start.

CORBA::StExcep_COMM_FAILURE:

A communications failure occurred.

CORBA::StExcep_NO_RESOURCES:

There were insufficient resources.

CORBA::StExcep_NO_MEMORY:

Failed to secure dynamic memory.

2.16.9 CosTransactions::Current::suspend Windows32 Solaris32 Linux32

Name

CosTransactions::Current::suspend

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
CosTransactions::Control  CosTransactions::Current::suspend(
                          CORBA::Environment&      env);
```

Description

This function removes the association between a transaction generated by an application and a thread, and stops the transaction. The function returns the Control object that manages the transaction context that manages the transaction.

When this function terminates normally, the transaction remains valid until the resume function is executed.

Specify in current the Current interface object reference fetched by the CORBA::ORB::resolve_initial_references method (CORBA::ORB::ObjectId_TransactionCurrent).

Return Values

For normal termination, a NULL object reference is set in the exception member in env.

For abnormal termination, a SystemException object reference is set in the exception member in env.

When a system exception occurs, the following detailed information is set:

ex_CORBA_StExcep_NO_MEMORY:

Failed to secure dynamic memory.

ex_CORBA_StExcep_INTERNAL:

A CORBA initialization error occurred.

2.16.10 CosTransactions::Current::resume Windows32 Solaris32 Linux32

Name

CosTransactions::Current::resume

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
```

```
void  CosTransactions::Current::resume(
                          CosTransactions::Control   which,
                        CORBA::Environment&      env);
```

Description

This function associates a transaction and a current thread.

Specify in which the Control object object reference. The Control object is fetched by the CosTransactions::Current::get_control method. Alternatively, use the Control object returned by the suspend function.

The transaction is associated with the current thread in the Control object. If a null object is specified in which, the association between the transaction and the current thread is removed.

Return Values

For normal termination, a NULL object reference is set in the exception member in env.

For abnormal termination, a SystemException object reference is set in the exception member in env.

When a user exception occurs, the following detailed information is set:

CosTransactions::InvalidControl

An object other than a Control object was specified.

When a system exception occurs, the following detailed information is set:

CORBA::StExcep_INTERNAL:

CORBA Service error.

2.17 Transaction Initialization Class Windows32 Solaris32 Linux32

This is not valid for Linux (64 bit).

This section describes the transaction initialization class.

2.17.1 OTS::init Windows32 Solaris32 Linux32

Name

OTS::init

Synopsis

```
#include  "OTS_cplus.h"
void    OTS::init(
     CORBA::ORB_ptr           orb,
     const CORBA::char                    *impl,
     CORBA::Environment& env)throw(CORBA::Exception);
```

Description

This function connects a server application to the Database Linkage Service. Specify the Implementation Repository ID of a server application for impl.

Use the CORBA::ORB::resolve_initial_references method (CORBA_ORB_ObjectID_TransactionServerInit) to obtain the object reference of the Database Linkage Service interface.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the SystemException or UserException object reference is set in the exception member of the env structure.

For UserException, the following exception class members are returned:

OTS::Bad_Description:

A description in the resource definition file is invalid.

OTS::No_DefFile:

The resource definition file cannot be found, or the resource definition file name is not specified as Implementation Repository.

OTS::Rm_Error:

A temporary error occurred during an attempt to open a database.

OTS::Permission_Denied:

Permission to use the resource definition file is denied.

OTS::IOError:

An I/O error occurred during an attempt to read the resource definition file.

Note

Do not call this function more than once in one process.

2.18 Transaction Termination Class Windows32 Solaris32 Linux32

This is not valid for Linux (64 bit).

This section describes the transaction termination class.

2.18.1 OTS::term Windows32 Solaris32 Linux32

Name

OTS::term

Synopsis

```
#include  "OTS_cplus.h"
void  OTS::term(
     CORBA::Environment& env)throw(CORBA::Exception);
```

Description

This function disconnects a server application from the Database Linkage Service.

Use the CORBA::ORB::resolve_initial_reference method (CORBA::ORB::ObjectID_TransactionServerInit) to obtain the object reference of the Current interface.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the SystemException or UserException object reference is set in the exception member of the env structure.

Note

Do not call this function more than once in one process.

2.19 Load Balance Function Class Windows32 Solaris32 Linux32

This is not valid for Linux (64 bit).

This section explains the functions provided by the load balance function.

Note

The load balance function can only be used in the Interstage Application Server Enterprise Edition.

2.19.1 Load Balance Option Interface Windows32 Solaris32 Linux32

This section describes the load balance option interface.

2.19.1.1 ISOD::LBO::create_LBG() Windows32 Solaris32 Linux32

Name

ISOD::LBO::create_LBG

Synopsis

```
#include <orb_cplus.h>
#include <OM_LBOcpp.h>
ISOD::LBG_ptr  ISOD::LBO::create_LBG(
     CosNaming::NamingContext_ptr  nc,
     CosNaming::Name  &n,
     ISOD::LBO::LoadBalanceType  loadbalancetype,
     CORBA::Object_ptr  defaultobjectref,
CORBA::Environment  &env);
```

Description

This function creates a load balance object group, and registers it under the name specified in n, in the naming context specified in nc.

If successful, the object reference of the load balance object group is returned.

Since this function acquires area to store the object reference, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

nc

The naming context object.

n

The created load balance object group name.

loadbalancetype

Specify roundrobin

defaultobjref

If the load balance function is not operating, specify the default object reference to be returned by the Naming Service.

If it is not specified, the object reference is not returned from the Naming Service because the load balance function is not operating.

The object reference cannot be registered in, or deleted from the ISOD::LBG bind() and ISOD::LBG unbind() functions. If it is specified, it can be changed by the ISOD::LBG rebind_default() function after creating the load balance object group.

Use OMG IDL format (format starting with "IDL:") for the Interface Repository ID of the object reference to be specified in defaultobjectref.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the UserException object reference is set in the exception member of the env structure.

Detailed information is set in UserException_id. The values of _id are as follows:

ISOD::LBO::NotFound

The naming context specified in n could not be found.

ISOD::LBO::CannotProceed

The naming context specified in nc does not exist.

ISOD::LBO::InvalidName

Invalid name specification.

ISOD::LBO::AlreadyExist

The specified binding name already exists.

ISOD::LBO::InvalidType

Invalid specification of load balance type.

ISOD::LBO::BadObject

The object specified in defaultobjectref is invalid

ISOD::LBO::OperationBusy

A request for multi-processing has reached the maximum limit. Retry later.

2.19.1.2 ISOD::LBO::resolve_LBG() Windows32 Solaris32 Linux32

Name

ISOD::LBO::resolve_LBG

Synopsis

```
#include <orb_cplus.h>
#include <OM_LBOcpp.h>
ISOD::LBG_ptr  ISOD::LBO::resolve_LBG(
     CosNaming::NamingContext_ptr  nc,
     CosNaming::_Name  &n,
CORBA::Environment  &env);
```

Description

This function searches the load balance object group specified at n in the naming context specified as nc, and returns an object reference of the load balance object group.

Since this function acquires area to store the object reference, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

nc

The naming context object.

n

The searched load balance object group name.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the UserException object reference is set in the exception member of the env structure. Detailed information is set in UserException_id.

The values of _id are as follows:

ISOD::LBO::NotFound

The name specified in n could not be found.

ISOD::LBO::CannotProceed

The naming context specified in nc does not exist.

ISOD::LBO::InvalidName

Invalid name specification.

ISOD::LBO::OperationBusy

A request for multi-processing has reached the maximum limit. Retry later.

2.19.1.3 ISOD::LBO::delete_LBG() Windows32 Solaris32 Linux32

Name

ISOD::LBO::delete_LBG

Synopsis

```
#include <orb_cplus.h>
#include <OM_LBOcpp.h>
CORBA::Object_ptr  ISOD::LBO::delete_LBG(
     CosNaming::NamingContext_ptr  nc,
     CosNaming::Name  &n,
CORBA::Environment  &env);
```

Description

This function deletes the load balance object group from the Naming Service.

The load balance object group specified at n in the naming context specified at nc is deleted.

The object reference of the default object (when it is set), is returned. If an object (other than the default object) is registered in the load balance object group to be deleted, an exception is returned.

Since this function acquires area to store the object reference, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

nc

The naming context object.

n

The deleted load balance object group name.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the UserException object reference is set in the exception member of the env structure. Detailed information is set in UserException_id.

The values of _id are as follows:

ISOD::LBO::NotFound

The name specified in n could not be found.

ISOD::LBO::CannotProceed

The naming context specified in nc does not exist.

ISOD::LBO::InvalidName

Invalid name specification.

ISOD::LBO::NotEmpty

More than one object is registered in the object group.

ISOD::LBO::OperationBusy

A request for multi-processing has reached the maximum limit. Retry later.

2.19.1.4 ISOD::LBO::list_LBG() Windows32 Solaris32 Linux32

Name

ISOD::LBO::list_LBG

Synopsis

```
#include <orb_cplus.h>
#include <OM_LBOcpp.h>
void  ISOD::LBO::list_LBG(
     CosNaming::NamingContext_ptr  nc,
     ISOD::LBO::LBGList  *&LBGlist,
CORBA::Environment  &env);
```

Description

This operation returns a list of the load balance object group specified at n in the naming context specified at nc.

The load balance object group name, and the structure list (BindingObjectGroup) containing information for the load balance type is returned.

The maximum size of the load balance object group list that can be retrieved is 128 elements.

Parameters

nc

The naming context object.

LBGlist

The set area of the list of the load balance object groups.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, the UserException object reference is set in the exception member of the env structure.

2.19.1.5 ISOD::LBO::notify_down() Windows32 Solaris32 Linux32

Name

ISOD::LBO::notify_down

Synopsis

```
#include <orb_cplus.h>
#include <OM_LBOcpp.h>
void  ISOD::LBO::notify_down(
     CORBA::string HostName,
CORBA::Environment  &env);
```

Description

This function notifies the load balance function that the server is down.

When this function is called, it stops the return of object references from the server address specified at HostName.

Parameters

HostName

The host name or IP address of the downed server.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, a UserException object reference is set in the exception member of the env structure, and detailed information is set in UserException_id.

The values of _id are as follows:

ISOD::LBO::InvalidArgument

The HostName specification is invalid.

ISOD::LBO::CannotProceed2

An abnormality occurred in the load balance function DB process.

2.19.1.6 ISOD::LBO::notify_recover() Windows32 Solaris32 Linux32

Name

ISOD::LBO::notify_recover

Synopsis

```
#include <orb_cplus.h>
#include <OM_LBOcpp.h>
void  ISOD::LBO::notify_recover(
     CORBA::string  HostName,
CORBA::Environment  &env);
```

Description

This function notifies the load balance function that the server has been restored.

When this function is called, it starts the return of object references from the server address specified at HostName.

Parameters

HostName

The host name or IP address of the restored server.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, a UserException object reference is set in the exception member of the env structure, and detailed information is set in UserException_id.

The values of _id are as follows:

ISOD::LBO::InvalidArgument

The HostName specification is invalid.

ISOD::LBO::CannotProceed2

An abnormality occurred in the load balance function DB process.

2.19.2 Load Balance Object Group Interface Windows32 Solaris32 Linux32

This section describes the load balance object group interface.

2.19.2.1 ISOD::LBG::bind() Windows32 Solaris32 Linux32

Name

ISOD::LBG::bind

Synopsis

```
#include <orb_cplus.h>
#include <OM_LBOcpp.h>
void  ISOD::LBG::bind(
     CORBA::Object_ptr  objectref,
CORBA::Environment  &env);
```

Description

This function registers the object to be load balanced (specified at objectref) in this load balance object group .

Parameters

objectref

The object reference registered to the load balance object group.

Use OMG IDL format (format starting with "IDL:") for the Interface Repository ID of the object reference to be specified in objectref.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, a UserException object reference is set in the exception member of the env structure, and detailed information is set in UserException_id.

The values of _id are as follows:

ISOD::LBG::AlreadyBound

An object containing the same information has already been registered.

ISOD::LBG::CannotProceed2

An abnormality occurred in the load balance function DB process.

ISOD::LBG::BadObject

The object specified in objectref is invalid.

2.19.2.2 ISOD::LBG::unbind() Windows32 Solaris32 Linux32

Name

ISOD::LBG::unbind

Synopsis

```
#include <orb_cplus.h>
#include <OM_LBOcpp.h>
void ISOD::LBG::unbind(
     CORBA::Object_ptr  objectref,
CORBA::Environment  &env);
```

Description

This function deletes the object to be load balanced (specified at objectref) from this load balance object group.

Parameters

objectref

The object reference deleted from the load balance object group.

Use OMG IDL format (format starting with "IDL:") for the Interface Repository ID of the object reference to be specified in objectref.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, a UserException object reference is set in the exception member of the env structure, and detailed information is set in UserException_id.

The values of _id are as follows:

ISOD::LBG::NotFound

The object specified at objectref could not be found.

ISOD::LBG::CannotProceed2

An abnormality occurred in the load balance function DB process.

ISOD::LBG::BadObject

The object specified in objectref is invalid.

2.19.2.3 ISOD::LBG::rebind_default() Windows32 Solaris32 Linux32

Name

ISOD::LBG::rebind_default

Synopsis

```
#include <orb_cplus.h>
#include <OM_LBOcpp.h>
```

```
CORBA::Object_ptr  ISOD::LBG::rebind_default(
     CORBA::Object_ptr  objectref,
CORBA::Environment  &env);
```

Description

This function changes the default object of this load balance object group to the object specified at objectref.

Since this function acquires area to store the object reference, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

objectref

The object reference registered as default object.

Use OMG IDL format (format starting with "IDL:") for the Interface Repository ID of the object reference to be specified in objectref.

env

A structure that may contain exception information.

Return Values

For normal termination, the object reference of the default object (prior to the change) is returned, and.a NULL object reference is set in the exception member of the env structure.

For abnormal termination, a UserException object reference is set in the exception member of the env structure, and detailed information is set in UserException_id.

The values of _id are as follows:

ISOD::LBG::CannotProceed

The default object is not set.

ISOD::LBG::CannotProceed2

An abnormality occurred in the load balance function DB process.

ISOD::LBG::BadObject

The object specified in objectref is invalid.

ISOD::LBG::OperationBusy

A request for multi-processing has reached the maximum limit. Retry later.

2.19.2.4 ISOD::LBG::list() Windows32 Solaris32 Linux32

Name

ISOD::LBG::list

Synopsis

```
#include <orb_cplus.h>
#include <OM_LBOcpp.h>
void  ISOD::LBG::list(
     ISOD::LBG::ObjectList  *&objectref,
CORBA::Environment   &env);
```

Description

This function returns a list of object references of objects already registered in this load balance object group.

Parameters

objectref

The event data transmitted to the consumer

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member of the env structure.

For abnormal termination, a UserException object reference is set in the exception member of the env structure, and detailed information is set in UserException_id.

The value of _id is as follows:

ISOD::LBG::CannotProceed2

An abnormality occurred in the load balance function DB process.

2.20 Event Service Interface

This interface enables communication with the event service and notification service event channels.

Include "EventService_cplus_h" when using the event service event channel, and ""NotificationService_cplus_h" when using the notification service event channel.

2.20.1 CosEventComm Class

This section describes the CosEventComm class.

2.20.1.1 CosEventComm::PushConsumer::push()

Name

CosEventComm::PushConsumer::push

Synopsis

```
#include  <EventService_cplus.h>
void  CosEventComm::PushConsumer::push(
     const CORBA::Any&       data,
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Transmits the event data specified by data to the consumer.

Parameters

data

The event data transmitted to the consumer

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. In the event of abnormal termination, the SystemException object reference or UserException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exception may be generated.

CosEventComm::Disconnected

Not connected to event channel.

If automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

2.20.1.2 CosEventComm::PushConsumer::disconnect_push_consumer()

Name

CosEventComm::PushConsumer::disconnect_push_consumer

Synopsis

```
#include  <EventService_cplus.h>
void  CosEventComm::PushConsumer::disconnect_push_consumer(
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Declares termination of event communication by supplier.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.20.1.3 CosEventComm::PushSupplier::disconnect_push_supplier()

Name

CosEventComm::PushSupplier::disconnect_push_supplier

Synopsis

```
#include  <EventService_cplus.h>
void  CosEventComm::PushSupplier::disconnect_push_supplier(
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Declares termination of event communication by consumer.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.20.1.4 CosEventComm::PullSupplier::pull()

Name

CosEventComm::PullSupplier::pull

Synopsis

```
#include  <EventService_cplus.h>
CORBA::Any  *CosEventComm::PullSupplier::pull(
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Requests event data from supplier. This operation is blocked until event data can be fetched, or until an exception is generated. If the event data cannot be fetched and an immediate response is required, then use CosEventComm::PullSupplier::try_pull.

When the area for the any type event data is no longer required, use delete to release the area.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env, and event data from the supplier is issued. In the event of abnormal termination, the SystemException object reference or UserException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exception may be generated.

CosEventComm::Disconnected

Not connected to event channel.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

2.20.1.5 CosEventComm::PullSupplier::try_pull()

Name

CosEventComm::PullSupplier::try_pull

Synopsis

```
#include  <EventService_cplus.h>
CORBA::Any  *CosEventComm::PullSupplier::try_pull(
     CORBA::Boolean&         has_event,
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Requests event data from supplier. Instantly returned if event data cannot be fetched from supplier. If you want to block this operation until event data can be fetched, then please use CosEventComm::PullSupplier::pull.

When the area for the any type event data is no longer required, use delete to release the area.

Parameters

has_event (out parameter)

When event data has been fetched, CORBA_TRUE is set.

When event data has not been fetched, then CORBA_FALSE is set

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env and the event data from the supplier is posted. When event data has been fetched, CORBA::TRUE is set in has_event. If event data has not been fetched, then CORBA::FALSE is set in has_event. In this case, use delete to release the area when the area for the any type event data is no longer required.

For abnormal termination, the SystemException object reference or UserException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exception may be generated.

CosEventComm::Disconnected

Not connected to event channel.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

Notes

The cancellation need not be notified using ES::ChannelUtil::local_rollback() when CORBA_FALSE is set to has_event when a local transaction is operated.

Notifies the channel of completing using ES::ChannelUtil::local_commit().

Windows32 Solaris32 Linux32

The cancellation need not be notified using CosTransactions::Current::rollback() when CORBA_FALSE is set to has_event when a globaltransaction is operated.

Notifies the channel of completing using CosTransactions::Current::commit().

2.20.1.6 CosEventComm::PullSupplier::disconnect_pull_supplier()

Name

CosEventComm::PullSupplier::disconnect_pull_supplier

Synopsis

```
#include  <EventService_cplus.h>
void  CosEventComm::PullSupplier::disconnect_pull_supplier(
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Declares termination of event communication by consumer..

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.20.1.7 CosEventComm::PullConsumer::disconnect_pull_consumer()

Name

CosEventComm::PullConsumer::disconnect_pull_consumer

Synopsis

```
#include  <EventService_cplus.h>
void  CosEventComm::PullConsumer::disconnect_pull_consumer(
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Declares termination of event communication by consumer..

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.20.2 CosEventChannelAdmin Class

This section describes the CosEventChannelAdmin class.

2.20.2.1 CosEventChannelAdmin::EventChannel::for_consumers()

Name

CosEventChannelAdmin::EventChannel::for_consumers

Synopsis

```
#include  <EventService_cplus.h>
CosEventChannelAdmin::ConsumerAdmin_ptr
CosEventChannelAdmin::EventChannel::for_consumers(
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Acquires object reference for event channel, in order to connect consumer to event channel.

Since this method acquires area to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.20.2.2 CosEventChannelAdmin::EventChannel::for_suppliers()

Name

CosEventChannelAdmin::EventChannel::for_suppliers

Synopsis

```
#include  <EventService_cplus.h>
CosEventChannelAdmin::SupplierAdmin_ptr
CosEventChannelAdmin::EventChannel::for_suppliers(
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Acquires object reference for event channel, in order to connect supplier to event channel.

Since this method acquires area to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.20.2.3 CosEventChannelAdmin::EventChannel::destroy()

Name

CosEventChannelAdmin::EventChannel::destroy

Synopsis

```
#include  <EventService_cplus.h>
void  CosEventChannelAdmin::EventChannel::destroy(
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Destroys the specified event channel.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.20.2.4 CosEventChannelAdmin::ConsumerAdmin::obtain_push_supplier()

Name

CosEventChannelAdmin::ConsumerAdmin::obtain_push_supplier

Synopsis

```
#include  <EventService_cplus.h>
CosEventChannelAdmin::ProxyPushSupplier_ptr
CosEventChannelAdmin::ConsumerAdmin::obtain_push_supplier(
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Acquires object reference for event channel, in order to connect Push model consumer to event channel.

Since this method acquires area to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.20.2.5 CosEventChannelAdmin::ConsumerAdmin::obtain_pull_supplier()

Name

CosEventChannelAdmin::ConsumerAdmin::obtain_pull_supplier

Synopsis

```
#include  <EventService_cplus.h>
CosEventChannelAdmin::ProxyPullSupplier_ptr
CosEventChannelAdmin::ConsumerAdmin::obtain_pull_supplier(
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Acquires object reference for event channel, in order to connect Pull model consumer to event channel.

Since this method acquires area to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.20.2.6 CosEventChannelAdmin::SupplierAdmin::obtain_push_consumer()

Name

CosEventChannelAdmin::SupplierAdmin::obtain_push_consumer

Synopsis

```
#include  <EventService_cplus.h>
CosEventChannelAdmin::ProxyPushConsumer_ptr
CosEventChannelAdmin::SupplierAdmin::obtain_push_consumer(
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Acquires object reference for event channel, in order to connect Push model supplier to event channel.

Since this method acquires area to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.20.2.7 CosEventChannelAdmin::SupplierAdmin::obtain_pull_consumer()

Name

CosEventChannelAdmin::SupplierAdmin::obtain_pull_consumer

Synopsis

```
#include  <EventService_cplus.h>
CosEventChannelAdmin::ProxyPullConsumer_ptr
CosEventChannelAdmin::SupplierAdmin::obtain_pull_consumer(
     CORBA::Environment&     env )
     throw( CORBA::Exception );
```

Description

Acquires object reference for event channel, in order to connect Pull model supplier to event channel.

Since this method acquires area to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.20.2.8 CosEventChannelAdmin::ProxyPushConsumer::connect_push_supplier()

Name

CosEventChannelAdmin::ProxyPushConsumer::connect_push_supplier

Synopsis

```
#include  <EventService_cplus.h>
void  CosEventChannelAdmin::ProxyPushConsumer::connect_push_supplier(
     CosEventComm::PushSupplier_ptr     push_supplier,
     CORBA::Environment&                env )
     throw( CORBA::Exception );
```

Description

Connects Push model supplier to event channel.

Parameters

push_supplier

The object reference of the supplier itself.

If a disconnection notice is not required when the event channel is terminated, set CORBA_OBJECT_NIL in push_supplier.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. In the event of abnormal termination, the SystemException object reference or UserException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exception may be generated.

CosEventChannelAdmin::AlreadyConnected

Event channel already connected.

Note

To reconnect to an event channel, start again from CosEventChannelAdmin::SupplierAdmin::obtain_push_consumer.

2.20.2.9 CosEventChannelAdmin::ProxyPullSupplier::connect_pull_consumer()

Name

CosEventChannelAdmin::ProxyPullSupplier::connect_pull_consumer

Synopsis

```
#include  <EventService_cplus.h>
void  CosEventChannelAdmin::ProxyPullSupplier::connect_pull_consumer(
     CosEventComm::PullConsumer_ptr     pull_consumer,
     CORBA::Environment&                env )
     throw( CORBA::Exception );
```

Description

Connects Pull model consumer to event channel.

Parameters

pull_consumer

The object reference of the consumer itself.

If a disconnection notice is not required when the event channel is terminated, set CORBA_OBJECT_NIL in pull_consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. In the event of abnormal termination, the SystemException object reference or UserException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exception may be generated.

CosEventChannelAdmin::AlreadyConnected

Event channel already connected.

Note

To reconnect to an event channel, start again from CosEventChannelAdmin::ConsumerAdmin::obtain_pull_supplier.

2.20.2.10 CosEventChannelAdmin::ProxyPullConsumer::connect_pull_supplier()

Name

CosEventChannelAdmin::ProxyPullConsumer::connect_pull_supplier

Synopsis

```
#include  <EventService_cplus.h>
void  CosEventChannelAdmin::ProxyPullConsumer::connect_pull_supplier(
     CosEventComm::PullSupplier_ptr     pull_supplier,
     CORBA::Environment&                env )
     throw( CORBA::Exception );
```

Description

Connects Pull model supplier to event channel.

Parameters

pull_supplier

The object reference of the supplier itself.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. In the event of abnormal termination, the SystemException object reference or UserException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exception may be generated.

CosEventChannelAdmin::AlreadyConnected

Event channel already connected.

CosEventChannelAdmin::TypeError

The specified object type is incorrect.

Note

To reconnect to an event channel, start again from CosEventChannelAdmin::SupplierAdmin::obtain_pull_consumer.

2.20.2.11 CosEventChannelAdmin::ProxyPushSupplier::connect_push_consumer()

Name

CosEventChannelAdmin::ProxyPushSupplier::connect_push_consumer

Synopsis

```
#include  <EventService_cplus.h>
void  CosEventChannelAdmin::ProxyPushSupplier::connect_push_consumer(
     CosEventComm::PushConsumer_ptr     push_consumer,
```

```
    CORBA::Environment&                env )
    throw( CORBA::Exception );
```

Description

Connects Push model consumer to event channel.

Parameters

push_consumer

The object reference of the consumer itself.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env. In the event of abnormal termination, the SystemException object reference or UserException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exception may be generated.

CosEventChannelAdmin::AlreadyConnected

Event channel already connected.

CosEventChannelAdmin::TypeError

The specified object type is incorrect.

Note

To reconnect to an event channel, start again from CosEventChannelAdmin::ConsumerAdmin::obtain_push_supplier.

2.20.2.12 Classes Usable when Inherited

The following classes can be used by inheritance. For details refer to org.omg.CosEventComm Class.

CosEventChannelAdmin::ProxyPushConsumer::push

CosEventChannelAdmin::ProxyPushConsumer::disconnect_push_consumer

CosEventChannelAdmin::ProxyPushSupplier::disconnect_push_supplier

CosEventChannelAdmin::ProxyPullSupplier::pull

CosEventChannelAdmin::ProxyPullSupplier::try_pull

CosEventChannelAdmin::ProxyPullSupplier::disconnect_pull_supplier

CosEventChannelAdmin::ProxyPullConsumer::disconnect_pull_consumer

2.20.3 EventFactory Class

This section describes the EventFactory class.

2.20.3.1 EventFactory::create()

Name

EventFactory::create

Synopsis

```
#include  <EventService_cplus.h>
CosEventChannelAdmin::EventChannel_ptr
EventFactory::create(
     const CORBA::Char*              key,
     const EventFactory::Option&     data,
     CORBA::Environment&             env )
     throw( CORBA::Exception );
struct Option{
     CORBA::Long             max_queuing;
     CORBA::Long             life_time;
     EventFactory::Model     model;
  };
```

Description

This operation generates an event channel and returns the object reference of the generated event channel.

The EventFactory object reference is acquired by specifying the following value in the identifier parameter of the CORBA::ORB::resolve_initial_refernces() method.

EventFactory::ObjectId_Factory

Since this method acquires area to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

key

The common keyword for consumer and supplier (maximum 64 characters).

When the common keyword is specified, an object reference indicating the same event channel is returned.

data

Specify the EventFactory::Option class.

Specify the values in the following table for each member of the EventFactory::Option class.

Table 2.1 EventFactory::Option Class Members

Member	Set Value
max_queuing	When you set EventFactory::ES_DEFAULT_VALUE, it uses the "maximum value of event data number which can be stored in the event channel" set in the event service configuration information setting.
life_time	Data holding time (seconds). When you set EventFactory::ES_DEFAULT_VALUE, it uses the value of "existing time (sec.) of the stored event data" set in the event service configuration information setting. If 0 is set, then timeout monitoring is not implemented.
model	Specifies the following connection models: EventFactory::ModelAny.....Determine when connected EventFactory::ModelPush.....Push model EventFactory::ModelPull.....Pull model EventFactory::ModelMixed.....Mixed model

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env and the object reference of the generated event channel is retuned. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.20.3.2 EventFactory::create_channel()

Name

EventFactory::create_channel

Synopsis

```
#include  <EventService_cplus.h>
CosEventChannelAdmin::EventChannel_ptr
EventFactory::create_channel(
              const CORBA::Char*                  key,
              const EventFactory::Option&         data,
              const EventFactory::EventProperty&  property,
              const CORBA::boolean*               create,
              CORBA::Environment&                 env )
              throw( CORBA::Exception );
struct Option{
      CORBA::Long             max_queuing;
      CORBA::Long             life_time;
      EventFactory::Model     model;
};
struct EventFactory_Property {
      CORBA::Char*            name;
      CORBA::Any              value;
};
typedef CORBA::sequence::EventFactory_Property  EventFactory::EventProperty;
```

Description

This operation generates an event channel and returns the object reference of the generated event channel. This method is used to change the host name and port number for the generated Channel.

The EventFactory object reference is acquired by specifying the following value in the identifier parameter of the CORBA::ORB::resolve_initial_refernces() method.

EventFactory::ObjectId_Factory

Since this method acquires an area to store object references, use CORBA::release() to release the area as soon as it is no longer needed.

Parameters

key

The common keyword for consumer and supplier (maximum 64 characters).

When the common keyword is specified, an object reference indicating the same event channel is returned.

data

Specify the EventFactory::Option class.

Specify the values in the following table for each member of the EventFactory::Option class.

If the Event Channel has already been generated, the value for this parameter is ignored.

Table 2.2 EventFactory::Option Class Members

Member	Set Value
max_queuing	When you set EventFactory::ES_DEFAULT_VALUE, it uses the "maximum value of event data number which can be stored in the event channel" set in the event service configuration information setting.

Member	Set Value
life_time	Data holding time (seconds). When you set EventFactory::ES_DEFAULT_VALUE, it uses the value of "existing time (sec.) of the stored event data" set in the event service configuration information setting. If 0 is set, then timeout monitoring is not implemented.
model	Specifies the following connection models: EventFactory::ModelAny.....Determine when connected EventFactory::ModelPush.....Push model EventFactory::ModelPull.....Pull model EventFactory::ModelMixed.....Mixed model

property

Specifies the value shown in the table below. If an invalid name is specified, the corresponding record is invalid. If either of HostName and PortNumber is set, the specified record is invalid.

If the Event Channel has already been generated, the value for this parameter is ignored.

Member (name)	Data type (value)	Set value
HostName	string	Specifies the host name or the IP address (maximum 64 characters),.
PortNumber	unsigned short	Specifies the port number.

create

If the Event Channel was generated, CORBA::TRUE is set.

If the Event Channel already exists, CORBA::FALSE is set.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env and returns the object reference of the generated event channel. For abnormal termination, the SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

Notes

This method cannot be used if the following host names are set, as the Event Channel will fail to start.

- The "Corba Host Name" Interstage operating environment definition
- "IIOP_hostname" in the config file (CORBA Service) (The host name used by the CORBA Service)

When specifying the port number, select one of the following host names specified for the CORBA Service port number:

- If SSL communication is enabled
  - The "SSL Port Number" Interstage operating environment definition
  - "UNO_IIOP_ssl_port" in the config file (CORBA Service)
- If SSL communication is disabled
  - The "Corba Port Number" Interstage operating environment definition
  - "IIOP_port" in the config file (CORBA Service)

2.20.3.3 EventFactory::get_event_channel()

Name

EventFactory::get_event_channel

Synopsis

```
#include  <EventService_cplus.h>
CosEventChannelAdmin::EventChannel_ptr
EventFactory::get_event_channel(
                        CORBA::Char*         key,
                        CORBA::Environment&  env );
                    throw( CORBA::Exception );
```

Description

This operation generates a object reference of event channel specified with key

The EventFactory object reference is acquired by specifying the following value in the identifier parameter of the CORBA::ORB::resolve_initial_refernces() method.

EventFactory::ObjectId_Factory

Parameters

key

The acquired Event Channel name.

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env and the object reference of the event channel specified with key, is returned.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

In the event of a user exception, the following exceptions may be generated.

EventFactory::ChannelNotFound

The specified event channel could not be found.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.21 Notification Service Interface

This interface enables communication with the notification service event channel.

If the notification service interface is used for the event service event channel, a BAD_OPERATION system exception occurs.

Note

The notification service function can only be used in Interstage Application Server Enterprise Edition.

2.21.1 CosNotification Interface

Type definitions

The formats for the data types used in this module are as follows:

```
class CosNotification
{
    typedef CORBA::Char    *Istring;
```

```
typedef Istring        PropertyName;
typedef CORBA::Any     PropertyValue;
typedef CORBA::Any_var PropertyValue_var;

struct Property {
    CORBA::String_var    name;
    PropertyValue_var    value;
};
typedef Property *Property_ptr;
class Property_var {
public:
    Property_var();
    Property_var( Property* );
    Property_var( const Property & );
    Property_var( const Property_var & );
    ~Property_var();
};

class PropertySeq {
public:
    PropertySeq();
    PropertySeq( CORBA::ULong );
    PropertySeq( CORBA::ULong max,
                 CORBA::ULong length,
                 Property_var *data,
                 CORBA::Boolean release = CORBA_FALSE );
    PropertySeq( const PropertySeq &s );
    ~PropertySeq();
};
typedef PropertySeq *PropertySeq_ptr;
class PropertySeq_var {
public:
    PropertySeq_var();
    PropertySeq_var( PropertySeq_ptr );
    PropertySeq_var( const PropertySeq_var & );
    ~PropertySeq_var();
};

typedef PropertySeq OptionalHeaderFields;
typedef PropertySeq_ptr OptionalHeaderFields_ptr;
typedef PropertySeq_var OptionalHeaderFields_var;

typedef PropertySeq FilterableEventBody;
typedef PropertySeq_ptr FilterableEventBody_ptr;
typedef PropertySeq_var FilterableEventBody_var;

typedef PropertySeq QoSProperties;
typedef PropertySeq_ptr QoSProperties_ptr;
typedef PropertySeq_var QoSProperties_var;

typedef PropertySeq AdminProperties;
typedef PropertySeq_ptr AdminProperties_ptr;
typedef PropertySeq_var AdminProperties_var;

struct EventType {
    CORBA::String_var    domain_name;
    CORBA::String_var    type_name;
};
typedef EventType *EventType_ptr;
class EventType_var {
public:
    EventType_var();
    EventType_var( EventType* );
```

```
    EventType_var( const EventType & );
    EventType_var( const EventType_var & );
    ~EventType_var();
};

class EventTypeSeq {
public:
    EventTypeSeq();
    EventTypeSeq( CORBA::ULong );
    EventTypeSeq( CORBA::ULong max,
                  CORBA::ULong length,
                  EventType_var *data,
                  CORBA::Boolean release = CORBA_FALSE );
    EventTypeSeq( const EventTypeSeq &s );
    ~EventTypeSeq();
};
typedef EventTypeSeq *EventTypeSeq_ptr;
class EventTypeSeq_var {
public:
    EventTypeSeq_var();
    EventTypeSeq_var( EventTypeSeq_ptr );
    EventTypeSeq_var( const EventTypeSeq_var & );
    ~EventTypeSeq_var();
};

struct PropertyRange {
    PropertyValue_var    low_val;
    PropertyValue_var    high_val;
};
typedef PropertyRange *PropertyRange_ptr;
class PropertyRange_var {
public:
    PropertyRange_var();
    PropertyRange_var( PropertyRange* );
    PropertyRange_var( const PropertyRange & );
    PropertyRange_var( const PropertyRange_var & );
    ~PropertyRange_var();
};

struct NamedPropertyRange {
    CORBA::String_var    name;
    PropertyRange_var    range;
};
typedef NamedPropertyRange *NamedPropertyRange_ptr;

class NamedPropertyRangeSeq {
public:
    NamedPropertyRangeSeq();
    NamedPropertyRangeSeq( CORBA::ULong );
    NamedPropertyRangeSeq( CORBA::ULong max,
                           CORBA::ULong length,
                           NamedPropertyRange_var *data,
                           CORBA::Boolean release = CORBA_FALSE );
    NamedPropertyRangeSeq( const NamedPropertyRangeSeq &s );
    ~NamedPropertyRangeSeq();
};
typedef NamedPropertyRangeSeq *NamedPropertyRangeSeq_ptr;
class NamedPropertyRangeSeq_var {
public:
    NamedPropertyRangeSeq_var();
    NamedPropertyRangeSeq_var( NamedPropertyRangeSeq_ptr );
    NamedPropertyRangeSeq_var( const NamedPropertyRangeSeq_var & );
    ~NamedPropertyRangeSeq_var();
```

```
};

enum QoSError_code {
    UNSUPPORTED_PROPERTY,
    UNAVAILABLE_PROPERTY,
    UNSUPPORTED_VALUE,
    UNAVAILABLE_VALUE,
    BAD_PROPERTY,
    BAD_TYPE,
    BAD_VALUE
};

struct PropertyError {
    QoSError_code        code;
    CORBA::String_var    name;
    PropertyRange_var    available_range;
};
typedef PropertyError *PropertyError_ptr;
class PropertyError_var {
public:
    PropertyError_var();
    PropertyError_var( PropertyError* );
    PropertyError_var( const PropertyError & );
    PropertyError_var( const PropertyError_var & );
    ~PropertyError_var();
};

class PropertyErrorSeq {
public:
    PropertyErrorSeq();
    PropertyErrorSeq( CORBA::ULong );
    PropertyErrorSeq( CORBA::ULong max,
                      CORBA::ULong length,
                      PropertyError_var *data,
                      CORBA::Boolean release = CORBA_FALSE );
    PropertyErrorSeq( const PropertyErrorSeq &s );
    ~PropertyErrorSeq();
};
typedef PropertyErrorSeq *PropertyErrorSeq_ptr;
class PropertyErrorSeq_var {
public:
    PropertyErrorSeq_var();
    PropertyErrorSeq_var( PropertyErrorSeq_ptr );
    PropertyErrorSeq_var( const PropertyErrorSeq_var & );
    ~PropertyErrorSeq_var();
};

struct FixedEventHeader {
    EventType_var        event_type;
    CORBA::String_var    event_name;
};
typedef FixedEventHeader *FixedEventHeader_ptr;
class FixedEventHeader_var {
public:
    FixedEventHeader_var();
    FixedEventHeader_var( FixedEventHeader* );
    FixedEventHeader_var( const FixedEventHeader & );
    FixedEventHeader_var( const FixedEventHeader_var & );
    ~FixedEventHeader_var();
};

struct EventHeader {
    FixedEventHeader_var        fixed_header;
```

```
        OptionalHeaderFields_var    variable_header;
    };
    typedef EventHeader *EventHeader_ptr;
    class EventHeader_var {
    public:
        EventHeader_var();
        EventHeader_var( EventHeader* );
        EventHeader_var( const EventHeader & );
        EventHeader_var( const EventHeader_var & );
        ~EventHeader_var();
    };

    struct StructuredEvent {
        EventHeader_var            header;
        FilterableEventBody_var    filterable_data;
        CORBA::Any_var             remainder_of_body;
    };
    typedef StructuredEvent *StructuredEvent_ptr;
    class StructuredEvent_var {
    public:
        StructuredEvent_var();
        StructuredEvent_var( StructuredEvent* );
        StructuredEvent_var( const StructuredEvent & );
        StructuredEvent_var( const StructuredEvent_var & );
        ~StructuredEvent_var();
    };

};
```

2.21.2 QosAdmin Interface

This section describes the QosAdmin interface.

2.21.2.1 CosNotification::QoSAdmin::get_qos()

Name

CosNotification::QoSAdmin::get_qos

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotification::QoSProperties *
CosNotification::QoSAdmin::get_qos(
                      CORBA::Environment&       env );
                      throw( CORBA::Exception );
```

Description

This function fetches QosProperties.

This method secures an area for QoSProperties. Use delete to release the area when it is no longer required.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.21.2.2 CosNotification::QoSAdmin::set_qos()

Name

CosNotification::QoSAdmin::set_qos

Synopsis

```
#include  <NotificationService_cplus.h>
void
CosNotification::QoSAdmin::set_qos(
                      const CosNotification::QoSProperties&        qos,
                      CORBA::Environment&                          env );
                      throw( CORBA::Exception );
```

Description

This function sets QoSProperties.

Parameters

qos

The set QoSProperties.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

ex_CosNotification::UnsupportedQoS

There is an error in the QoSProperties item or value.

2.21.3 CosNotifyComm Module

This section describes the CosNotifyComm module.

2.21.3.1 CosNotifyComm::StructuredPushConsumer::push_structured_event()

Name

CosNotifyComm::StructuredPushConsumer::push_structured_event

Synopsis

```
#include  <NotificationService_cplus.h>
void
CosNotifyComm::StructuredPushConsumer::push_structured_event(
              const CosNotification::StructuredEvent&              data,
```

```
              CORBA::Environment&                                    env );
              throw( CORBA::Exception );
```

Description

Sends to the consumer the StructuredEvent type event data specified in data.

Parameters

data

The StructuredEvent type event data sent to consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

CosEventComm::Disconnected

The event channel was not connected.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

2.21.3.2 CosNotifyComm::StructuredPullSupplier::pull_structured_event()

Name

CosNotifyComm::StructuredPullSupplie::pull_structured_event

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotification::StructuredEvent *
CosNotifyComm::StructuredPullSupplier::pull_structured_event (
              CORBA::Environment&                    env );
              throw( CORBA::Exception );
```

Description

Requests from the supplier StructuredEvent type event data. Event data can be fetched or it may be blocked until an exception occurs. Use CosNotifyComm::StructuredPullSupplier::try_pull_structured_event to return immediately if event data is not fetched.

Specify the object reference returned by CosNotifyChannelAdmin::ConsumerAdmin::obtain_notification_pull_supplier in obj.

When the area secured for the StructuredEvent type data is no longer required, use delete to release the area.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and event data is returned from the supplier.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

CosEventComm::Disconnected

The event channel was not connected.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

2.21.3.3 CosNotifyComm::StructuredPullSupplier::try_pull_structured_event()

Name

CosNotifyComm::StructuredPullSupplier::try_pull_structured_event

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotification::StructuredEvent *
CosNotifyComm::StructuredPullSupplier::try_pull_structured_event (
              CORBA::boolean&                  has_event,
              CORBA::Environment&              env );
              throw( CORBA::Exception );
```

Description

Requests from the supplier StructuredEvent type event data. Immediately returns if event data is not fetched. Use CosNotifyComm::StructuredPullSupplier::pull_structured_event to block until event data is fetched.

When the area secured for the StructuredEvent type data is no longer required, use CORBA_free() to release the area.

Parameters

has_event (out parameter)

When event data is fetched, CORBA_TRUE is set.

When event data is not fetched, CORBA_FALSE is set.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference. When event data is fetched, CORBA_TRUE is set in has_event, and the event data from the supplier is returned. When event data is not fetched, CORBA_FALSE is set in has_event. In this case, because an area is secured for the StructuredEvent type event data, use delete to release the area.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

CosEventComm::Disconnected

The event channel was not connected.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

Notes

The cancellation need not be notified using ES::ChannelUtil::local_rollback() when CORBA_FALSE is set to has_event when a local transaction is operated.

Notifies the channel of completing using ES::ChannelUtil::local_commit().

The cancellation need not be notified using CosTransactions::Current::rollback() when CORBA_FALSE is set to has_event when a globaltransaction is operated.

Notifies the channel of completing using CosTransactions::Current::commit().

2.21.3.4 CosNotifyComm::StructuredPushConsumer::disconnect_structured_push_consumer()

Name

CosNotifyComm::StructuredPushConsumer::disconnect_structured_push_consumer

Synopsis

```
#include  <NotificationService_cplus.h>
void
CosNotifyComm::StructuredPushConsumer::disconnect_structured_push_consumer (
              CORBA::Environment&                    env );
              throw( CORBA::Exception );
```

Description

Declares the end of event transmission from the supplier.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.21.3.5 CosNotifyComm::StructuredPullSupplier::disconnect_structured_pull_supplier()

Name

CosNotifyComm::StructuredPullSupplier::disconnect_structured_pull_supplier

Synopsis

```
#include  <NotificationService_cplus.h>
void
CosNotifyComm::StructuredPullSupplier::disconnect_structured_pull_supplier (
              CORBA::Environment&                env );
              throw( CORBA::Exception );
```

Description

Declares the end of event transmission from the consumer.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.21.3.6 Inherited Interfaces

The following interfaces can be inherited and used.

For details, refer to 2.20.1 CosEventComm Class.

(1) CosNotifyComm::PushConsumer::push

(2) CosNotifyComm::PushConsumer::disconnect_push_consumer

(3) CosNotifyComm::PullSupplier::pull

(4) CosNotifyComm::PullSupplier::try::pull

(5) CosNotifyComm::PullSupplier::disconnect_pull_supplier

2.21.4 CosNotifyChannelAdmin Module

This section describes the CosNotifyChannelAdmin module.

2.21.4.1 CosNotifyChannelAdmin::ConsumerAdmin::obtain_notification_pull_supplier()

Name

CosNotifyChannelAdmin::ConsumerAdmin::obtain_notification_pull_supplier

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotifyChannelAdmin::ProxySupplier_ptr
CosNotifyChannelAdmin::ConsumerAdmin::obtain_notification_pull_supplier(
                        CosNotifyChannelAdmin::ClientType           ctype,
                        CosNotifyChannelAdmin::ProxyID&             proxy_id,
                          CORBA::Environment&                         env );
                          throw( CORBA::Exception );
```

Description

Creates a ProxySupplier object of the client type specified in ctype.

This method secures an area for holding the object references. Use CORBA::release() to release the area when it is no longer required.

Parameters

ctype

The client type by which the ProxySupplier object is created is specified as follows:

CosNotifyChannelAdmin::ANY_EVENT: Handles any type event

CosNotifyChannelAdmin::STRUCTURED_EVENT: Handles StructuredEvent type event

proxy_id (out parameter)

The created ProxySupplier ID is set.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and the ProxySupplier object reference is returned.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

CosNotifyChannelAdmin::AdminLimitExceeded

The upper limit for creating ProxySuppliers has been reached - no more can be created.

2.21.4.2 CosNotifyChannelAdmin::SupplierAdmin::obtain_notification_push_consumer()

Name

CosNotifyChannelAdmin::SupplierAdmin::obtain_notification_push_consumer

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotifyChannelAdmin::ProxyConsumer_ptr
CosNotifyChannelAdmin::SupplierAdmin::obtain_notification_push_consumer(
                        CosNotifyChannelAdmin::ClientType           ctype,
                        CosNotifyChannelAdmin::ProxyID&             proxy_id,
                          CORBA::Environment&                         env );
                          throw( CORBA::Exception );
```

Description

Creates a ProxyConsumer object of the client type specified in ctype.

This method secures an area for holding the object references. Use CORBA::release() to release the area when it is no longer required.

Parameters

ctype

The client type by which the ProxyConsumer object is created is specified as follows:

CosNotifyChannelAdmin::ANY_EVENT: Handles any type event

CosNotifyChannelAdmin::STRUCTURED_EVENT: Handles StructuredEvent type event

proxy_id (out parameter)

The created ProxyConsumer ID is set.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and the ProxyConsumer object reference is returned.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

CosNotifyChannelAdmin::AdminLimitExceeded

The upper limit for creating ProxyConsumers has been reached - no more can be created.

2.21.4.3 CosNotifyChannelAdmin::ConsumerAdmin::MyChannel()

Name

CosNotifyChannelAdmin::ConsumerAdmin::MyChannel

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotifyChannelAdmin::EventChannel_ptr
CosNotifyChannelAdmin::ConsumerAdmin::MyChannel(
                          CORBA::Environment&              env );
                          throw( CORBA::Exception );
```

Description

Returns the object reference for the event channel that generated ConsumerAdmin.

This method secures an area for holding the object references. Use CORBA::release() to release the area when it is no longer required.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and the Event Channel object reference is returned.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.21.4.4 CosNotifyChannelAdmin::SupplierAdmin::MyChannel()

Name

CosNotifyChannelAdmin::SupplierAdmin::MyChannel

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotifyChannelAdmin::EventChannel_ptr
CosNotifyChannelAdmin::SupplierAdmin::MyChannel(
                          CORBA::Environment&                env );
                          throw( CORBA::Exception );
```

Description

Returns the object reference for the event channel that generated SupplierAdmin.

This method secures an area for holding the object references. Use CORBA::release() to release the area when it is no longer required.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and the Event Channel object reference is returned.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.21.4.5 CosNotifyChannelAdmin::EventChannel::default_consumer_admin()

Name

CosNotifyChannelAdmin::EventChannel::default_consumer_admin

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotifyChannelAdmin::ConsumerAdmin_ptr
CosNotifyChannelAdmin::EventChannel::default_consumer_admin(
                         CORBA::Environment&               env );
                      throw( CORBA::Exception );
```

Description

Returns the object reference of the ConsumerAdmin object that is standard for the event channel.

This method secures an area for holding the object references. Use CORBA::release() to release the area when it is no longer required.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and the ConsumerAdmin object reference is returned.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.21.4.6 CosNotifyChannelAdmin::EventChannel::default_supplier_admin()

Name

CosNotifyChannelAdmin::EventChannel::default_supplier_admin

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotifyChannelAdmin::SupplierAdmin_ptr
CosNotifyChannelAdmin::EventChannel::default_supplier_admin(
                         CORBA::Environment&              env );
                       throw( CORBA::Exception );
```

Description

Returns the object reference of the SupplierAdmin object that is standard for the event channel.

This method secures an area for holding the object references. Use CORBA::release() to release the area when it is no longer required.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and the SupplierAdmin object reference is returned.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.21.4.7 CosNotifyChannelAdmin::ProxyPushConsumer::connect_any_push_supplier( )

Name

CosNotifyChannelAdmin::ProxyPushConsumer::connect_any_push_supplier

Synopsis

```
#include  <NotificationService_cplus.h>
void
CosNotifyChannelAdmin::ProxyPushConsumer::connect_any_push_supplier(
                     CosEventComm::PushSupplier_ptr              push_supplier,
                     CORBA::Environment&                         env );
                     throw( CORBA::Exception );
```

Description

Connects an any type supplier to the event channel.

Parameters

push_supplier

The own object reference.

Specify CosEventComm::PushSupplier::_nil in push_supplier if notification of disconnection is not required when the event channel is terminated.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

CosEventChannelAdmin::AlreadyConnected

The event channel is already connected.

Note

On reconnection to the event channel, start again from CosNotifyChannelAdmin::SupplierAdmin::obtain_notification_push_consumer.

2.21.4.8 CosNotifyChannelAdmin::ProxyPullSupplier::connect_any_pull_consumer()

Name

CosNotifyChannelAdmin::ProxyPullSupplier::connect_any_pull_consumer

Synopsis

```
#include  <NotificationService_cplus.h>
void
CosNotifyChannelAdmin::ProxyPullSupplier::connect_any_pull_consumer(
                      CosEventComm::PullConsumer_ptr              pull_consumer,
                      CORBA::Environment&                         env );
                      throw( CORBA::Exception );
```

Description

Connects an any type consumer to the event channel.

Parameters

pull_consumer

The own object reference.

Specify CosEventComm::PullConsumer::_nil in pull_consumer if notification of disconnection is not required when the event channel is terminated.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

CosEventChannelAdmin::AlreadyConnected

The event channel is already connected.

Note

On reconnection to the event channel, start again from CosNotifyChannelAdmin::ConsumerAdmin::obtain_notification_pull_supplier.

2.21.4.9 CosNotifyChannelAdmin::ProxyConsumer::MyType()

Name

CosNotifyChannelAdmin::ProxyConsumer::MyType

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotifyChannelAdmin::ProxyType
CosNotifyChannelAdmin::ProxyConsumer::MyType(
                      CORBA::Environment&                          env );
                        throw( CORBA::Exception );
```

Description

Returns the following values for the Proxy object type:

CosNotifyChannelAdmin::PUSH_ANY: any type PUSH model

CosNotifyChannelAdmin::PULL_ANY: any type PULL model

CosNotifyChannelAdmin::PUSH_STRUCTURED: StructuredEvent type PUSH model

CosNotifyChannelAdmin::PULL_STRUCTURED: StructuredEvent type PULL model

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and the Proxy object type is returned.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.21.4.10 CosNotifyChannelAdmin::ProxySupplier::MyType()

Name

CosNotifyChannelAdmin_ProxySupplier::MyType

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotifyChannelAdmin::ProxyType
CosNotifyChannelAdmin::ProxySupplier::MyType(
                      CORBA::Environment&                        env );
                      throw( CORBA::Exception );
```

Description

Returns the following values for the Proxy object type:

CosNotifyChannelAdmin::PUSH_ANY: any type PUSH model

CosNotifyChannelAdmin::PULL_ANY: any type PULL model

CosNotifyChannelAdmin::PUSH_STRUCTURED: StructuredEvent type PUSH model

CosNotifyChannelAdmin::PULL_STRUCTURED: StructuredEvent type PULL model

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and the Proxy object type is returned.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.21.4.11 CosNotifyChannelAdmin::StructuredProxyPushConsumer::connect_structured_ push_supplier()

Name

CosNotifyChannelAdmin_StructuredProxyPushConsumer_connect_structured_push_supplier

Synopsis

```
#include  <NotificationService_cplus.h>
void
CosNotifyChannelAdmin::StructuredProxyPushConsumer::connect_structured_push_supplier(
              CosNotifyComm::StructuredPushSupplier_ptr              push_supplier,
              CORBA::Environment&                                    env );
              throw( CORBA::Exception );
```

Description

Connects to the event channel as a StructuredEvent type supplier.

Parameters

push_supplier

The own object reference.

Specify CosNotifyComm::StructuredPushSupplier::_nil() in push_supplier if notification of disconnection is not required when the event channel is terminated.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

CosEventChannelAdmin::AlreadyConnected

The event channel is already connected.

Note

When reconnecting to the event channel, start again from CosNotifyChannelAdmin::SupplierAdmin::obtain_notification_push_consumer.

2.21.4.12 CosNotifyChannelAdmin::StructuredProxyPullSupplier::connect_structured_pull_consumer()

Name

CosNotifyChannelAdmin::StructuredProxyPullSupplier::connect_structured_pull_consumer

Synopsis

```
#include  <NotificationService_cplus.h>
void
CosNotifyChannelAdmin::StructuredProxyPullSupplier::connect_structured_pull_consumer(
              CosNotifyComm::StructuredPullConsumer_ptr              pull_consumer,
              CORBA::Environment&                                    env );
              throw( CORBA::Exception );
```

Description

Connects to the event channel as a StructuredEvent type consumer.

Parameters

pull_consumer

The own object reference.

Specify CosNotifyComm::StructuredPullConsumer::_nil() in pull_consumer if notification of disconnection is not required when the event channel is terminated.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

CosEventChannelAdmin::AlreadyConnected

The event channel is already connected.

Note

When reconnecting to the event channel, start again from CosNotifyChannelAdmin::SupplierAdmin::obtain_notification_pull_supplier.

2.21.4.13 Interfaces Inherited

The following interfaces can be inherited and used. For details, refer to 2.21.3 CosNotifyComm Module.

(1) CosNotifyChannelAdmin::StructuredProxyPushConsumer::push_structured_event

(2) CosNotifyChannelAdmin::StructuredProxyPullSupplier::pull_structured_event

(3) CosNotifyChannelAdmin::StructuredProxyPullSupplier::try_pull_structured_event

(4) CosNotifyChannelAdmin::StructuredProxyPushConsumer::disconnect_structured_push_consumer

(5) CosNotifyChannelAdmin::StructuredProxyPullSupplier::disconnect_structured_pull_supplier

(6) CosNotifyChannelAdmin::ProxyPushConsumer::push

(7) CosNotifyChannelAdmin::ProxyPullSupplier::pull

(8) CosNotifyChannelAdmin::ProxyPullSupplier::try_pull

(9) CosNotifyChannelAdmin::ProxyPushConsumer::disconnect_push_consumer

(10) CosNotifyChannelAdmin::ProxyPullSupplier::disconnect_pull_supplier

2.21.5 EventChannelFactory Interface

CosNotifyChannelAdmin::EventChannelFactory is the factory interface specified by the OMG- NotificationService. The old EventFactory interface cannot be used with the NotificationService-API.

2.21.5.1 CosNotifyChannelAdmin::EventChannelFactory::create_channel()

Name

CosNotifyChannelAdmin::EventChannelFactory::create_channel

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotifyChannelAdmin::EventChannel_ptr
CosNotifyChannelAdmin::EventChannelFactory::create_channel(
                      const CosNotification::QoSProperties&       initial_qos,
                      const CosNotification::AdminProperties&     initial_admin,
                      CosNotifyChannelAdmin::ChannelID&           id,
                      CORBA::Environment&                         env );
                        throw( CORBA::Exception );
```

Description

Generates an event channel with the QoS property items specified in initial_qos and initial_admin and the Admin property item, and specifies the event channel ID in id. Returns the event channel object reference as the return value.

The object reference specified in obj specifies and fetches the following value in the identifier parameter in CORBA::ORB::resolve_initial_references(): "NotificationService"

This method secures an area for holding the object references. Use CORBA::release() to release the area when it is no longer required.

Parameters

initial_qos

The QoS property item which generates event channel.

initial_admin

The Admin property item which generates event channel.

id (out parameter)

The event channel ID is set.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and returns object references for event channels.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

CosNotification::UnsupportedQoS

There is an error in the specified Qos property item or value.

CosNotification::UnsupportedAdmin

There is an error in the specified Admin property item or value.

2.21.5.2 CosNotifyChannelAdmin::EventChannelFactory::get_all_channels()

Name

CosNotifyChannelAdmin::EventChannelFactory::get_all_channels

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotifyChannelAdmin::ChannelIDSeq *
CosNotifyChannelAdmin::EventChannelFactory::get_all_channels(
                        CORBA::Environment&                          env );
                      throw( CORBA::Exception );
```

Description

Returns as sequence types the IDs for all event channels managed by the event factory.

This method secures sequence type areas for holding the object references. Use CORBA::delete() to release the areas when they are no longer required.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and returns in sequence type the IDs of all event channels that are managed by the event factory.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.21.5.3 CosNotifyChannelAdmin::EventChannelFactory::get_event_channel()

Name

CosNotifyChannelAdmin::EventChannelFactory::get_event_channel

Synopsis

```
#include  <NotificationService_cplus.h>
CosNotifyChannelAdmin::EventChannel_ptr
CosNotifyChannelAdmin::EventChannelFactory::get_event_channel(
                          CosNotifyChannelAdmin::ChannelID            id,
                          CORBA::Environment&                         env );
                      throw( CORBA::Exception );
```

Description

Returns the object references for event channels with the IDs specified in id.

This method secures areas for holding the object references. Use CORBA::delete() to release the area when it is no longer required.

Parameters

id

The acquired event channel ID.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and returns object references for event channels that have the IDs specified in id.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

CosNotifyChannelAdmin::ChannelNotFound

The specified channel could not be found.

2.22 Connection Information Collection Function Interface

This interface is used to collect connection information for the event channel.

2.22.1 CosEventChannelAdmin class

This section describes the CosEventChannelAdmin class.

2.22.1.1 CosEventChannelAdmin::EventChannel::create_util()

Name

CosEventChannelAdmin::EventChannel::create_util

Synopsis

```
#include  <EventService_cplus.h>
  CORBA::Object_ptr  CosEventChannelAdmin::EventChannel::create_util (
                        CosEventChannelAdmin::EventChannel::Utiltype     type,
                        CORBA::Environment&                              env )
                         throw( CORBA::Exception );
```

Description

Creates the ES_ChannelUtil interace object reference.

This method secures an area for holding object references. Use CORBA:: release() to release the area when it is no longer required.

Parameters

type

The interface type.

Specify the interface type in type.

Value Specifiable in type	Return Value Information
CosEventChannelAdmin::EventChannel::CHANNEL_UTIL	Fetches ES::ChannelUtil interface object reference

env

A structure that may contain exception information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env, and the ES_ChannelUtil interface object reference is returned.

For abnormal termination, a SystemException object reference is set in the exception member in env.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.22.1.2 Interface Inherited

The following interface can be inherited and used.

(1) CosNotifyChannelAdmin::EventChannel::create_util

2.22.2 ES Class

This section describes the ES class.

2.22.2.1 ES::ChannelUtil::get_proxys()

Name

ES::ChannelUtil::get_proxys

Synopsis

```
#include  <EventService_cplus.h>
  ES::ChannelUtil::ProxyDataSeq  ES::ChannelUtil::get_proxys (
                                    ES::ChannelUtil::ProxyType       kind,
                                    CORBA::Environment&              env )
                                  throw( CORBA::Exception );
  typedef struct {
        CORBA::Object_var          proxy;
        CORBA::Long                time;
        CORBA::Long                ipaddress;
        ES::ChannelUtil::ProxyKind kind;
  } ES::ChannelUtil::ProxyData;
  typedef sequence< ES::ChannelUtil::ProxyData> ES::ChannelUtil::ProxyDataSeq;
```

Description

Fetches consumers/suppliers connected to the event channel, and connection information.

This method secures areas for holding the connection information and suppliers/consumers connected to the event channel. Use delete to release the areas when they are no longer required.

Parameters

kind

The type of Proxy object fetched.

The connected information fetched depends on the value specified in kind. The following table lists the values specified in kind and the corresponding return inforrmation.

Table 2.3 Kind Values and Corresponding Return Information

Value Specifiable in Kind	Return Value Information
ES::ChannelUtil::ALL_PROXYS	Fetches all connection information
ES::ChannelUtil::PROXY_CONSUMER	Fetches a list of all ProxyConsumers
ES::ChannelUtil::PROXY_SUPPLIER	Fetches a list of all ProxySuppliers
ES::ChannelUtil::PROXY_PULL_CONSUMER_EVENT	Fetches a list of CosEventChannelAdmin ProxyPullConsumers
ES::ChannelUtil::PROXY_PULL_SUPPLIER _EVENT	Fetches a list of CosEventChannelAdmin ProxyPullSuppliers
ES::ChannelUtil::PROXY_PUSH_CONSUMER_EVENT	Fetches a list of CosEventChannelAdmin ProxyPushConsumers
ES::ChannelUtil::PROXY_PUSH_SUPPLIER _EVENT	Fetches a list of CosEventChannelAdmin ProxyPushSuppliers
ES::ChannelUtil::PROXY_PULL_CONSUMER_NOTIFY	Fetches a list of CosNotifyChannelAdmin ProxyPullConsumers
ES::ChannelUtil::PROXY_PULL_SUPPLIER _NOTIFY	Fetches a list of CosNotifyChannelAdmin ProxyPullSuppliers
ES::ChannelUtil::PROXY_PUSH_CONSUMER_NOTIFY	Fetches a list of CosNotifyChannelAdmin ProxyPushConsumers
ES::ChannelUtil::PROXY_PUSH_SUPPLIER_NOTIFY	Fetches a list of CosNotifyChannelAdmin ProxyPushSuppliers
ES::ChannelUtil::STRUCTURED_PROXY_PULL_ CONSUMER	Fetches a list of CosNotifyChannelAdmin StructuredProxyPullConsumers
ES::ChannelUtil::STRUCTURED_PROXY_PULL_ SUPPLIER	Fetches a list of CosNotifyChannelAdmin StructuredProxyPullSuppliers
ES::ChannelUtil::STRUCTURED_PROXY_PUSH_ CONSUMER	Fetches a list of CosNotifyChannelAdmin StructuredProxyPushConsumers
ES::ChannelUtil::STRUCTURED_PROXY_PUSH_ SUPPLIER	Fetches a list of CosNotifyChannelAdmin StructuredProxyPushSuppliers

The values in the following table can be specified in ES::ChannelUtil::ProxyData structure members:

Table 2.4 ES::ChannelUtil::ProxyData Structure Members and Settings

Member	Setting
Proxy	Object references for consumers/suppliers connected to the event channel
time	Time connected to the event channel
ipaddress	IP addresses of consumers/suppliers connected to the event channel
kind	Proxy types of consumers/suppliers connected to the event channel ES::ChannelUtil::PROXY_PULL_CONSUMER_EVENT ES::ChannelUtil::PROXY_PULL_SUPPLIER _EVENT ES::ChannelUtil::PROXY_PUSH_CONSUMER_EVENT ES::ChannelUtil::PROXY_PUSH_SUPPLIER _EVENT ES::ChannelUtil::PROXY_PULL_CONSUMER_NOTIFY ES::ChannelUtil::PROXY_PULL_SUPPLIER _NOTIFY ES::ChannelUtil::PROXY_PUSH_CONSUMER_NOTIFY

Member	Setting
	ES::ChannelUtil::PROXY_PUSH_SUPPLIER _NOTIFY
	ES::ChannelUtil::STRUCTURED_PROXY_PULL_CONSUMER
	ES::ChannelUtil::STRUCTURED_PROXY_PULL_SUPPLIER
	ES::ChannelUtil::STRUCTURED_PROXY_PUSH_CONSUMER
	ES::ChannelUtil::STRUCTURED_PROXY_PUSH_SUPPLIER

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and returns connection information for consumers/suppliers connected to the event channel.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

Notes

The 'Ipaddress' member is set to 0 when using the IP address in an IPv6 environment. In an IPv6 environment, use ES::ChannelUtil::get_proxys6().

2.22.2.2 ES::ChannelUtil::get_proxys6()

Name

ES::ChannelUtil::get_proxys6()

Synopsis

```
#include  <EventService_cplus.h>
ES::ChannelUtil::ProxyDataSeq  ES::ChannelUtil::get_proxys6 (
                                ES::ChannelUtil::ProxyType       kind,
                                CORBA::Environment&              env )
                               throw( CORBA::Exception );
typedef struct {
      CORBA::Object             proxy;
      CORBA::Long               time;
      CORBA::Octet              ipaddress[16];
      CORBA::Long               ip_format;
      ES::ChannelUtil::ProxyKind kind;
} ES::ChannelUtil::ProxyData6;
typedef sequence<ES::ChannelUtil::ProxyData6> ES::ChannelUtil::ProxyData6Seq;
```

Description

Fetches consumers/suppliers connected to the event channel and connection information.

When 'ipaddress' is set via the IPv6 form, 'ip_format' is set to 1. When 'ipaddress' is set via the IPv4 form, 'ip_format' is set to 0.

This method secures areas for storing the connection information and suppliers/consumers connected to the event channel. Use CORBA_free() to release these areas when they are no longer required.

Parameters

kind

The type of Proxy object fetched.

The connection information fetched depends on the value specified in *kind*. For details values that can be specified in *kind*, and the corresponding return inforrmation, refer to the Table 2.3 Kind Values and Corresponding Return Information lists under 2.22.2.1 ES::ChannelUtil::get_proxys().

env

A structure that can contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and connection information for consumers/suppliers connected to the event channel is returned.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

Notes

The IPv4 client member 'Ipaddress' is set to the parallel IPv4 address when operating using the IP address in an IPv6 environment. For details of settings when operating in an IPv6 environment, refer to 'Operating in an IPv6 Environment' and 'config'(IP-version parameter) under 'CORBA Service Environment Definition' of the Tuning Guide.

2.22.2.3 ES::ChannelUtil::get_consumer_count()

Name

ES::ChannelUtil::get_consumer_count

Synopsis

```
#include  <EventService_cplus.h>
  CORBA::ULong ES::ChannelUtil::get_consumer_count(
                                    CORBA::Environment&              env )
                                   throw( CORBA::Exception );
```

Description

Fetches the number of consumers connected to the event channel.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and returns the number of consumers connected to the event channel.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.22.2.4 ES::ChannelUtil::get_supplier_count()

Name

ES::ChannelUtil::get_supplier _count

Synopsis

```
#include  <EventService_cplus.h>
  CORBA::ULong ES::ChannelUtil::get_supplier_count(
                                    CORBA::Environment&             env )
                                   throw( CORBA::Exception );
```

Description

Fetches the number of suppliers connected to the event channel.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and the number of suppliers connected to the event channel is returned.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.22.2.5 ES::ChannelUtil::get_queue_length()

Name

ES::ChannelUtil::get_queue _length

Synopsis

```
#include  <EventService_cplus.h>
  CORBA::ULong ES::ChannelUtil::get_queue_length(
                                    CORBA::Environment&             env )
                                   throw( CORBA::Exception );
```

Description

Fetches the number of event data items queued in the event channel.

Parameters

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference, and the number of event data items queued in the event channel is returned.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.22.2.6 ES::ChannelUtil::local_begin()

Name

ES::ChannelUtil::local_begin

Synopsis

```
#include  <NotificationService_cplus.h>
void
ES::ChannelUtil::local_begin(
              CORBA::Object_ptr  proxy,
              CORBA::Environment&  env )
           throw( CORBA::Exception );
```

Description

Notifies the channel of starting the local transaction.

In the case of consumer, the following methods can be issued before ES::ChannelUtil::local_commit() or ES::ChannelUtil::local_rollback() is issued.

- CosEventComm::PullSupplier::pull()
- CosEventComm::PullSupplier::try_pull()
- CosNotifyComm::StructuredPullSupplier::pull_structured_event()
- CosNotifyComm::StructuredPullSupplier::try_pull_structured_event()

In the case of supplier, the following methods can be issued before ES::ChannelUtil::local_commit() or ES::ChannelUtil::local_rollback() is issued.

- CosEventComm::PushConsumer::push()
- CosNotifyComm::StructuredPushConsumer::push_structured_event()

If ES::ChannelUtil::local_commit() or ES::ChannelUtil::local_rollback() is not issued after issuance of this method, they are automatically rolled back by the notification service immediately after the local transaction timeout time passes.

This method cannot be issued twice for a proxy.

Parameters

proxy

The object reference returned by CosNotifyChannelAdmin::ConsumerAdmin::obtain_notification_pull_supplier or CosNotifyChannelAdmin::SupplierAdmin::obtain_notification_push_consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.22.2.7 ES::ChannelUtil::local_commit()

Name

ES::ChannelUtil::local_commit

Synopsis

```
#include  <NotificationService_cplus.h>
void
ES::ChannelUtil::local_commit(
              CORBA::Object_ptr  proxy,
              CORBA::Environment&  env )
            throw( CORBA::Exception );
```

Description

Notifies the channel of completing the local transaction.

In the case of consumer, message reception from the event channel completes when this method completes.

In the case of supplier, sending of data from the event channel completes when this method completes.

Parameters

proxy

The object reference returned by CosNotifyChannelAdmin::ConsumerAdmin::obtain_notification_pull_supplier or CosNotifyChannelAdmin::SupplierAdmin::obtain_notification_push_consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.22.2.8 ES::ChannelUtil::local_rollback()

Name

ES::ChannelUtil::local_rollback

Synopsis

```
#include  <NotificationService_cplus.h>
void
ES::ChannelUtil::local_rollback(
              CORBA::Object_ptr  proxy,
              CORBA::Environment&  env )
            throw( CORBA::Exception );
```

Description

Notifies the channel of cancellation of the local transaction. Issuance of this method allows the event channel status to return to that before ES::ChannelUtil::local_begin() is issued.

Parameters

proxy

The object reference returned by CosNotifyChannelAdmin::ConsumerAdmin::obtain_notification_pull_supplier or CosNotifyChannelAdmin::SupplierAdmin::obtain_notification_push_consumer.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.22.2.9 ES::ChannelUtil::pull_cancel()

Name

ES::ChannelUtil::pull_cancel

Synopsis

```
#include  <NotificationService_cplus.h>
void
ES::ChannelUtil::pull_cancel(
              CORBA::Object_ptr  proxy,
              CORBA::Environment&  env )
            throw( CORBA::Exception );
```

Description

This method cancels the pull waiting status (waiting status by issuance of pull and ES::ChannelUtil::pull_wait()) of the target proxy.

Parameters

proxy

The object reference returned by CosNotifyChannelAdmin::ConsumerAdmin::obtain_notification_pull_supplier.

Note

Specify the object reference that was specified in pull and ES::ChannelUtil::pull_wait().

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

2.22.2.10 ES::ChannelUtil::pull_wait()

Name

ES::ChannelUtil::pull_wait

Synopsis

```
#include  <NotificationService_cplus.h>
void
ES::ChannelUtil::pull_wait(
              CORBA::Object_ptr  proxy,
              CORBA::Environment&  env )
            throw( CORBA::Exception );
```

Description

This method waits until the status of the target proxy becomes enabled to allow the proxy to fetch data with the pull method or the try_pull method. Returns to the invoking side when the status becomes enabled to allow fetching data after execution of this method.

Returns immediately if the target proxy has already been enabled to fetch data with the pull method. If the status does not become enabled to allow fetching data within the waiting time, this method returns with timeout.

Parameters

proxy

The object reference returned by CosNotifyChannelAdmin::ConsumerAdmin::obtain_notification_pull_supplier.

env

A structure that may contain exception information.

Return Values

For normal termination, the exception member of env is set to a NULL object reference.

For abnormal termination, the exception member of env is set to a SystemException object reference or a UserException object reference.

Refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual to correct system exceptions.

In the event of a user exception, the following exceptions may be generated.

CosEventComm::Disconnected

Not connected to an event channel.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

2.23 Other Distributed Transaction Linkage Interfaces Windows32 Solaris32 Linux32

This section describes the distributed transaction linkage interfaces not describes in the previous sections.

2.23.1 TransactionFactory Interface Windows32 Solaris32 Linux32

This section describes the TransactionFactory interface.

2.23.1.1 CosTransactions::TransactionFactory::create Windows32 Solaris32 Linux32

Name

CosTransactions::TransactionFactory::create

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
CosTransaction::Control_ptr  CosTransactions::TransactionFactory::create(
                        CORBA::ULong                time_out,
                        CORBA::Environment&         env);
```

Description

Generates a new transaction. Generates a Control object for the transaction context and returns the object reference.

Used to create functions equivalent to those of begin in the Current interface. For that reason, this function cannot be used in conjunction with the Current interface.

Specify the transaction timeout time in time_out. If the CosTransactions::Current::set_timeout method is used before the transaction starts, specify the time specified in that method.

Return Values

For normal termination, a NULL object reference is set in the exception member in env.

For abnormal termination, a SystemException or UserException object reference is set in the exception member in env.

The following detailed information is set when a system exception occurs:

CORBA::StExcep::NO_IMPLEMENT:

The event channel has not been started.

CORBA::StExcep::COMM_FAILURE:

There was a communications failure.

CORBA::StExcep::NO_RESOURCES:

There are insufficient resources.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

CORBA::StExcep::INTERNAL:

CORBA Service error.

2.23.1.2 CosTransactions::TransactionFactory::recreate Windows32 Solaris32 Linux32

Name

CosTransactions::TransactionFactory::recreate

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
CosTransactions::Control_ptr  CosTransactions::TransactionFactory::recreate(
                        CosTransactions::PropagationContext& ctx,
                        CORBA::Environment&            *env);
```

Description

Generates a Control object for the transaction context and returns the object reference.

This Control object can also be used to terminate transactions in server applications.

Specify the PropagationContext fetched by the CosTransactions::Coordinator::get_txcntext function in ctx.

Return Values

For normal termination, a NULL object reference is set in the exception member in env.

For abnormal termination, a SystemException object reference is set in the exception member in env.

The following detailed information is set when a system exception occurs:

CORBA::StExcep::NO_IMPLEMENT

The event channel has not been started.

CORBA::StExcep::COMM_FAILURE

There was a communications failure.

CORBA::StExcep::NO_RESOURCES

There are insufficient resources.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

CORBA::StExcep::INTERNAL:

CORBA Service error.

2.23.2 Control Interface Windows32 Solaris32 Linux32

This section describes the control interface.

2.23.2.1 CosTransactions::Control::get_terminator Windows32 Solaris32 Linux32

Name

CosTransactions::Control::get_terminator

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
CosTransactions::Terminator CosTransactions::Control::get_terminator(
                                CORBA::Environment&       env);
```

Description

Returns a Terminator object that terminates transactions.

This Terminator object can also be used to terminate transactions from server applications.

Return Values

For normal termination, a NULL object reference is set in the exception member in env.

For abnormal termination, a SystemException or UserException object reference is set in the exception member in env.

The following detailed information is set when a user exception occurs:

CosTransactions::Unavailable

The Terminator object cannot be used.

The following detailed information is set when a system exception occurs:

CORBA::StExcep::NO_IMPLEMENT:

The event channel has not been started.

CORBA::StExcep::COMM_FAILURE:

There was a communications failure.

CORBA::StExcep::NO_RESOURCES:

There are insufficient resources.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

CORBA::StExcep::INTERNAL:

CORBA Service error.

2.23.2.2 CosTransactions::Control::get_coordinator Windows32 Solaris32 Linux32

Name

CosTransactions::Control::get_coordinator

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
CosTransactions::Coordinator  CosTransactions::Control::get_coordinator(
                                CORBA::Environment&       env);
```

Description

Returns a Coordinator object that coordinates transactions.

This Coordinator object can also be used to coordinate transactions in server applications.

Return Values

For normal termination, a NULL object reference is set in the exception member in env.

For abnormal termination, a SystemException or UserException object reference is set in the exception member in env.

The following detailed information is set when a user exception occurs:

CosTransactions::Unavailable

The Coordinator object cannot be used.

The following detailed information is set when a system exception occurs:

CORBA::StExcep::NO_IMPLEMENT:

The event channel has not been started.

CORBA::StExcep::COMM_FAILURE:

There was a communications failure.

CORBA::StExcep::NO_RESOURCES:

There are insufficient resources.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

CORBA::StExcep::INTERNAL:

CORBA Service error.

2.23.3 Coordinator Interface Windows32 Solaris32 Linux32

In the Coordinator interface, the following functions are the same functions as in the Current interface. Refer to the Current interface for these specifications.

- get_status
- rollback_only
- get_transaction_name

2.23.3.1 CosTransactions::Coordinator::register_synchronization Windows32 Solaris32 Linux32

Name

CosTransactions::Coordinator::register_synchronization

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
void  CosTransactions::Coordinator::register_synchronization(
                                CosTransactions::Synchronization sync,
                                CORBA::Environment&       env);
```

Description

Registers the Synchronization object in the transaction.

Specifies in sync the object reference for the Synchronization object created by the user. The Synchronization object is created by using the Synchronization interface.

The Synchronization object is called when it is registered, and before and after a transaction is committed.

Return Values

For normal termination, a NULL object reference is set in the exception member in env.

For abnormal termination, a SystemException or UserException object reference is set in the exception member in env.

The following detailed information is set when a user exception occurs:

CosTransactions::Inactive

The transaction has already been prepared.

CosTransactions::SynchronizationUnavailable

The Coordinator does not support the Synchronization interface.

The following detailed information is set when a system exception occurs:

CORBA::StExcep::NO_IMPLEMENT:

The event channel has not been started.

CORBA::StExcep::COMM_FAILURE:

There was a communications failure.

CORBA::StExcep::NO_RESOURCES:

There are insufficient resources.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

CORBA::StExcep::INTERNAL:

CORBA Service error.

2.23.3.2 CosTransactions::Coordinator::get_txcontext Windows32 Solaris32 Linux32

Name

CosTransactions::Coordinator::get_txcontext

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
```

```
CosTransactions::PropagationContext  CosTransactions::Coordinator::get_txcontext(
                                    CORBA::Environment&       env);
```

Description

Returns the transaction context that manages the transaction information.

Return Values

For normal termination, a NULL object reference is set in the exception member in env.

For abnormal termination, a SystemException or UserException object reference is set in the exception member in env.

The following detailed information is set when a user exception occurs:

CosTransactions::Unavailable

The Coordinator object cannot be used.

The following detailed information is set when a system exception occurs:

CORBA::StExcep::NO_IMPLEMENT:

The event channel has not been started.

CORBA::StExcep::COMM_FAILURE:

There was a communications failure.

CORBA::StExcep::NO_RESOURCES:

There are insufficient resources.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

CORBA::StExcep::INTERNAL:

CORBA Service error.

2.23.4 Terminator Interface Windows32 Solaris32 Linux32

The functions provided by the Terminator interface are identical to the commit and rollback functions of the Current interface. Thus, this interface cannot be used in conjunction with the Current interface.

2.23.4.1 CosTransactions::Terminator::commit Windows32 Solaris32 Linux32

Name

CosTransactions::Terminator::commit

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
void  CosTransactions::Terminator::commit(
                        CORBA::boolean       report_heuristics,
                        CORBA::Environment&      env);
```

Description

Commits a transaction.

A heuristic error is posted if CORBA_TRUE is specified in report_heuristics. Used to create functions equivalent to those of commit in the Current interface. For that reason, cannot be used in conjunction with the Current interface.

Return Values

For normal termination, a NULL object reference is set in the exception member in env.

For abnormal termination, a SystemException or UserException object reference is set in the exception member in env.

The following detailed information is set when a user exception occurs:

CosTransactions::HeuristicMixed:

The completion statuses of the transactions are that some have been committed and some have been rolled back.

CosTransactions::HeuristicHazard:

The completion status of the transaction is uncertain.

The following detailed information is set when a system exception occurs:

CORBA::StExcep::TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

CORBA::StExcep::NO_IMPLEMENT:

The event channel has not been started.

CORBA::StExcep::COMM_FAILURE:

There was a communications failure.

CORBA::StExcep::NO_RESOURCES:

There are insufficient resources.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

CORBA::StExcep::INTERNAL:

CORBA Service error.

2.23.4.2 CosTransactions::Terminator::rollback Windows32 Solaris32 Linux32

Name

CosTransactions::Terminator::rollback

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
void  CosTransactions::Terminator::rollback(
                                CORBA::Environment&      env);
```

Description

Rolls back a transaction.

Used to create functions equivalent to those of rollback in the Current interface. For that reason, cannot be used in conjunction with the Current interface.

Return Values

For normal termination, a NULL object reference is set in the exception member in env.

For abnormal termination, a SystemException object reference is set in the exception member in env.

The following detailed information is set when a system exception occurs:

CORBA::StExcep::TRANSACTION_ROLLEDBACK:

The transaction has been rolled back.

CORBA::StExcep::NO_IMPLEMENT:

The event channel has not been started.

CORBA::StExcep::COMM_FAILURE:

There was a communications failure.

CORBA::StExcep::NO_RESOURCES:

There are insufficient resources.

CORBA::StExcep::NO_MEMORY:

Failed to secure dynamic memory.

CORBA::StExcep::INTERNAL:

CORBA Service error.

2.23.5 Synchronization Interface Windows32 Solaris32 Linux32

This section describes the synchronization interface.

2.23.5.1 CosTransactions::Synchronization::before_completion Windows32 Solaris32 Linux32

Name

CosTransactions::Synchronization::before_completion

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
CosTransactions::Synchronization::before_completion(
                                CORBA::Environment&   env);
                                throw( CORBA::Exception );
```

Description

This function is used to synchronize transactions that have been created. When the Database Linkage Service receives a commit request, it executes before a commit request to a resource manager.

Return Values

For normal termination, processing continues.

For abnormal termination, the Database Linkage Service rolls the transaction back.

2.23.5.2 CosTransactions::Synchronization::after_completion Windows32 Solaris32 Linux32

Name

CosTransactions::Synchronization::after_completion

Synopsis

```
#include  "orb_cplus.h"
#include  "CosTransactions_cplus.h"
CosTransactions::Synchronization::after_completion(
                                CosTransactions::Status  status,
                                CORBA::Environment&       env);
                                throw( CORBA::Exception );
```

Description

This function is used to synchronize transactions that have been created. When the Database Linkage Service receives a commit or rollback request, it executes the commit or rollback request after a request has been received from a resource, and the transaction has been completed.

The transaction status is set in status. The transaction status is the same as that referenced with the get_status method.

Return Values

For normal or abnormal termination, processing continues, and a commit or rollback request is returned to the requesting user application.

Chapter 3 Java Interface

This chapter describes the Java interface.

3.1 The ORB Class

This section explains the ORB (Object Request Broker) class providing the basic interface for initializing ORB. The ORB function mediates communications between clients and servers.

The APIs which are supported by the CORBA Service (ObjectDirector) and IIOP Service (Java EE client) in the ORB class are shown below.

API	CORBA Service (ObjectDirector)	IIOP Service (Java EE client)
org.omg.CORBA.ORB.init()	Y	Y
org.omg.CORBA.ORB.list_initial_services()	Y	Y
org.omg.CORBA.ORB.resolve_initial_references()	Y	Y
org.omg.CORBA.ORB.resolve_initial_references_remote()	Y (unique to Fujitsu)	-
org.omg.CORBA.ORB.object_to_string()	Y	Y
org.omg.CORBA.ORB.string_to_object()	Y	Y
org.omg.CORBA.ORB.create_list()	Y	X
org.omg.CORBA.ORB.create_operation_list()	Y	X
org.omg.CORBA.ORB.create_named_value()	Y	X
org.omg.CORBA.ORB.create_exception_list()	Y	X
org.omg.CORBA.ORB.create_context_list()	Y	X
org.omg.CORBA.ORB.get_default_context()	Y	X
org.omg.CORBA.ORB.create_environment()	Y	X
org.omg.CORBA.ORB.send_multiple_requests_oneway()	Y	X
org.omg.CORBA.ORB.send_multiple_requests_deferred()	Y	X
org.omg.CORBA.ORB.poll_next_response()	Y	X
org.omg.CORBA.ORB.get_next_response()	Y	X
org.omg.CORBA.ORB.get_primitive_tc()	Y	Y
org.omg.CORBA.ORB.create_any()	Y	Y
org.omg.CORBA.ORB.run()	Y	X
org.omg.CORBA.ORB.destroy()	Y (portable-ORB only)	Y
org.omg.CORBA.ORB.create_array_tc()	Y	Y
org.omg.CORBA.ORB.create_enum_tc()	Y	Y
org.omg.CORBA.ORB.create_exception_tc()	Y	Y
org.omg.CORBA.ORB.create_sequence_tc()	Y	Y
org.omg.CORBA.ORB.create_string_tc()	Y	Y
org.omg.CORBA.ORB.create_struct_tc()	Y	Y
org.omg.CORBA.ORB.create_union_tc()	Y	Y
org.omg.CORBA.ORB.create_wstring_tc()	Y	Y

```
//Java
package org.omg.CORBA;
public abstract class ORB
{
    public static org.omg.CORBA.ORB
        init(java.lang.String[] args, java.util.Properties props){...};
    public static org.omg.CORBA.ORB
        init(java.applet.Applet app, java.util.Properties props){...};
    public static org.omg.CORBA.ORB  init( ){...};
    public abstract java.lang.String[]  list_initial_services( );
    public abstract org.omg.CORBA.Object
        resolve_initial_references(java.lang.String object_name)
        throws org.omg.CORBA.ORBPackage.InvalidName;
    public abstract java.lang.String  object_to_string(org.omg.CORBA.Object obj);
    public abstract org.omg.CORBA.Object  string_to_object(java.lang.String str);
    public abstract org.omg.CORBA.NVList  create_list(int count);
    public abstract org.omg.CORBA.NVList
        create_operation_list(org.omg.CORBA.OperationDef oper);
    public abstract org.omg.CORBA.NamedValue
        create_named_value(java.lang.String name, Any value, int flags);
    public abstract org.omg.CORBA.ExceptionList  create_exception_list( );
    public abstract org.omg.CORBA.ContextList  create_context_list( );
    public abstract org.omg.CORBA.Context get_default_context( );
    public abstract org.omg.CORBA.Environment  create_environment( );
    public abstract void  send_multiple_requests_oneway(org.omg.CORBA.Request[] req);
    public abstract void  sent_multiple_requests_deferred(org.omg.CORBA.Request[] req);
    public abstract boolean  poll_next_response( );
    public abstract org.omg.CORBA.Request  get_next_response( );
    public abstract org.omg.CORBA.TypeCode  get_primitive_tc(org.omg.CORBA.TCKind tcKind);
    public abstract org.omg.CORBA.Any  create_any( );
    public void  run( );
    public void  destroy( );
    abstract public org.omg.CORBA.TypeCode
        create_array_tc(int length, org.omg.CORBA.TypeCode element_type);
    abstract public org.omg.CORBA.TypeCode
        create_enum_tc(java.lang.String id, java.lang.String name, java.lang.String[] members);
    abstract public org.omg.CORBA.TypeCode
        create_exception_tc(java.lang.String id, java.lang.String name, org.omg.CORBA.StructMember[]
members);
    abstract public org.omg.CORBA.TypeCode
        create_sequence_tc(int bound, org.omg.CORBA.TypeCode element_type);
    abstract public org.omg.CORBA.TypeCode
        create_string_tc(int bound);
    abstract public org.omg.CORBA.TypeCode
        create_struct_tc(java.lang.String id, java.lang.String name, org.omg.CORBA.StructMember[]
members);
    abstract public org.omg.CORBA.TypeCode
        create_union_tc(java.lang.String id, java.lang.String name, org.omg.CORBA.TypeCode
discriminator_type, org.omg.CORBA.UnionMember[] members);
    abstract public org.omg.CORBA.TypeCode
        create_wstring_tc(int bound);
}
```

3.1.1 org omg.CORBA.ORB.init ()

Name

org.omg.CORBA.ORB.init

Format

```
(1) public  static org.omg.CORBA.ORB init()
(2) public  static org.omg.CORBA.ORB init(
        java.lang.String[] args,
        java.util.Properties props )
(3) public  static org.omg.CORBA.ORB init(
        java.applet.Applet app,
        java.util.Properties props )
```

Description

(1) Returns an instance of ORB (Singleton ORB). It is usually used for specific ORB class method calls.

(2) Generates a new ORB instance for an application. This method can only be called from an application.

(3) Generates a new ORB instance for an applet. This method can only be called from an applet. It can be used in the pre-installed Java library and Portable-ORB.

Distributed transaction linkage

- When using Java EE

  The IIOP Service cannot be used to execute distributed transaction linkage.

- When using the pre-installed Java library

  To perform distributed transaction linkage using the pre-installed Java library, add the following specification to the property:

  com.fujitsu.ObjectDirector.CORBA.GlobalTransactionMode=true

  Retrieve the property in the following order:

  1. The props parameter of format (2) and (3) of this method.
  2. The system property passed when the JVM is started.

- When using the Portable-ORB

  Distributed transaction linkage cannot be performed using the Portable-ORB.

Parameters

(1) None

(2)

args

The maximum number of command line arguments that can be given to an application's main method. Null Usable

props

application-specific properties null Usable.

(3)

app

Applet null Unusable props - Applet-specific properties null Usable.

props

Applet-specific properties. Null permitted.

To fetch the client's individual initial service object reference with org.omg.CORBA.ORB.resolve.initial.references(), the value must be set in args. Refer to the Distributed Application Development Guide (CORBA Service Edition) for details.

Specifying ORB arguments

(1) Specification in the args parameter

To specify an ORB argument in the args parameter when the java command is run, specify the argument list transferred to the main function without modification in args.

The following is an example of the program coding when the java command is run with the ORBInitRef argument specified:

```
> java Sample -ORBInitRef NameService=corbaloc::nshost/NameService

  public class Sample
  {
      public static void main( String[] args ) {
          org.omg.CORBA.ORB Orb;
          Orb = org.omg.CORBA.ORB.init(args, null);
                :
                :
      }
}
```

Always specify in the args parameter the argument list transferred to the main function, regardless of whether the ORB argument is present.

(2) Specification in the app parameter (When the pre-installed Java library/Portable-ORB are used)

Use the <PARAM> tab in the HTML. The information for the ORB argument specified in the HTML is passed to ORB by specifying in the appl parameter the applet class when the applet is executed.

The following is an example of the coding when the JBK plugin is used, when the ORBInitRef argument is specified:

```
<HTML>
  <HEAD><!--sample.html-->
  <TITLE>Java sample Applet </TITLE>
  </HEAD>
  <BODY>
  <OBJECT CLASSID="CLSID:BEA62964-C40B-11D1-AACA-00A0C9216A67"  WIDTH=300 HEIGHT=250>
    <PARAM NAME="TYPE" VALUE="application/x-JBK-Plugin">
    <PARAM NAME="CODE" VALUE="Sample.class">
    <PARAM NAME="ORBInitRef" VALUE="NameService=corbaloc::nshost/NameService">
    <COMMENT>
      <EMBED TYPE="application/x-JBK-Plugin"
        CODE="Sample.class" WIDTH=300 HEIGHT=250
        ORBInitRef="NameService=corbaloc::nshost/NameService" >
      </EMBED>
    </COMMENT>
  </OBJECT>
  </BODY>
  </HTML>
```

When specifying the <EMBED> tab, code as "ORB argument name = argument name", and do not use the <PARAM> tab. There is no need to read this ORB information in the applet program. The information is read automatically in ORB.

Always specify the applet class in the app parameter, whether the ORB argument is used or not.

When multiple ORBInitRef and ORBDefaultInitRef arguments are specified, separate them using spaces as delimiters in the VALUE argument. Refer to the Distributed Application Development Guide (CORBA Service Edition).

Return Values

Normal termination: Returns the ORB object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

Notes

- Issue the init() method which has the format shown in either (2) or (3) once only in the application/applet.

- When the pre-installed type Java liabrary is used, the parameter information, etc. is ignored in the second and the subsequent callings. For an applet, only the init() method that was issued first, becomes valid when moving to the HTML of a different browser.
- ORB object returned by init() method described at the format (1) above, is called Singleton ORB, and available method types are limited. The available methods are in the following.

```
  create_list()
  create_operation_list()
  create_named_value()
  create_exception_list()
  create_context_list()
  create_environment()
  create_any()
  get_primitive_tc()
  create_array_tc()
  create_enum_tc()
  create_enum_tc()
  create_exception_tc()
  create_sequence_tc()
  create_string_tc()
  create_struct_tc()
  create_union_tc()
  create_wstring_tc()
```

Example

- Example for HTTP tunneling.

  This can be used in the pre-installed Java library or Portable-ORB.

```
String  args[] = {
    "",    /* The first argument is ignored */
    "-ORB_FJ_HTTP", "yes",
    "-ORB_FJ_HTTPGW", "http://host.com/od-httpgw",
    null };

org.omg.CORBA.ORB  orb;

orb = org.omg.CORBA.ORB.init( args, null );
```

- Example for initial service.

```
String  args[] = {
    "",    /* The first argument is ignored */
    "-ORBInitRef", "NameService=corbaloc::nshost:8002/NameService",
    "-ORBDefaultInitRef", "corbaloc::inithost:8002",
    null };

org.omg.CORBA.ORB  orb;

orb = org.omg.CORBA.ORB.init( args, null );
```

3.1.2 org.omg.CORBA.ORB.list_initial_services()

Name

org.omg.CORBA.ORB.list_initial_services

Format

```
public  java.lang.String[] list_initial_services()
```

Description

Returns an identifier list containing "NameService", "InterfaceRepository" and other usable services.

Return Values

Normal termination: Sets up a string array that displays the identifier for the usable services.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

Note

- This method cannot be used from Singleton ORB. For details refer to "Notes" in "3.1.1 org omg.CORBA.ORB.init ()".
- When the IIOP Service (Java EE client) feature is used, the following cannot be used even if they are returned.
  - "RootPOA"
  - "POACurrent"
  - "PICurrent"
  - "CodecFactory"
  - "DynAnyFactory"

3.1.3 org.omg.CORBA.ORB.resolve_initial_references()

Name

org.omg.CORBA.ORB.resolve_initial_references

Format

```
public  org.omg.CORBA.Object resolve_initial_references(
        java.lang.String object_name )
throws org.omg.CORBA.ORBPackage.InvalidName;
```

Description

Returns an initial service object reference for the initial service specified by object_name.

For details of object references, refer to the Distributed Application Development Guide (CORBA Service Edition).

Parameters

object_name

A string object that shows the name of the initial service. You can use the following typical initial services:

"InterfaceRepository"

"NameService"

"RootPOA"

"FJ_LoadBalancingOption" (*1)

*1 This is not valid for or Windows Server(R) x64 Editions and Linux for Intel64.

Return Values

Normal termination: Returns an object reference for the specified name.

If the specified object does not exist, a NULL object is returned. If NULL is returned with "FJ_LoadBalancingOption" specified as the object name, the load balance function environment is not configured.

If NULL is returned with "FJ_LoadBalancingOption" specified as the object name, the load balance function environment is not configured.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.ORBPackage.InvalidName

  Invalid value specified in the initial service name

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

Note

- This method cannot be used from Singleton ORB. For details refer to "Notes" in "3.1.1 org omg.CORBA.ORB.init ()".
- When the IIOP Service (Java EE client) feature is used, only object references returned when "NameService" was specified can be used.

3.1.4 org.omg.CORBA.ORB.resolve_initial_references_remote()

Name

org.omg.CORBA.ORB.resolve_initial_references_remote

Format

```
public  org.omg.CORBA.Object resolve_initial_references_remote(
        java.lang.String identifier,
        java.lang.String[] m)
throws org.omg.CORBA.ORBPackage.InvalidName;
```

Description

Returns an object reference for the object specified by identifier. This function searches the initial_services of the host defined at m in URL format to retrieve the object reference.

Multiple URLs can be specified at m. If so, the hosts are searched in order of specification, and the search is discontinued as soon as the object reference is found.

The URL specification format is as follows:

iiop://<address>[:<port>]

Specify either the host name, DNS name, or the IP address at <address>. The address specification can not be omitted. For <port>, specify the ORB port number at the connection destination.

Parameters

identifier

A string object indicating the initial service name. The following objects can be used:

"InterfaceRepository"

"NameService"

m

List of hosts to be searched

Return Values

Normal termination: Returns the object reference.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.ORBPackage.InvalidName

  Object specified by identifier not found

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Note

- This method cannot be used from Singleton ORB. For details refer to "Notes" in "3.1.1 org omg.CORBA.ORB.init ()".
- This method cannot be used for ORB products that are not part of this product, such as GS.
- This method is unique to Fujitsu. It cannot be used in IIOP Service (Java EE client).

Example

Before calling this method, cast the ORB class instance fetched by the ORB.init() method in the com.fujitsu.ObjectDirector.CORBA.ORB type. Do not specify the com.fujitsu.ObjectDirector.CORBA package in the import declaration.

```
java.lang.String[] m = new String[2];

m[0] = new String("iiop://remotehost01:8002");
m[1] = new String("iiop://remotehost02:8002");

org.omg.CORBA.Object Obj =
    Orb.resolve_initial_references_remote( "NameService", m );
```

3.1.5 org.omg.CORBA.ORB.object_to_string()

Name

org.omg.CORBA.ORB.object_to_string

Format

```
public  java.lang.String object_to_string( org.omg.CORBA.Object obj )
```

Description

Converts the object reference of an object specified in obj to a String object.

Parameters

obj

The object reference that you want to convert to a String object.

Return Values

Normal termination: Returns a string object that shows the object reference.

Abnormal termination: Issues the following exception:

- java.lang.SecurityException

  This was used from Singleton ORB.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

Note

This method cannot be used from Singleton ORB. For details refer to "Notes" in "3.1.1 org omg.CORBA.ORB.init ()".

3.1.6 org.omg.CORBA.ORB.string_to_object()

Name

org.omg.CORBA.ORB.string_to_object

Format

```
public  org.omg.CORBA.Object string_to_object( java.lang.String str )
```

Description

object_to_string() converts the String object you generated to an object reference. The character strings for all of the formats shown in "corbaloc URL Schema" in the Distributed Application Development Guide (CORBA Service Edition) can be converted.

Parameters

str

The String object that you want to convert to an object reference.

Return Values

Normal termination: Returns an org.omg.CORBA.Object object that contains the converted object reference.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.BAD_PARAM

  There is an error in the format of the specified URL schema.

- java.lang.SecurityException

  This was used from Singleton ORB.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

Note

This method cannot be used from Singleton ORB. For details refer to "Notes" in "org.omg.CORBA.ORB.init()".

Example

- Example for URL Schema.

```
import org.omg.CORBA.*;
import org.omg.CosNaming.*;
import test1.*;

public static void main( String args[] )
{
    ORB  Orb;
    org.omg.CORBA.Object  obj_ns, obj_sv, obj_sv2;
    NamingContextExt Cos;
    intf1    target1, target2;

    Orb = ORB.init( args, null );

    String NCid   = new String( "test1::intf1" );
    String NCkind = new String( "" );
    NameComponent nc =  new NameComponent( NCid, NCkind );
    NameComponent NCo[] = { nc };

    obj_ns = Orb.string_to_object( "corbaloc::nshost:8002/NameService" );

    Cos  = NamingContextExtHelper.narrow( obj_ns );
    obj_sv = Cos.resolve( NCo );
    target1 = intf1Helper.narrow( obj_sv );
    target1.add( 1, 2 );

    obj_sv2 = Orb.string_to_object( "corbaname::nshost:8002/NameService#test1::intf1" );

    target2 = intf1Helper.narrow( obj_sv2 );
    target2.add( 1, 2 );
}
```

3.1.7 org.omg.CORBA.ORB.create_list()

Name

org.omg.CORBA.ORB.create_list

Format

```
public org.omg.CORBA.NVList create_list(int count);
```

Description

Generates an NVList object. This list object can store the number of item parameters (NamedValue objects) specified by count. Use the NVList method org.omg.CORBA.NVList.add_item to add parameters to the list object.

Parameters

count

The number of parameters to be stored in the list object.

Return Values

Normal termination: Returns the newly generated NVList object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.1.8 org.omg.CORBA.ORB.create_operation_list()

Name

org.omg.CORBA.ORB.create_operation_list

Format

```
(1) public org.omg.CORBA.NVList
    create_operation_list(org.omg.CORBA.Object oper);
(2) public org.omg.CORBA.NVList
    create_operation_list(org.omg.CORBA.OperationDef oper);
```

Description

Generates a list object (NVList object) initialized by parameter information that is related to the operation specified by oper. The parameter order in the list will be the same as the one in the IDL operation definition.

Parameters

(1) oper

CORBA object with Operation information used to generate the list object

(2) oper

OperationDef object used to generate the list object

Return Values

Normal termination: Returns the newly created NVList object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Note

When "(2) create_operation_list" is used, cast the type org.omg.CORBA.ORB to the com.fujitsu.ObjectDirector.CORBA.ORB type before calling this method.

Do not specify the com.fujitsu.ObjectDirector.CORBA package using the import declaration.

The following is an example of the code.

```
 org.omg.CORBA.ORB orb=org.omg.CORBA.ORB.init();
                   :
 org.omg.CORBA.NVList nvlist =
  ((com.fujitsu.ObjectDirector.CORBA.ORB)orb).create_operation_list( oper );
```

3.1.9 org.omg.CORBA.ORB.create_named_value()

Name

org.omg.CORBA.ORB.create_named_value

Format

```
public org.omg.CORBA.ORB.NamedValue
create_named_value(java.lang.String name,
                   org.omg.CORBA.ORB.Any value,
                   int flags);
```

Description

Generates a NamedValue object. The NamedValue object uses dynamic invocation to store parameter and return values.

Parameters

name

Parameter name (for a return value select null)

value

parameter

flags

Direction to pass parameters Select the following values:

- org.omg.CORBA.ARG_IN.value

  in parameter (input only)

- org.omg.CORBA.ARG_OUT.value

  out parameter (output only)

- org.omg.CORBA.ARG_INOUT.value

  inout parameter (input and output)

Return Values

Normal termination: Returns the NamedValue object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.1.10 org.omg.CORBA.ORB.create_exception_list()

Name

org.omg.CORBA.ORB.create_exception_list

Format

```
public org.omg.CORBA.ExceptionList create_exception_list()
```

Description

Generates and returns an ExceptionList object.

Return Values

Normal termination: Returns the ExceptionList object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.1.11 org.omg.CORBA.ORB.create_context_list()

Name

org.omg.CORBA.ORB.create_context_list

Format

```
public org.omg.CORBA.ContextList create_context_list()
```

Description

Generates and returns a ContextList object.

Return Values

Normal termination: Returns the ContextList object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.1.12 org.omg.CORBA.ORB.get_default_context()

Name

org.omg.CORBA.ORB.get_default_context

Format

```
public  org.omg.CORBA.Context get_default_context()
```

Description

Returns the default Context object's object reference.

Return Values

Normal termination: Returns the default Context object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Note

This method cannot be used from Singleton ORB. For details refer to "Notes" in "3.1.1 org omg.CORBA.ORB.init ()".

3.1.13 org.omg.CORBA.ORB.create_environment()

Name

org.omg.CORBA.ORB.create_environment

Format

```
public org.omg.CORBA.Environment create_environment()
```

Description

Generates and returns an Environment object. An environment object uses a Request object to receive exception information issued when an operation is invoked.

Return Values

Normal termination: Returns the Environment object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.1.14 org.omg.CORBA.ORB.send_multiple_requests_oneway()

Name

org.omg.CORBA.ORB.send_multiple_requests_oneway

Format

```
public void send_multiple_requests_oneway(org.omg.CORBA.Request[] req)
```

Description

Sends multiple requests in parallel. This method returns to the caller without waiting for a response from the server application.

Parameters

req

The request object arrays ordered by the send request

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Note

This method cannot be used from Singleton ORB. For details refer to "Notes" in "3.1.1 org omg.CORBA.ORB.init ()".

3.1.15 org.omg.CORBA.ORB.send_multiple_requests_deferred()

Name

org.omg.CORBA.ORB.send_multiple_requests_deferred

Format

```
public void send_multiple_requests_deferred(org.omg.CORBA.Request[] req)
```

Description

Sends multiple requests in parallel. This method waits for a response from the server application; then it returns to the caller.

Parameters

req

The request object arrays ordered by the send request

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Note

This method cannot be used from Singleton ORB. For details refer to "Notes" in "3.1.1 org omg.CORBA.ORB.init ()".

3.1.16 org.omg.CORBA.ORB.poll_next_response()

Name

org.omg.CORBA.ORB.poll_next_response

Format

```
public boolean poll_next_response()
```

Description

Tells if it received a termination notification for a request that was sent with send_multiple_requests_deferred().

Return Values

Normal termination: The following values are returned:

- true

  Notification received

- false

  No notification received

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Note

This method cannot be used from Singleton ORB. For details refer to "Notes" in "3.1.1 org omg.CORBA.ORB.init ()".

3.1.17 org.omg.CORBA.ORB.get_next_response()

Name

org.omg.CORBA.ORB.get_next_response

Format

```
public org.omg.CORBA.Request get_next_response()
throws WrongTransaction
```

Description

Returns the request objects transmitted by the send_multiple_requests_deferred() method, processed by the server side, but not completed in the client side. The sequential relationship of the finished requests is not guaranteed.

Return Values

Normal termination: Returns a Request object for the request following the one just terminated.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Note

This method cannot be used from Singleton ORB. For details refer to "Notes" in "3.1.1 org omg.CORBA.ORB.init ()".

3.1.18 org.omg.CORBA.ORB.get_primitive_tc()

Name

org.omg.CORBA.ORB.get_primitive_tc

Format

```
public org.omg.CORBA.TypeCode
get_primitive_tc(public org.omg.CORBA.TCKind tcKind);
```

Description

Returns a TypeCode object that contains the specified primitive attribute information (IDL primitive data type information).

Parameters

tcKind

Attribute information. You can specify the following TCKind object values.

org.omg.CORBA.TCKind.tk_null

org.omg.CORBA.TCKind.tk_void

org.omg.CORBA.TCKind.tk_short

org.omg.CORBA.TCKind.tk_long

org.omg.CORBA.TCKind.tk_ushort

org.omg.CORBA.TCKind.tk_ulong

org.omg.CORBA.TCKind.tk_float

org.omg.CORBA.TCKind.tk_double

org.omg.CORBA.TCKind.tk_boolean

org.omg.CORBA.TCKind.tk_char

org.omg.CORBA.TCKind.tk_octet

org.omg.CORBA.TCKind.tk_any

org.omg.CORBA.TCKind.tk_TypeCode

org.omg.CORBA.TCKind.tk_Principal

org.omg.CORBA.TCKind.tk_objref (Note)

org.omg.CORBA.TCKind.tk_enum (Note)

org.omg.CORBA.TCKind.tk_string

org.omg.CORBA.TCKind.tk_longlong

org.omg.CORBA.TCKind.tk_ulonglong

org.omg.CORBA.TCKind.tk_longdouble

org.omg.CORBA.TCKind.tk_wchar

org.omg.CORBA.TCKind.tk_wstring

Return Values

Normal termination: Returns a TypeCode object that contains the selected attribute information.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

Note

When the IIOP Service (Java EE client) feature is used, note the following:

- When "org.omg.CORBA.TCKind.tk_objref" is specified for the TCKind object, a TypeCode object with an "IDL:omg.org/CORBA/Object:1.0" ID is returned.
- "org.omg.CORBA.TCKind.tk_enum" cannot be specified for the TCKind object.

3.1.19 org.omg.CORBA.ORB.create_any()

Name

org.omg.CORBA.ORB.create_any

Format

```
public org.omg.CORBA.Any
create_any();
```

Description

Generates an Any type object.

Return Values

Normal termination: Returns the generated Any type object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.1.20 org.omg.CORBA.ORB.run()

Name

org.omg.CORBA.ORB.run

Format

```
public void run();
```

Description

This method is called at the end of the application. This method notifies ORB to be ready to receive requests.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Notes

- This method invokes to that the server application stands by for receiving requests. For ObjectDirector, this method do noting because invoking org.omg.PortableServer.POAManager.activate() before invoking this method causes that the server application stands by for receiving requests. But it is better to invoke this method for portability of application.
- This method cannot be used from Singleton ORB. For details refer to "Notes" in "3.1.1 org omg.CORBA.ORB.init ()".

3.1.21 org.omg.CORBA.ORB.destroy()

Name

org.omg.CORBA.ORB.destroy

Format

```
public void destroy() ;
```

Description

Explicitly releases the communication resource and internal thread/internal instance owned by the ORB instance.

Return Value

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

Example

Casts the ORB class instance fetched by the ORB.init() method in the com.fujitsu.ObjectDirector.CORBA.ORB mold and then calls this method. Do not specify the com.fujitsu.ObjectDirector.CORBA package in the import declaration.

The following is an example of the coding.

- When Portable-ORB is used

```
org.omg.CORBA.ORB orb=org.omg.CORBA.ORB.init();
          :
  ((com.fujitsu.ObjectDirector.CORBA.ORB)orb).destroy();
```

- When the IIOP Service (Java EE client) is used

```
org.omg.CORBA.ORB orb=org.omg.CORBA.ORB.init();
          :
  orb.destroy();
```

Notes

- The ORB instance cannot be used after this method has been issued. Initialize ORB if the ORB functions are required again.
- System operation is not guaranteed if this method is used for the ORB instance once this method has already been used.
- Use this method when terminating applications.
- This method supports Portable-ORB and the IIOP Service (Java EE client). No pre-installed Java clients or Java servers can be used.
- This method cannot be used from Singleton ORB. For details refer to "Notes" in "3.1.1 org omg.CORBA.ORB.init ()".

3.1.22 org.omg.CORBA.ORB.create_array_tc()

Name

org.omg.CORBA.ORB.create_array_tc

Format

```
public org.omg.CORBA.TypeCode create_array_tc(
    int length,
    org.omg.CORBA.TypeCode element_type)
```

Description

This method creates and returns an IDL array type TypeCode object.

Parameters

length

Array length

element_type

TypeCode object of the type of element stored in the array

Return Values

Normal termination: Returns the array type TypeCode object.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.1.23 org.omg.CORBA.ORB.create_enum_tc()

Name

org.omg.CORBA.ORB.create_enum_tc

Format

```
public org.omg.CORBA.TypeCode create_enum_tc(
    java.lang.String id,
    java.lang.String name,
    java.lang.String[] members)
```

Description

This method creates and returns an IDL enumeration type TypeCode object.

Parameters

id

Repository ID of enumeration type

name

Enumeration type name

members

Array to describe enumeration type members

Return Values

Normal termination: Returns the enumeration type TypeCode object.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.1.24 org.omg.CORBA.ORB.create_exception_tc()

Name

org.omg.CORBA.ORB.create_exception_tc

Format

```
public org.omg.CORBA.TypeCode create_exception_tc(
    java.lang.String id,
    java.lang.String name,
    org.omg.CORBA.StructMember[] members)
```

Description

This method creates and returns an IDL exception TypeCode object.

Parameters

id

Exception repository ID

name

Exception name

members

Array to store exception members

Return Values

Normal termination: Returns the exception TypeCode object.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.1.25 org.omg.CORBA.ORB.create_sequence_tc()

Name

org.omg.CORBA.ORB.create_sequence_tc

Format

```
public org.omg.CORBA.TypeCode create_sequence_tc(
    int bound,
    org.omg.CORBA.TypeCode element_type)
```

Description

This method creates and returns an IDL sequence type TypeCode object.

Parameters

bound

Maximum sequence length

element_type

TypeCode object of the type of element stored in the sequence

Return Values

Normal termination: Returns the sequence type TypeCode object.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.1.26 org.omg.CORBA.ORB.create_string_tc()

Name

org.omg.CORBA.ORB.create_string_tc

Format

```
public org.omg.CORBA.TypeCode create_string_tc(int bound)
```

Description

This method creates and returns an IDL character type TypeCode object.

Parameters

bound

Maximum string length

Return Values

Normal termination: Returns the character type TypeCode object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.1.27 org.omg.CORBA.ORB.create_struct_tc()

Name

org.omg.CORBA.ORB.create_struct_tc

Format

```
public org.omg.CORBA.TypeCode create_struct_tc(
    java.lang.String id,
    java.lang.String name,
    org.omg.CORBA.StructMember[] members)
```

Description

This method creates and returns an IDL structure type TypeCode object.

Parameters

id

Structure repository ID

name

Structure name

members

Array to store structure members

Return Values

Normal termination: Returns the structure type TypeCode object.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.1.28 org.omg.CORBA.ORB.create_union_tc()

Name

org.omg.CORBA.ORB.create_union_tc

Format

```
public org.omg.CORBA.TypeCode create_union_tc(
    java.lang.String id,
    java.lang.String name,
    org.omg.CORBA.TypeCode discriminator_type,
    org.omg.CORBA.UnionMember[] members)
```

Description

This method creates and returns an IDL union type TypeCode object.

Parameters

id

Union repository ID

name

Union name

discriminator_type

TypeCode to represent a union discriminator type

members

Array to store union members

Return Values

Normal termination: Returns the union type TypeCode object.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.1.29 org.omg.CORBA.ORB.create_wstring_tc()

Name

org.omg.CORBA.ORB.create_wstring_tc

Format

```
public org.omg.CORBA.TypeCode create_wstring_tc(int bound)
```

Description

This method creates and returns an IDL wide string type TypeCode object.

Parameters

bound

Maximum wide string length

Return Values

Normal termination: Returns the wide string type TypeCode object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.2 Object Interface

This section describes the Object interface that controls object references.

The APIs which are supported by the CORBA Service (ObjectDirector) and IIOP Service (Java EE client) in the Object class are shown below.

API	CORBA Service (ObjectDirector)	IIOP Service (Java EE client)
org.omg.CORBA.Object._is_a()	Y	Y
org.omg.CORBA.Object._is_equivalent()	Y	Y
org.omg.CORBA.Object._non_existent()	Y	Y
org.omg.CORBA.Object._hash()	Y	Y
org.omg.CORBA.Object._duplicate()	Y	Y
org.omg.CORBA.Object._release()	Y	Y
org.omg.CORBA.Object._get_interface_def()	Y	X
org.omg.CORBA.Object._request()	Y	X
org.omg.CORBA.Object._create_request()	Y	X

```
// Java
package org.omg.CORBA;
public interface Object
{
    boolean  _is_a(java.lang.String Identifier);
    boolean  _is_equivalent(java.lang.Object that);
    boolean  _non_existent();
    int  _hash(int maximum);
    org.omg.CORBA.Object  _duplicate();
    void  _release();
    public org.omg.CORBA.ImplementationDef  _get_implementation();
    public org.omg.CORBA.Object  _get_interface_def();
    org.omg.CORBA.Request  _request(java.lang.String s);
    org.omg.CORBA.Request  _create_request(
        org.omg.CORBA.Context ctx,
        java.lang.String operation,
        org.omg.CORBA.NVList arg_list,
        NamedValue result);
    org.omg.CORBA.Request  _create_request(
        org.omg.CORBA.Context ctx,
```

```
        java.lang.String operation,
        org.omg.CORBA.NVList arg_list,
        org.omg.CORBA.NamedValue result,
        org.omg.CORBA.ExceptionList exclist,
        org.omg.CORBA.ContextList ctxlist);
}
```

3.2.1 org.omg.CORBA.Object._is_a()

Name

org.omg.CORBA.Object._is_a

Format

```
boolean _is_a( java.lang.String Identifier );
```

Description

Checks whether or not an object is an instance of the repository ID's implementation class.

Parameters

Identifier

The repository ID you want to check: for example, "IDL:Module_A/Intf_A:1.0".

Return Values

Normal termination: Returns the following values:

- true

  The object is an instance of the specified repository ID's implementation class.

- false

  It is something else.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.2.2 org.omg.CORBA.Object._is_equivalent()

Name

org.omg.CORBA.Object._is_equivalent

Format

```
boolean _is_equivalent( java.lang.Object that );
```

Description

Checks whether or not the object references of this and a selected object are the same.

Parameters

that

The object you want to check.

Return Values

Normal termination: Returns the following values:

- true

  The two objects have the same object reference.

- false

  It is something else.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.2.3 org.omg.CORBA.Object._non_existent()

Name

org.omg.CORBA.Object._non_existent

Format

```
public boolean _non_existent();
```

Description

Checks whether or not an object's reference has been removed.

Return Values

Normal termination: Returns the following values:

- true

  ORB does not recognize the object.

- false

  It is something else.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.2.4 org.omg.CORBA.Object._hash()

Name

org.omg.CORBA.Object._hash

Format

```
public int _hash(int maximum);
```

Description

Returns the hash value associated with an object's object reference.

Parameters

maximum

Maximum hash value (greater than 0)

Return Values

Normal termination: Returns a hash value for object identification.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.2.5 org.omg.CORBA.Object._duplicate()

Name

org.omg.CORBA.Object._duplicate

Format

```
public org.omg.CORBA.Object _duplicate();
```

Description

Duplicates an object reference and returns the object reference of the duplicated object.

Return Values

Normal termination: Returns the duplicated object's object reference.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.2.6 orb.omg.CORBA.Object._release()

Name

orb.omg.CORBA.Object._release

Format

```
public void _release()
```

Description

Releases the object reference that is currently in use.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.2.7 org.omg.CORBA.Object._get_interface_def()

Name

org.omg.CORBA.Object._get_interface_def

Format

```
public org.omg.CORBA.Object getinterface_def();
```

Description

Returns the object reference for the Interface Repository that manages the interface information for the target org.omg.CORBA.Object object.

Return Values

Normal termination: Returns the object reference for the Interface Repository. When this object reference is not found in the interface repository, NULL is returned.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.2.8 org.omg.CORBA.Object._request()

Name

org.omg.CORBA.Object._request

Format

```
public org.omg.CORBA.Request
_request(java.lang.String operation);
```

Description

Generates a Request object that contains the server application method name (operation name) that was specified by operation.

Parameters

operation

A string object that shows the server application method name.

Return Values

Normal termination: Returns the Request object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.2.9 org.omg.CORBA.Object._create_request()

Name

org.omg.CORBA.Object._create_request

Format

```
(1) public org.omg.CORBA.Request _create_request(
           org.omg.CORBA.Context ctx,
           java.lang.String operation,
           org.omg.CORBA.NVList arg_list,
           org.omg.CORBA.NamedValue result );
(2) public org.omg.CORBA.Request _create_request(
           org.omg.CORBA.Context ctx,
           java.lang.String operation,
           org.omg.CORBA.NVList arg_list,
           org.omg.CORBA.NamedValue result,
           org.omg.CORBA.ExceptionList exceptions,
           org.omg.CORBA.ContextList contexts );
```

Description

Generates a Request object initialized with the specified arguments.

Parameters

(1)(2) Both are the same

ctx

The request context (Context object) that is passed to the server application.

operation

Server application method name (String object).

arg_list

An NVList object that contains parameter information to be passed to the server application method.

result

A NamedValue object that stores server application method invocation return values.

(2)

exceptions

A list of exceptions thrown by the server application method (ExceptionList object).

contexts

The request context list (ContextList object) that is passed to the server application.

Return Values

Normal termination: Returns the Request object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.3 NamedValue Class

This section explains the NamedValue class used by the dynamic invocation interface.

Note

The NamedValue class is not supported in the IIOP Service (Java EE client).

```
// Java
package org.omg.CORBA;
public  abstract class  NamedValue
{
        public abstract java.lang.String  name();
        public abstract org.omg.CORBA.Any  value();
        public abstract int  flags();
}
```

3.3.1 org.omg.CORBA.NamedValue.name()

Name

org.omg.CORBA.NamedValue.name

Format

```
public java.lang.String name();
```

Description

Returns the name of the parameter information (NamedValue object) that was added to the NVList, either by org.omg.CORBA.NVList.add_item(), or by org.omg.CORBA.NVList.add_value().

Return Values

Normal termination: Returns the parameter information name (java.lang.String object).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.3.2 org.omg.CORBA.NamedValue.value()

Name

org.omg.CORBA.NamedValue.value

Format

```
public abstract org.omg.CORBA.Any  value();
```

Description

Returns the value (Any type) of the parameter information (NamedValue object) that was added to the NVList object by org.omg.CORBA.NVList.add_item().

Return Values

Normal termination: Returns the parameter information value (Any type object).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.3.3 org.omg.CORBA.NamedValue.flags()

Name

org.omg.CORBA.NamedValue.flags

Format

```
public int flags()
```

Description

Returns a flag (int type) for the parameter information (NamedValue object) that was added to the NVList either by org.omg.CORBA.NVList.add(), org.omg.CORBA.NVList.add_item(), or org.omg.CORBA.NVList.add_value().

Return Values

Normal termination: Returns a flag for parameter information (NamedValue object).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.4 NVList Class

This section explains the NVList class used by the dynamic invocation interface.

Note

The NVList class is not supported in the IIOP Service (Java EE client).

```
// Java
package org.omg.CORBA;
public abstract class  NVList
{
   public abstract  int count();
   public abstract NamedValue  add(int flags);
   public abstract NamedValue  add_item(java.lang.String item_name, int flags);
```

```
   public abstract NamedValue  add_value(java.lang.String item_name, Any val, int
   flags);
  public abstract NamedValue  item(int index)
          throws org.omg.CORBA.Bounds;
  public abstract void  remove(int index)
          throws org.omg.CORBA.Bounds;
}
```

3.4.1 org.omg.CORBA.NVList. count()

Name

org.omg.CORBA.NVList.count

Format

```
public int count();
```

Description

Returns the number of parameter information (NamedValue objects) specified in the NVList object.

Return Values

Normal termination: Returns the number of NamedValue objects specified in the NVList object. Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.4.2 org.omg.CORBA.NVList. add()

Name

org.omg.CORBA.NVList.add

Format

```
public org.omg.CORBA.NamedValue add(int flags)
```

Description

Generates parameter information (NamedValue object) and adds it to the NVList object. It appends it to any previously stored parameter information.

Parameters

flags

Direction to pass parameters Select the following values:

org.omg.CORBA.ARG_IN.value

in parameter (input only)

org.omg.CORBA.ARG_OUT.value

out parameter (output only)

org.omg.CORBA.ARG_INOUT.value

inout parameter (input and output)

Return Values

Normal termination: Returns the newly generated NamedValue object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.4.3 org.omg.CORBA.NVList.add_item()

Name

org.omg.CORBA.NVList.add_item

Format

```
public org.omg.CORBA.NamedValue
add_item(String item_name,int flags);
```

Description

Generates parameter information (NamedValue object) that contains the specified name and flag, and adds it to the NVList object. It appends it to any previously stored parameter information.

Parameters

item_name

Parameter name

flags

Direction to pass parameters Select the following values:

org.omg.CORBA.ARG_IN.value

in parameter (input only)

org.omg.CORBA.ARG_OUT.value

out parameter (output only)

org.omg.CORBA.ARG_INOUT.value

inout parameter (input and output)

Return Values

Normal termination: Returns the newly generated NamedValue object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.4.4 org.omg.CORBA.NVList.add_value()

Name

org.omg.CORBA.NVList.add_value

Format

```
public org.omg.CORBA.NamedValue
add_value(String item_name,org.omg.CORBA.Any val,int flags);
```

Description

Generates parameter information (NamedValue object) that contains the specified name, value and flag, and adds it to the NVList object. It appends it to any previously appended parameter information.

Parameters

item_name

parameter name

val

parameter value to store in the object

flags

direction to pass parameters Select the following values:

org.omg.CORBA.ARG_IN.value

in parameter (input only)

org.omg.CORBA.ARG_OUT.value

out parameter (output only)

org.omg.CORBA.ARG_INOUT.value

inout parameter (input and output)

Return Values

Normal termination: Returns the newly generated NamedValue object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.4.5 org.omg.CORBA.NVList.item()

Name

org.omg.CORBA.NVList.item

Format

```
public org.omg.CORBA.NamedValue item(int index)
throws org.omg.CORBA.Bounds;
```

Description

Gets index-th parameter information from an NVList object (the first index is 0).

Parameters

index

The index of the parameter information you want to get

Return Values

Normal termination: Returns the index-th parameter information (NamedValue object).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.Bounds

  When not within the index range (number of parameters -1).

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.4.6 org.omg.CORBA.NVList.remove()

Name

org.omg.CORBA.NVList.remove

Format

```
public void remove(int index)
throws org.omg.CORBA.Bounds;
```

Description

Deletes the index-th parameter information from an NVList object (the first index is 0).

Parameters

index

the index of the parameter information you want to delete

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.Bounds

  When not within the index range (number of parameters -1).

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.5 Context Class

This section explains the Context class that controls Context objects.

Note

The Context class is not supported in the IIOP Service (Java EE client).

```
//Java
package org.omg.CORBA;
public abstract class Context
{
   public abstract java.lang.String  context_name();
```

```
   public abstract org.omg.CORBA.Context  parent();
   public abstract org.omg.CORBA.Context  create_child(java.lang.String child_ctx_name);
   public abstract void  set_one_value(java.lang.String propname, org.omg.CORBA.Any
   propvalue);
   public abstract void  set_values(org.omg.CORBA.NVList values);
   public abstract org.omg.CORBA.NVList get_ values(
      java.lang.String start_scpe, int op_flags, java.lang.String pattern);
      public abstract void  delete_values(java.lang.String propname);
}
```

3.5.1 org.omg.CORBA.Context.context_name()

Name

org.omg.CORBA.Context.context_name

Format

```
public  java.lang.String context_name();
```

Description

Returns the Context object's name.

Return Values

Normal termination: Returns the Context object's name.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.5.2 orb.omg.CORBA.Context.parent()

Name

org.omg.CORBA.Context.parent

Format

```
public org.omg.CORBA.Context parent();
```

Description

Returns a parent for a Context object.

Return Values

Normal termination: Returns a parent for a Context object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.5.3 org.omg.CORBA.Context.create_child()

Name

org.omg.CORBA.Context.create_child

Format

```
public org.omg.CORBA.Context
create_child( java.lang.String child_ctx_name );
```

Description

Generates a child context object with a specified name.

Parameters

child_ctx_name

The String object that specifies the new Context object's name

Return Values

Normal termination: Returns the newly created child context object with the selected name.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.5.4 orb.omg.CORBA.Context.set_one_value()

Name

org.omg.CORBA.Context.set_one_value

Format

```
public  void set_one_value( java.lang.String propname,
     org.omg.CORBA.Any propvalue );
```

Description

This method will set one attribute value for a Context object. Specify the property name with propname and the value with propvalue.

Parameters

propname

Property name

propvalue

Property value

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.5.5 org.omg.CORBA.Context.set_values()

Name

org.omg.CORBA.Context.set_values

Format

```
public  void set_values( org.omg.CORBA.NVList values );
```

Description

Sets multiple attribute values for a Context object. Set them with the values parameter.

Parameters

values

The NVList that contains property names and values

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.5.6 org.omg.CORBA.Context.get_values()

Name

org.omg.CORBA.Context.get_values

Format

```
public org.omg.CORBA.NVList get_values(
       java.lang.String start_scope,
       int op_flags,
       java.lang.String pattern );
```

Description

Acquires context attribute values for a Context object.

Parameters

start_scope

Specifies the context object level from which to begin searching for the attributes.

op_flags

You can specify the following flags.

A search that uses org.omg.CORBA.CTX_RESTRICT_SCOPE is limited by the scope set in start_scope, and by the context object.

pattern

Name of the property containing the value you want to get. If you use the wildcard "*", all matching attribute names and values will be returned.

Return Values

Normal termination: Returns context attribute values.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.5.7 org.omg.CORBA.Context.delete_values()

Name

org.omg.CORBA.Context.delete_values

Format

```
public  void delete_values( java.lang.String propname );
```

Description

Deletes the attribute value specified in propname from a context object. If the wildcard "*" is included in the String object propname, all matching NamedValue objects will be deleted. An exception will be issued if there are no matching properties.

Parameters

propname

Property name to delete If you use the wildcard "*", all matching attribute names and their values will be deleted.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.6 ContextList Class

This section explains the ContextList class that controls Context objects.

Note

The ContextList class is not supported in the IIOP Service (Java EE client).

```
//Java
package org.omg.CORBA;
public abstract class ContextList
{
    public abstract void  add(java.lang.String ctx);
    public abstract java.lang.String  item(int index)
       throws org.omg.CORBA.Bounds;
    public abstract void  remove(int index)
       throws org.omg.CORBA.Bounds;
    public abstract int  count();
}
```

3.6.1 org.omg.CORBA.ContextList.add()

Name

org.omg.CORBA.ContextList.add

Format

```
public  void add( java.lang.String ctx )
```

Description

Adds elements to a context object.

Parameters

ctx

the element to add

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.6.2 org.omg.CORBA.ContextList.item()

Name

org.omg.CORBA.ContextList.item

Format

```
public  String item( int index )
throws org.omg.CORBA.Bounds;
```

Description

Returns an element for the specified index.

Parameters

index

Index of the element to acquire. Values range from 0 to (number of elements -1).

Return Values

Normal termination: Returns an element for the specified index.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.Bounds

  When not within the index range (number of parameters -1).

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.6.3 org.omg.CORBA.ContextList.remove()

Name

org.omg.CORBA.ContextList.remove

Format

```
public  void remove( int index )
throws org.omg.CORBA.Bounds;
```

Description

Deletes the element from the specified index.

Parameters

Index

The element to delete. Values range from 0 to (number of elements -1).

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.Bounds

  When not within the index range (number of parameters -1).

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.6.4 org.omg.CORBA.ContextList.count()

Name

org.omg.CORBA.ContextList.count

Format

```
public  int count()
```

Description

Returns the number of elements contained in a ContextList object.

Return Values

Normal termination: Returns the number of elements contained in a ContextList object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7 Request Class

This section explains the Request class used by the dynamic invocation interface.

Note

The Request class is not supported in the IIOP Service (Java EE client).

```
//Java
package org.omg.CORBA;
public abstract class Request
{
    public abstract Object  target();
    public abstract java.lang.String  operation();
    public abstract org.omg.CORBA.NVList  arguments();
    public abstract org.omg.CORBA.NamedValue  result();
    public abstract org.omg.CORBA.Environment  env();
    public abstract org.omg.CORBA.ExceptionList  exceptions();
    public abstract org.omg.CORBA.ContextList  contexts();
    public abstract org.omg.CORBA.Context  ctx();
    public abstract void  ctx(Context c);
    public abstract org.omg.CORBA.Any  add_in_arg();
    public abstract org.omg.CORBA.Any  add_named_in_arg(java.lang.String name);
    public abstract org.omg.CORBA.Any  add_inout_arg();
    public abstract org.omg.CORBA.Any  add_named_inout_arg(java.lang.String name);
    public abstract org.omg.CORBA.Any  add_out_arg();
    public abstract org.omg.CORBA.Any  add_named_out_arg(java.lang.String name);
    public abstract void  set_return_type(org.omg.CORBA.TypeCode tc);
    public abstract org.omg.CORBA.Any  return_value();
    public abstract void  invoke();
    public abstract void  send_oneway();
    public abstract void  send_deferred();
    public abstract void  get_response()
       throws  org.omg.CORBA.WrongTransaction;
    public abstract boolean  poll_response();
}
```

3.7.1 org.omg.CORBA.Request.target()

Name

org.omg.CORBA.Request.target

Format

```
public org.omg.CORBA.Object target();
```

Description

Returns the Request object's object reference.

Return Values

Normal termination: Returns the object reference for the Request object.

Abnormal termination: Issues the following exception

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.2 org.omg.CORBA.Request.operation()

Name

org.omg.CORBA.Request.operation

Format

```
public java.lang.String operation();
```

Description

Retrieves the operation name that is set in the Request object.

Return Values

Normal termination: Returns the operation name (java.lang.String object).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.3 org.omg.CORBA.Request.arguments()

Name

org.omg.CORBA.Request.arguments

Format

```
public org.omg.CORBA.NVList arguments();
```

Description

Gets the NVList object (the object containing method call time parameters) that is specified by the Request object.

Return Values

Normal termination: Returns the NVList object specified by the Request object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.4 org.omg.CORBA.Request.result()

Name

org.omg.CORBA.Request.result

Format

```
public org.omg.CORBA.NamedValue result()
```

Description

Gets the NamedValue object (the object that holds method call results) that is specified by the Request object.

Return Values

Normal termination: Returns the NamedValue object specified by the Request object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.5 org.omg.CORBA.Request.env()

Name

org.omg.CORBA.Request.env

Format

```
public org.omg.CORBA.Environment env()
```

Description

Gets the Environment object that is in the Request object. The environment object holds exception information that is set as the result of method calls.

Return Values

Normal termination: Returns the Environment object that is in Request object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.6 org.omg.CORBA.Request.exceptions()

Name

org.omg.CORBA.Request.exceptions

Format

```
public org.omg.CORBA.ExceptionList exceptions()
```

Description

Gets the ExceptionList object that is specified by the Request object. This object holds, as a TypeCode object list, the exception types that are issued as the result of method calls.

Return Values

Normal termination: Returns the ExceptionList object specified by the Request object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.7 org.omg.CORBA.Request.contexts()

Name

org.omg.CORBA.Request.contexts

Format

```
public org.omg.CORBA.ContextList contexts()
```

Description

Gets the ContextList object that is specified by the Request object. This object holds a list of the Context objects to be passed to a server application.

Return Values

Normal termination: Returns the ContextList object that is in the Request object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.8 org.omg.CORBA.Request.ctx()

Name

org.omg.CORBA.Request.ctx

Format

```
(1)  public org.omg.CORBA.Context ctx()
(2)  public void ctx(org.omg.CORBA.Context c)
```

Description

(1) Gets the Context object that is specified by the Request object. This object holds, as parameter values, environment information, etc. that are to be passed to a server application.

(2) Sets Context objects that have been specified as c in the Request object.

Parameters

(1) None

(2)

c

The Context object to be specified by the Request object.

Return Values

(1) Normal termination: Returns the acquired Context object.

(2) Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.9 org.omg.CORBA.Request.add_in_arg()

Name

org.omg.CORBA.Request.add_in_arg

Format

```
public org.omg.CORBA.Any add_in_arg();
```

Description

Generates an in parameter and adds it to a Request object. The in parameter is returned as a return value. (Automatically generates a NamedValue object containing an ARG_IN flag, then adds it to the NVList object. Returns the NamedValue object's parameter value store field as a return value.)

Return Values

Normal termination: Returns the newly added in parameter (Any type object).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.10 org.omg.CORBA.Request.add_named_in_arg()

Name

org.omg.CORBA.Request. add_named_in_arg

Format

```
public org.omg.CORBA.Any add_named_in_arg(java.lang.String name);
```

Description

Generates an in parameter that contains the name specified by name and adds it to a Request object. The in parameter is returned as a return value. (Automatically generates a NamedValue object containing an ARG_IN flag, and adds it to the NVList object. Returns the NamedValue object's parameter value store field as a return value.)

Parameters

name

Name of the in parameter you want to add

Return Values

Normal termination: Returns the newly added in parameter (Any type object).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.11 org.omg.CORBA.Request.add_inout_arg()

Name

org.omg.CORBA.Request.add_inout_arg

Format

```
public org.omg.CORBA.Any add_inout_arg();
```

Description

Generates an inout parameter and adds it to a Request object. The inout parameter is returned as a return value. (Automatically generates a NamedValue object containing an ARG_INOUT flag, and adds it to the NVList object. Returns the NamedValue object's parameter value store field as a return value.)

Return Values

Normal termination: Returns the newly added inout parameter (Any type object).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.12 org.omg.CORBA.Request.add_named_inout_arg()

Name

org.omg.CORBA.Request.add_named_inout_arg

Format

```
public org.omg.CORBA.Any add_named_inout_arg(java.lang.String name);
```

Description

Generates an inout parameter that contains the name specified by name and adds it to a Request object. The inout parameter is returned as a return value. (Automatically generates a NamedValue object containing an ARG_INOUT flag, and adds it to the NVList object. Returns the NamedValue object's parameter value store field as a return value.)

Parameters

name

Name of the inout parameter you want to add

Return Values

Normal termination: Returns the newly added inout parameter (Any type object).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.13 org.omg.CORBA.Request.add_out_arg()

Name

org.omg.CORBA.Request.add_out_arg

Format

```
public org.omg.CORBA.Any add_out_arg();
```

Description

Generates an out parameter and adds it to a Request object. The out parameter is returned as a return value. (Automatically generates a NamedValue object containing an ARG_OUT flag, and adds it to the NVList object. Returns the NamedValue object's parameter store value field as a return value.)

Return Values

Normal termination: Returns the newly added out parameter (Any type object).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.14 org.omg.CORBA.Request.add_named_out_arg()

Name

org.omg.CORBA.Request. add_named_out_arg

Format

```
public org.omg.CORBA.Any add_named_out_arg(java.lang.String name);
```

Description

Generates an out parameter that contains the name specified by name and adds it to a Request object. The out parameter is returned as a return value. (Automatically generates a NamedValue object containing an ARG_OUT flag, and adds it to the NVList object. Returns the NamedValue object's parameter value store field as a return value.)

Parameters

name

Name of the out parameter you want to add

Return Values

Normal termination: Returns the newly added out parameter (Any type object).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.15 org.omg.CORBA.Request.set_return_type()

Name

org.omg.CORBA.Request.set_return_type

Format

```
public void set_return_type(org.omg.CORBA.TypeCode tc)
```

Description

Sets the server application method return value type. It is specified by the Request object.

Parameters

tc

The TypeCode object that shows the method's return value type

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.16 org.omg.CORBA.Request.return_value()

Name

org.omg.CORBA.Request.return_value

Format

```
public org.omg.CORBA.Any return_value()
```

Description

Returns the server application method return value as an ANY type object. It is specified by the Request object.

Return Values

Normal termination: Returns an Any type object that contains the method return values.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.17 org.omg.CORBA.Request.invoke()

Name

org.omg.CORBA.Request.invoke

Format

```
public void invoke();
```

Description

Invokes the server application's method. The method is specified by the Request object. Results are stored in the NamedValue object (the result parameter specified when org.omg.CORBA.Object_create_request() generates a Request object) specified by the Request object. You can get the NamedValue object that contains these results with the Request object's results() method. A caller will be forced to wait until processing for the server application is completed.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.18 org.omg.CORBA.Request.send_oneway()

Name

org.omg.CORBA.Request.send_oneway

Format

```
public void send_oneway();
```

Description

Invokes the server application's method. The method is specified by the Request object. Unlike org.omg.CORBA.Request.invoke(), this will return control to the caller without waiting for the server application to complete. There is no notification of operation termination. Use this method for IDL operation's defined as oneway.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.19 org.omg.CORBA.Request.send_deferred()

Name

org.omg.CORBA.Request.send_deferred

Format

```
public void send_deferred();
```

Description

Invokes the server application's method. The server method is specified by the Request object. Results are stored in the NamedValue object (the result parameter specified when org.omg.CORBA.Object_create_request() generates a Request object) specified by the Request object. You can get the NamedValue object that contains these results with the Request object's results() method.

Unlike org.omg.CORBA.Request.invoke(), this will return control to the caller without waiting for the server application to complete. When the operation is completed, org.omg.CORBA.Request.get_response() will notify.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.20 org.omg.CORBA.Request.get_response()

Name

org.omg.CORBA.Request.get_response

Format

```
public void get_response()
throws org.omg.CORBA.WrongTransaction;
```

Description

Gets the Request object's process results. After calling org.omg.CORBA.Request.send_deferred(), you can call it to get server application process results.

A caller will be forced to wait until processing for the server application is completed. After the server application is completed this method will return. Also return values for output parameters and methods associated with the Request object will become valid.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.7.21 org.omg.CORBA.Request.poll_response()

Name

org.omg.CORBA.Request.poll_response

Format

```
public boolean poll_response();
```

Description

When you generate a Request object, this will ask if the specified server application method has been completed. After calling org.omg.CORBA.Request.send_deferred(), you can call it to check if the server application method has been completed.

If the server application method is completed, return values for output parameters and methods associated with the Request object will become valid.

Unlike org.omg.CORBA.Request.get_response(), this method will return even while the server application method is running.

Return Values

Normal termination: Returns the following values:

- true

  Server application method has already completed.

- false

  Server application method has not yet completed.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.8 ServerRequest Class

This section explains the ServerRequest class used by the dynamic skeleton interface.

Note

The ServerRequest class is not supported in the IIOP Service (Java EE client).

```
//Java
package org.omg.CORBA;
public abstract class ServerRequest
{
    abstract public java.lang.String operation();
    abstract public void arguments(org.omg.CORBA.NVList params);
    abstract public void set_result(org.omg.CORBA.Any result);
    abstract public void set_exception(org.omg.CORBA.Any except);
    abstract public org.omg.CORBA.Context ctx();
}
```

3.8.1 org.omg.CORBA.ServerRequest.operation()

Name

org.omg.CORBA.ServerRequest.operation

Format

```
public java.lang.String operation();
```

Description

Gets the method name (operation name) from the ServerRequest object.

Return Values

Normal termination: Returns a String type object that shows the method name stored in the ServerRequest object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.8.2 org.omg.CORBA.ServerRequest.arguments()

Name

org.omg.CORBA.ServerRequest.arguments

Format

```
public void arguments(org.omg.CORBA.NVList params);
```

Description

Analyzes a ServerRequest object and gets parameter information for the NVList object specified by params.

Parameters

params

the NVList object that stores parameter information

Return Values

Normal termination: Sets parameter information in the NVList object that is specified by params.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.BAD_INV_ORDER

  When this method is issued more than once from the same ServerRequest.

- org.omg.CORBA.BAD_INV_ORDER

  When this method is issued from the same ServerRequest, after the set=exception() method has been issued.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.8.3 org.omg.CORBA.ServerRequest.set_result()

Name

org.omg.CORBA.ServerRequest.set_result

Format

```
public void set_result(org.omg.CORBA.Any result);
```

Description

Sets the server method return value (Any type object) that is specified with result in the ServerRequest object.

Parameters

result

The Any type object for storing server application method return values.

Return Values

Normal termination: Sets parameter information in the NVList object that is specified by params.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.BAD_INV_ORDER

  When this method is issued more than once from the same ServerRequest.

  When this method is issued from the same ServerRequest, before the argument() method

  When this method is issued from the same ServerRequest, after the set=exception() method

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.8.4 org.omg.CORBA.ServerRequest.set_exception()

Name

org.omg.CORBA.ServerRequest.set_exception

Format

```
public void set_exception(org.omg.CORBA.Any except);
```

Description

Specifies the user exceptions that the server application method will issue, and informs clients. (These user exceptions are specified in the IDL operation definition.)

Parameters

except

The Any type object that specifies which user exceptions to return to clients.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.8.5 org.omg.CORBA.ServerRequest.ctx()

Name

org.omg.CORBA.ServerRequest.ctx

Format

```
public org.omg.CORBA.Context ctx()
```

Description

Obtains a Context object from a ServerRequest object.

Return Values

Normal termination: Returns the Context object that is stored in the ServerRequest object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.BAD_INV_ORDER

  When this method is issued more than once from the same ServerRequest.

  When this method is issued from the same server ServerRequest, before the argument() method is issued.

  When this method is issued from the same ServerRequest, after the set_result() method has been issued.

  When this method is issued from the same ServerRequest, after the set=exception() method has been issued.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.9 TypeCode Class

This chapter will explain the TypeCode class that controls TypeCode.

The APIs which are supported by the CORBA Service (ObjectDirector) and IIOP Service (Java EE client) in the TypeCode class are shown below.

API	CORBA Service (ObjectDirector)	IIOP Service (Java EE client)
org.omg.CORBA.TypeCode.equal()	Y	Y
org.omg.CORBA.TypeCode.kind()	Y	Y
org.omg.CORBA.TypeCode.id()	Y	Y
org.omg.CORBA.TypeCode.name()	Y	Y
org.omg.CORBA.TypeCode.member_count()	Y	Y
org.omg.CORBA.TypeCode.member_name()	Y	Y
org.omg.CORBA.TypeCode.member_type()	Y	Y
org.omg.CORBA.TypeCode.member_label()	Y	X
org.omg.CORBA.TypeCode.discriminator_type()	Y	X
org.omg.CORBA.TypeCode.default_index()	Y	X
org.omg.CORBA.TypeCode.length()	Y	Y
org.omg.CORBA.TypeCode.content_type()	Y	Y

```
//Java
package org.omg.CORBA;
public abstract class TypeCode
{
    // for all TypeCode kinds
    public abstract boolean  equal(TypeCode tc);
    public abstract org.omg.CORBA .TCKind  kind();
    // for objref, struct, unio, enum, alias, and except
    public abstract java.lang.String  id()
        throws org.omg.CORBA.TypeCodePackage.BadKind;
    public abstract java.lang.String  name()
        throws org.omg.CORBA.TypeCodePackage.BadKind;
    // for struct, union, enum, and except
    public abstract int  member_count()
        throws org.omg.CORBA.TypeCodePackage.BadKind;
    public abstract java.lang.String  member_name(int index)
        throws org.omg.CORBA.TypeCodePackage.BadKind,
        org.omg.CORBA.TypeCodePackage.Bounds;
    // for struct, union, and except
    public abstract org.omg.CORBA .TypeCode  member_type(int index)
        throws org.omg.CORBA.TypeCodePackage.BadKind,
        org.omg.CORBA.TypeCodePackage.Bounds;
    // for union
    public abstract org.omg.CORBA .Any  member_label(int index)
        throws org.omg.CORBA.TypeCodePackage.BadKind,
        org.omg.CORBA.TypeCodePackage.Bounds;
    public abstract org.omg.CORBA .TypeCode  discriminator_type()
```

```
        throws org.omg.CORBA.TypeCodePackage.BadKind;
    public abstract int  default_index()
        throws org.omg.CORBA.TypeCodePackage.BadKind;
    // for string, sequence, and array
    public abstract int  length()
        throws org.omg.CORBA.TypeCodePackage.BadKind;
    public abstract org.omg.CORBA .TypeCode  content_type()
        throws org.omg.CORBA.TypeCodePackage.BadKind;
}
```

3.9.1 org.omg.CORBA.TypeCode.equal()

Name

org.omg.CORBA.TypeCode.equal

Format

```
public  boolean equal( org.omg.CORBA.TypeCode tc )
```

Description

Compares the TypeCode object with another, specified by tc, and returns the results.

Parameters

tc

the TypeCode object you want to compare with

Return Values

Normal termination: Returns one of the following:

- true

  If they are the same

- false

  If they differ.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.9.2 org.omg.CORBA.TypeCode.kind()

Name

org.omg.CORBA.TypeCode.kind

Format

```
public  org.omg.CORBA.TCKind kind()
```

Description

Returns a TypeCode object's attribute information (TCKind object). Following is an attribute information list.

org.omg.CORBA.TCKind.tk_null

org.omg.CORBA.TCKind.tk_void

org.omg.CORBA.TCKind.tk_short

org.omg.CORBA.TCKind.tk_long

org.omg.CORBA.TCKind.tk_ushort

org.omg.CORBA.TCKind.tk_ulong

org.omg.CORBA.TCKind.tk_float

org.omg.CORBA.TCKind.tk_double

org.omg.CORBA.TCKind.tk_boolean

org.omg.CORBA.TCKind.tk_char

org.omg.CORBA.TCKind.tk_octet

org.omg.CORBA.TCKind.tk_any

org.omg.CORBA.TCKind.tk_TypeCode

org.omg.CORBA.TCKind.tk_Principal

org.omg.CORBA.TCKind.tk_objref

org.omg.CORBA.TCKind.tk_struct

org.omg.CORBA.TCKind.tk_union

org.omg.CORBA.TCKind.tk_enum

org.omg.CORBA.TCKind.tk_string

org.omg.CORBA.TCKind.tk_sequence

org.omg.CORBA.TCKind.tk_array

org.omg.CORBA.TCKind.tk_alias

org.omg.CORBA.TCKind.tk_except

org.omg.CORBA.TCKind.tk_longlong

org.omg.CORBA.TCKind.tk_ulonglong

org.omg.CORBA.TCKind.tk_longdouble

org.omg.CORBA.TCKind.tk_wchar

org.omg.CORBA.TCKind.tk_wstring

org.omg.CORBA.TCKind.tk_fixed

Return Values

Normal termination: Returns the attribute information.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.9.3 org.omg.CORBA.TypeCode.id()

Name

org.omg.CORBA.TypeCode.id

Format

```
public  java.lang.String id()
throws org.omg.CORBA.TypeCodePackage.BadKind;
```

Description

Returns the TypeCode object's repository ID. This method works with TypeCode objects that contain the following attribute information.

org.omg.CORBA.TCKind.tk_objref

org.omg.CORBA.TCKind.tk_struct

org.omg.CORBA.TCKind.tk_union

org.omg.CORBA.TCKind.tk_enum

org.omg.CORBA.TCKind.tk_alias

org.omg.CORBA.TCKind.tk_except

Return Values

Normal termination: Returns the repository ID. If there is no repository ID it will return an empty string object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.TypeCodePackage.BadKind

  Invalid TypeCode attribute

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.9.4 org.omg.CORBA.TypeCode.name()

Name

org.omg.CORBA.TypeCode.name

Format

```
public  java.lang.String name()
throws org.omg.CORBA.TypeCodePackage.BadKind;
```

Description

Returns the object's name. This method works with TypeCode objects that contain the following attribute information.

org.omg.CORBA.TCKind.tk_objref

org.omg.CORBA.TCKind.tk_struct

org.omg.CORBA.TCKind.tk_union

org.omg.CORBA.TCKind.tk_enum

org.omg.CORBA.TCKind.tk_alias

org.omg.CORBA.TCKind.tk_except

Return Values

Normal termination: Returns member names.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.TypeCodePackage.BadKind

  Invalid TypeCode attribute

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.9.5 org.omg.CORBA.TypeCode.member_count()

Name

org.omg.CORBA.TypeCode.member_count

Format

```
public  int member_count()
throws org.omg.CORBA.TypeCodePackage.BadKind;
```

Description

Returns the number of TypeCode object members. This method works with TypeCodes that contain the following attribute information.

org.omg.CORBA.TCKind.tk_struct

org.omg.CORBA.TCKind.tk_union

org.omg.CORBA.TCKind.tk_enum

org.omg.CORBA.TCKind.tk_except

Return Values

Normal termination: Returns the number of TypeCode object member names.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.TypeCodePackage.BadKind

  Invalid TypeCode attribute

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.9.6 org.omg.CORBA.TypeCode.member_name()

Name

org.omg.CORBA.TypeCode.member_name

Format

```
public  java.lang.String member_name( int index )
throws org.omg.CORBA.BadKind, org.omg.CORBA.Bounds;
```

Description

Returns the member names specified by the TypeCode object's index. This method works with TypeCodes that contain the following attribute information.

org.omg.CORBA.TCKind.tk_struct

org.omg.CORBA.TCKind.tk_union

org.omg.CORBA.TCKind.tk_enum

org.omg.CORBA.TCKind.tk_except

Parameters

index

Member index from which you require names.

Return Values

Normal termination: Returns a String object that shows the member names identified by index.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.TypeCodePackage.BadKind

  Invalid TypeCode attribute

- org.omg.CORBA.TypeCodePackage.Bound

  Invalid index value (Index value is greater to or equal than the number of members.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.9.7 org.omg.CORBA.TypeCode.member_type()

Name

org.omg.CORBA.TypeCode.member_type

Format

```
public org.omg.CORBA.TypeCode member_type( int index )
throws org.omg.CORBA.TypeCodePackage.BadKind,
       org.omg.CORBA.TypeCodePackage.Bounds;
```

Description

Returns an object that shows the member types specified by the TypeCode object's index. This method works with TypeCodes that contain the following attribute information.

org.omg.CORBA.TCKind.tk_struct

org.omg.CORBA.TCKind.tk_union

org.omg.CORBA.TCKind.tk_except

Parameters

index

Member index from which you require type information

Return Values

Normal termination: Returns member TypeCode.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.TypeCodePackage.BadKind

  Invalid TypeCode attribute

- org.omg.CORBA.TypeCodePackage.Bound

  Invalid index value

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.9.8 org.omg.CORBA.TypeCode.member_label()

Name

org.omg.CORBA.TypeCode.member_label

Format

```
public org.omg.CORBA.Any member_label( int index )
throws org.omg.CORBA.TypeCodePackage.BadKind,
       org.omg.CORBA.TypeCodePackage.Bounds;
```

Description

Returns the member labels specified by the TypeCode object's index. This method works with TypeCodes that contain the following attribute information.

org.omg.CORBA.TCKind.tk_union

Parameters

index

Member index from which you require labels

Return Values

Normal termination: Returns member labels.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.TypeCodePackage.BadKind

  This method was invoked by a non-union TypeCode object.

- org.omg.CORBA.TypeCodePackage.Bound

  Invalid index value

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.9.9 org.omg.CORBA.TypeCode.discriminator_type()

Name

org.omg.CORBA.TypeCode.discriminator_type

Format

```
public org.omg.CORBA.TypeCode discriminator_type()
throws org.omg.CORBA.TypeCode.BadKind;
```

Description

Returns the TypeCode object's discrimination information definition TypeCode. This method works with TypeCode objects that contain the following attribute information.

org.omg.CORBA.TCKind.tk_union

Return Values

Normal termination: Returns the TypeCode of the discrimination information definition.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.TypeCodePackage.BadKind

  This method was invoked by a non-union TypeCode object.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.9.10 org.omg.CORBA.TypeCode.default_index()

Name

org.omg.CORBA.TypeCode.default_index

Format

```
public  int default_index()
throws org.omg.CORBA.TypeCodePackage.BadKind;
```

Description

Returns the default TypeCode object's default member index. This method works with TypeCodes that contain the following attribute information:

org.omg.CORBA.TCKind.tk_union

Return Values

Normal termination: the default member index is returned. If there are no default members it will return -1.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.TypeCodePackage.BadKind

  This method was invoked by a non-union TypeCode object.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.9.11 org.omg.CORBA.TypeCode.length()

Name

org.omg.CORBA.TypeCode.length

Format

```
public  int length()
throws org.omg.CORBA.TypeCodePackage.BadKind;
```

Description

Returns the TypeCode object's length: org.omg.CORBA.TCKind.tk_string returns string length; org.omg.CORBA.TCKind.tk_sequence and org.omg.CORBA.TCKind.tk_array return the number of elements. This method works with TypeCodes that contain the following attribute information.

org.omg.CORBA.TCKind.tk_string

org.omg.CORBA.TCKind.tk_sequence

org.omg.CORBA.TCKind.tk_array

Return Values

Normal termination: Returns the TypeCode object's length.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.TypeCodePackage.BadKind

  Invalid TypeCode attribute

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.9.12 org.omg.CORBA.TypeCode.content_type()

Name

org.omg.CORBA.TypeCode.content_type

Format

```
public org.omg.CORBA.TypeCode  content_type()
throws org.omg.CORBA.TypeCodePackage.BadKind;
```

Description

Returns the TypeCode object member's TypeCode object. This method works with TypeCodes that contain the following attribute information.

org.omg.CORBA.TCKind.tk_sequence

org.omg.CORBA.TCKind.tk_array

org.omg.CORBA.TCKind.tk_alias

Return Values

Normal termination: Returns the TypeCode object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.TypeCodePackage.BadKind

  Invalid TypeCode attribute

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.10 ExceptionList Class

This section explains the ExceptionList class. It controls exceptions issued as the result of operations defined by IDL. It does this by treating them as a TypeCode object list. (Used for the dynamic invocation interface.)

Note

The ExceptionList class is not supported in the IIOP Service (Java EE client).

```
//Java
package org.omg.CORBA;
public abstract class ExceptionList
{
    public abstract int  count();
    public abstract void  add(org.omg.CORBA.TypeCode exc);
    public abstract TypeCode  item(int index)
       throws org.omg.CORBA.Bounds;
    public abstract void  remove(int index)
       throws org.omg.CORBA.Bounds;
}
```

3.10.1 org.omg.CORBA.ExceptionList.count()

Name

org.omg.CORBA.ExceptionList.count

Format

```
public int count();
```

Description

Returns the number of TypeCode objects contained in the ExceptionList object. The ExceptionList object controls exceptions issued as the result of the execution of operations defined by IDL. To do this it treats them as a TypeCode object.

Return Values

Normal termination: Returns the number of TypeCode objects (int values) contained in the ExceptionList.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.10.2 org.omg.CORBA.ExceptionList.add()

Name

org.omg.CORBA.ExceptionList.add

Format

```
public void add(org.omg.CORBA.TypeCode exc);
```

Description

Adds the TypeCode object specified by exc to the ExceptionList object. The TypeCode object is one that shows the exceptions generated by IDL defined operations.

Parameters

exc

TypeCode object to add

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.10.3 org.omg.CORBA.ExceptionList.item()

Name

org.omg.CORBA.ExceptionList.item

Format

```
public org.omg.CORBA.TypeCode item(int index);
```

Description

Obtains the index-th TypeCode object from an ExceptionList object.

Parameters

index

Index of the TypeCode object you want to obtain (begins at 0).

Return Values

Normal termination: Returns the TypeCode object corresponding to the specified index.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.Bounds

  Not within the index range (number of TypeCode objects - 1).

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.10.4 org.omg.CORBA.ExceptionList.remove()

Name

org.omg.CORBA.ExceptionList.remove

Format

```
public void remove(int index);
```

Description

Deletes the index-th TypeCode object from an ExceptionList object.

Parameters

index

Index of the TypeCode object you want to delete (begins at 0).

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.Bounds

  Not within the index range (number of TypeCode objects - 1).

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.11 Environment Class

This section explains the Environment class that handles exception information.

Note

The Environment class is not supported in the IIOP Service (Java EE client).

```
//Java
package org.omg.CORBA;
public abstract class Environment
{
    abstract public void  exception(java.lang.Exception except);
    abstract public java.lang.Exception  exception();
    abstract public void  clear();
}
```

3.11.1 org.omg.CORBA.Environment.exception()

Name

org.omg.CORBA.Environment.exception

Format

```
(1)  public void exception( java.lang.Exception except );
(2)  Obtains the Exception class specified by the Environment object.
```

Description

(1) Sets exception classes that have been specified by the Environment object.

(2) Obtains exception classes that have been specified by the Environment object.

Parameters

(1)

except

the exception class to specify

(2) None

Return Values

(1) Normal termination: None

(2) Normal termination: Returns the exception class objects contained in the Environment object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.11.2 org.omg.CORBA.Environment.clear()

Name

org.omg.CORBA.Environment.clear

Format

```
public void clear();
```

Description

Removes exception information from an Environment object.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.12 Any Class

This section explains the Any class that handles all IDL data types.

The APIs which are supported by the CORBA Service (ObjectDirector) and IIOP Service (Java EE client) in the Any class are shown below.

API	CORBA Service (ObjectDirector)	IIOP Service (Java EE client)
org.omg.CORBA.Any.equal()	Y	Y
org.omg.CORBA.Any.type()	Y	Y
org.omg.CORBA.Any.insert_<type>()	Y	Y
org.omg.CORBA.Any.extract_<type>()	Y	Y
org.omg.CORBA.Any.create_output_stream()	Y	Y
org.omg.CORBA.Any.create_input_stream()	Y	Y
org.omg.CORBA.Any.read_value()	X	Y

```
//Java
package org.omg.CORBA;
abstract public class Any
{
    abstract public boolean  equal(org.omg.CORBA.Any a);
    abstract public org.omg.CORBA.TypeCode  type();
    abstract public void  type(org.omg.CORBA.TypeCode t);
    abstract public short  extract_short()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_short(short param);
    abstract public int  extract_long()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_long(int param);
    abstract public long  extract_longlong()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_longlong(long param);
    abstract public short  extract_ushort()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_ushort(short param);
    abstract public int  extract_ulong()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_ulong(int param);
    abstract public long  extract_ulonglong()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_ulonglong(long param);
    abstract public float  extract_float()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_float(float param);
    abstract public double  extract_double()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_double(double param);
```

```
    abstract public boolean  extract_boolean()
        throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_boolean(boolean param);
    abstract public char  extract_char()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_char(char param)
         throws org.omg.CORBA.DATA_CONVERSION;
    abstract public char  extract_wchar()
        throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_wchar(char param);
    abstract public byte  extract_octet()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_octet(byte param);
    abstract public org.omg.CORBA.Any  extract_any()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_any(org.omg.CORBA.Any param);
    abstract public org.omg.CORBA.Object  extract_Object()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_Object(org.omg.CORBA.Object param);
    abstract public void  insert_Object(org.omg.CORBA.Object param,org.omg.CORBA.TypeCode t)
         throws org.omg.CORBA.MARSHAL;
    abstract public java.lang.String  extract_string()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void  insert_string(java.lang.String param)
         throws org.omg.CORBA.DATA_CONVERSION, org.omg.CORBA.MARSHAL;
    abstract public java.lang.String  extract_wstring()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void insert_wstring(java.lang.String param)
         throws org.omg.CORBA.MARSHAL;
    abstract public org.omg.CORBA.TypeCode extract_TypeCode()
         throws org.omg.CORBA.BAD_OPERATION;
    abstract public void insert_TypeCode( org.omg.CORBA.TypeCode param);
    abstract public org.omg.CORBA.portable.OutputStream
        create_output_stream();
    abstract public org.omg.CORBA.portable.InputStream
        create_input_stream();
    abstract void read_value(org.omg.CORBA.portable.InputStream is, org.omg.CORBA.TypeCode t)
        throws org.omg.CORBA.MARSHAL;
}
```

3.12.1 org.omg.CORBA.Any.equal()

Name

org.omg.CORBA.Any.equal

Format

```
public boolean equal(org.omg.CORBA.Any rhs);
```

Description

Compares an Any object with one specified by rhs and returns the result.

Parameters

rhs

the Any object you want to compare with

Return Values

Normal termination: Returns the following values:

- true

  When the two objects are equal (TypeCode and values are equal).

- false

  When the two objects are not equal.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.12.2 org.omg.CORBA.Any.type()

Name

org.omg.CORBA.Any.type

Format

```
(1)  public  org.omg.CORBA.TypeCode type();
(2)  public  void type( org.omg.CORBA.TypeCode t );
```

Description

(1) Obtains the type code of an Any type object.

(2) Sets the type code of an Any type object.

Parameters

(1) None

(2)

t

TypeCode object you want to set.

Return Values

(1) Normal termination: Returns the type code of the Any type object.

(2) Normal termination: None

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.12.3 org.omg.CORBA.Any.insert_<type>()

Name

org.omg.CORBA.Any.insert_<type>

Format

```
public  void insert_short( short param );
public  void insert_long( int param );
public  void insert_longlong( long param );
public  void insert_ushort( short param );
public  void insert_ulong( int param );
public  void insert_ulonglong( long param );
public  void insert_float( float param );
public  void insert_double( double param );
public  void insert_boolean( boolean param );
public  void insert_char( char param )
        throws org.omg.CORBA.DATA_CONVERSION;
public  void insert_wchar( char param );
public  void insert_octet( byte param );
public  void insert_any( org.omg.CORBA.Any param );
public  void insert_Object( org.omg.CORBA.Object param );
public  void insert_Object( org.omg.CORBA.Object param, org.omg.CORBA.TypeCode t )
        throws  org.omg.CORBA.MARSHAL;
public  void insert_string( java.lang.String param )
        throws org.omg.CORBA.DATA_CONVERSION, org.omg.CORBA.MARSHAL;
public  void insert_wstring( java.lang.String param )
        throws org.omg.CORBA.MARSHAL;
public void insert_TypeCode(TypeCode param);
```

Description

Stores the values specified in the Any object.

Parameters

param

The values you want to store in the Any object.

t

The TypeCode object that shows the type of the values you will store (only the insert_Object() methods).

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.DATA_CONVERSION

  Values exceed the boundary.

- org.omg.CORBA.MARSHAL

  data write error

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.12.4 org.omg.CORBA.Any.extract_<type>()

Name

org.omg.CORBA.Any.extract_<type>

Format

```
public  short extract_short()
        throws org.omg.CORBA.BAD_OPERATION;
public  int extract_long()
        throws org.omg.CORBA.BAD_OPERATION;
public  long extract_longlong()
        throws org.omg.CORBA.BAD_OPERATION;
public  short extract_ushort()
        throws org.omg.CORBA.BAD_OPERATION;
public  int extract_ulong()
        throws org.omg.CORBA.BAD_OPERATION;
public  long extract_ulonglong()
        throws org.omg.CORBA.BAD_OPERATION;
public  float extract_float()
        throws org.omg.CORBA.BAD_OPERATION;
public  double extract_double()
        throws org.omg.CORBA.BAD_OPERATION;
public  boolean extract_boolean()
        throws org.omg.CORBA.BAD_OPERATION;
public  char extract_char()
        throws org.omg.CORBA.BAD_OPERATION;
public  char extract_wchar()
        throws org.omg.CORBA.BAD_OPERATION;
public  byte extract_octet()
        throws org.omg.CORBA.BAD_OPERATION;
public  org.omg.CORBA.Any  extract_any()
        throws org.omg.CORBA.BAD_OPERATION;
public  org.omg.CORBA.Object extract_Object()
        throws org.omg.CORBA.BAD_OPERATION;
public  java.lang.String extract_string()
        throws org.omg.CORBA.BAD_OPERATION;
public  java.lang.String  extract_wstring()
        throws org.omg.CORBA.BAD_OPERATION;
public  org.omg.CORBA.TypeCode extract_TypeCode()
        throws org.omg.CORBA.BAD_OPERATION;
```

Description

Obtains stored values from an Any type object.

Return Values

Normal termination: Returns values or objects obtained from an Any type object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.BAD_OPERATION

  IDL data type of the value specified by the Any type object and the one of the method do not match. No values are specified in the Any type object.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.12.5 org.omg.CORBA.Any.create_output_stream()

Name

org.omg.CORBA.Any.create_output_stream

Format

```
public org.omg.CORBA.portable.OutputStream boolean create_output_stream()
```

Description

This method creates an Any object output stream.

Return Values

Normal termination: Returns the made OutputStream object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

Note

When the IIOP Service (Java EE client) feature is used, it is necessary to use org.omg.CORBA.Any.read_value() in order to reflect data written to the created output stream in the Any type. A code example is shown below.

```
/* When write_xxx is used */
  org.omg.CORBA.ORB orb=org.omg.CORBA.ORB.init();
                    :
org.omg.CORBA.Any any = orb.create_any();
org.omg.CORBA.portable.OutputStream os = any.create_output_stream();
  char write_value = 'a';
  os.write_char(write_value);
  org.omg.CORBA.portable.InputStream is_tmp = os.create_input_stream();
  any.read_value(is_tmp, orb.get_primitive_tc(org.omg.CORBA.TCKind.tk_char));
  org.omg.CORBA.portable.InputStream is = any.create_input_stream();
  char result = is.read_char();
/* When write_xxx_array is used */
  org.omg.CORBA.ORB orb=org.omg.CORBA.ORB.init();
                    :
org.omg.CORBA.Any any = orb.create_any();
org.omg.CORBA.portable.OutputStream os = any.create_output_stream();
  char write_value[] = {'a', 'b', 'c', 'd', 'e'};
  int write_length = write_value.length;
```

```
int offset = 0;
os.write_long(write_length);
os.write_char_array(write_value, offset, write_length);
org.omg.CORBA.portable.InputStream is_tmp = os.create_input_stream();
any.read_value(is_tmp, orb.get_primitive_tc(org.omg.CORBA.TCKind.tk_long));
org.omg.CORBA.TypeCode write_tc = orb.get_primitive_tc(org.omg.CORBA.TCKind.tk_char);
any.read_value(is_tmp, orb.create_array_tc(write_length, write_tc));
org.omg.CORBA.portable.InputStream is = any.create_input_stream();
int read_length = is.read_long();
char result[] = new char[read_length];
is.read_char_array(result, offset, read_length);
```

3.12.6 org.omg.CORBA.Any.create_input_stream()

Name

org.omg.CORBA.Any.create_input_stream

Format

```
public org.omg.CORBA.portable.InputStream create_input_stream()
```

Description

This method creates an Any object input stream.

Return Values

Normal termination: Returns the made InputStream object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.12.7 org.omg.CORBA.Any.read_value()

Name

org.omg.CORBA.Any.read_value

Format

```
public void read_value( org.omg.CORBA.portable.InputStream is, org.omg.CORBA.TypeCode t )
      throws org.omg.CORBA.MARSHAL;
```

Description

This method reads (unmarshals) the Any object value from the specified input stream.

Parameters

is

InputStream object which reads the value stored in the Any type object

t

TypeCode object which represents the type of the value which has been read

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.MARSHAL

  There is an inconsistency between the specified TypeCode object and the value stored in the input stream.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to "Exception Information and Minor Codes to be Reported from the CORBA Service" and "Exception information output when Java EE is used" in the Java EE Operator's Guide.

Note

This method is not supported in the CORBA Service (Portable-ORB and pre-installed Java library).

3.13 OutputStream Class

This section explains the OutputStream class to insert an IDL data type in Any.

The APIs which are supported by the CORBA Service (ObjectDirector) and IIOP Service (Java EE client) in the OutputStream class are shown below.

API	CORBA Service (ObjectDirector)	IIOP Service (Java EE client)
Org.omg.CORBA.portable.OutputStream.write_<type>()	Y	Y
org.omg.CORBA.portable.OutputStream.write_<type>_array()	Y	Y

```
//Java
package org.omg.CORBA.portable;
abstract public class OutputStream
{
    abstract public void write_any(org.omg.CORBA.Any value);
    abstract public void write_boolean(boolean value);
    abstract public void write_char(char value);
    abstract public void write_double(double value);
    abstract public void write_float(float value);
    abstract public void write_long(int value);
    abstract public void write_longlong(long value);
    abstract public void write_Object(org.omg.CORBA.Object value);
    abstract public void write_octet(byte value);
    abstract public void write_short(short value);
    abstract public void write_string(java.lang.String value);
    abstract public void write_ulong(int value);
    abstract public void write_ulonglong(long value);
    abstract public void write_ushort(short value);
    abstract public void write_wchar(char value);
    abstract public void write_wstring(java.lang.String value);
    abstract public void write_boolean_array(boolean[] value, int offset, int length);
    abstract public void write_char_array(char[] value, int offset, int length);
    abstract public void write_double_array(double[] value, int offset, int length);
    abstract public void write_float_array(float[] value, int offset, int length);
    abstract public void write_long_array(int[] value, int offset, int length);
    abstract public void write_longlong_array(long[] value, int offset, int length);
    abstract public void write_octet_array(byte[] value, int offset, int length);
    abstract public void write_short_array(short[] value, int offset, int length);
    abstract public void write_ulong_array(int[] value, int offset, int length);
```

```
    abstract public void write_ulonglong_array(long[] value, int offset, int length);
    abstract public void write_ushort_array(short[] value, int offset, int length);
    abstract public void write_wchar_array(char[] value, int offset, int length);
}
```

3.13.1 org.omg.CORBA.portable.OutputStream.write_<type>()

Name

org.omg.CORBA.portable.OutputStream.write_<type>

Format

```
public void write_any(org.omg.CORBA.Any value)
public void write_boolean(boolean value)
public void write_char(char value)
public void write_double(double value)
public void write_float(float value)
public void write_long(int value)
public void write_longlong(long value)
public void write_Object(org.omg.CORBA.Object value)
public void write_octet(byte value)
public void write_short(short value)
public void write_string(java.lang.String value)
public void write_ulong(int value)
public void write_ulonglong(long value)
public void write_ushort(short value)
public void write_wchar(char value)
public void write_wstring(java.lang.String value)
```

Description

This method writes the specified value to this stream.

Parameters

value

The value to be written to the stream

Return Values

Normal termination: None.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.13.2 org.omg.CORBA.portable.OutputStream.write_<type>_array()

Name

org.omg.CORBA.portable.OutputStream.write_<type>_array

Format

```
public void write_boolean_array(boolean[] value, int offset, int length)
public void write_char_array(char[] value, int offset, int length)
public void write_double_array(double[] value, int offset, int length)
public void write_float_array(float[] value, int offset, int length)
public void write_long_array(int[] value, int offset, int length)
public void write_longlong_array(long[] value, int offset, int length)
public void write_octet_array(byte[] value, int offset, int length)
public void write_short_array(short[] value, int offset, int length)
public void write_ulong_array(int[] value, int offset, int length)
public void write_ulonglong_array(long[] value, int offset, int length)
public void write_ushort_array(short[] value, int offset, int length)
public void write_wchar_array(char[] value, int offset, int length)
```

Description

This method writes the specified array value to this stream.

Parameters

value

The value to be written to the stream

offset

The offset on the stream

length

The length of the buffer to be written

Return Values

Normal termination: None.

Abnormal termination: Issues the following exception:

- java.lang.ArrayIndexOutOfBoundsException
- java.lang.NegativeArraySizeException

  The value set for the parameter is invalid.

  This occurs when the IIOP Service (Java EE client) feature is used.
- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.14 InputStream Class

This section explains the InputStream class to extract an IDL data type from Any.

The APIs which are supported by the CORBA Service (ObjectDirector) and IIOP Service (Java EE client) in the InputStream class are shown below.

API	CORBA Service (ObjectDirector)	IIOP Service (Java EE client)
org.omg.CORBA.portable.InputStream.read_<type>()	Y	Y

API	CORBA Service (ObjectDirector)	IIOP Service (Java EE client)
org.omg.CORBA.portable.InputStream.read_<type>_array()	Y	Y

```
//Java
package org.omg.CORBA.portable;
abstract public class InputStream
{
    abstract public org.omg.CORBA.Any read_any();
    abstract public boolean read_boolean();
    abstract public char read_char();
    abstract public double read_double();
    abstract public float read_float();
    abstract public int read_long();
    abstract public long read_longlong();
    abstract public org.omg.CORBA.Object read_Object();
    abstract public byte read_octet();
    abstract public short read_short();
    abstract public java.lang.String read_string();
    abstract public int read_ulong();
    abstract public long read_ulonglong();
    abstract public short read_ushort();
    abstract public char read_wchar();
    abstract public java.lang.String read_wstring();
    abstract public void read_boolean_array(boolean[] value, int offset, int length);
    abstract public void read_char_array(char[] value, int offset, int length);
    abstract public void read_double_array(double[] value, int offset, int length);
    abstract public void read_float_array(float[] value, int offset, int length);
    abstract public void read_long_array(int[] value, int offset, int length);
    abstract public void read_longlong_array(long[] value, int offset, int length);
    abstract public void read_octet_array(byte[] value, int offset, int length);
    abstract public void read_short_array(short[] value, int offset, int length);
    abstract public void read_ulong_array(int[] value, int offset, int length);
    abstract public void read_ulonglong_array(long[] value, int offset, int length);
    abstract public void read_ushort_array(short[] value, int offset, int length);
    abstract public void read_wchar_array(char[] value, int offset, int length);
}
```

3.14.1 org.omg.CORBA.portable.InputStream.read_<type>()

Name

org.omg.CORBA.portable.InputStream.read_<type>

Format

```
public org.omg.CORBA.Any read_any()
public boolean read_boolean()
public char read_char()
public double read_double()
public float read_float()
public int read_long()
public long read_longlong()
public org.omg.CORBA.Object read_Object()
public byte read_octet()
public short read_short()
public java.lang.String read_string()
public int read_ulong()
public long read_ulonglong()
public short read_ushort()
```

```
public char read_wchar()
public java.lang.String read_wstring()
```

Description

This method reads the specified type from this stream.

Return Values

Normal termination: Returns the read data value.

Abnormal termination: Issues the following exception:

- java.lang.ArrayIndexOutOfBoundsException

  The TypeCode of the Any type used or the value used is invalid.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.14.2 org.omg.CORBA.portable.InputStream.read_<type>_array()

Name

org.omg.CORBA.portable.InputStream.read_<type>_array

Format

```
public void read_boolean_array(boolean[] value, int offset, int length)
public void read_char_array(char[] value, int offset, int length)
public void read_double_array(double[] value, int offset, int length)
public void read_float_array(float[] value, int offset, int length)
public void read_long_array(int[] value, int offset, int length)
public void read_longlong_array(long[] value, int offset, int length)
public void read_octet_array(byte[] value, int offset, int length)
public void read_short_array(short[] value, int offset, int length)
public void read_ulong_array(int[] value, int offset, int length)
public void read_ulonglong_array(long[] value, int offset, int length)
public void read_ushort_array(short[] value, int offset, int length)
public void read_wchar_array(char[] value, int offset, int length)
```

Description

This method reads the specified array type's value from this stream.

Parameters

value

The array where the data read from the stream is written

offset

The offset on the stream

length

The length of the buffer to be written

Return Values

Normal termination: None.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

  This occurs when the IIOP Service (Java EE client) feature is used.

- java.lang.ArrayIndexOutOfBoundsException

  The value set for the parameter is invalid.

  This occurs when the IIOP Service (Java EE client) feature is used.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.15 POA Linkage Interface

This section explains the PortableServer package's interface. It uses POA (Portable Object Adapter) for very portable server application implementations.

Note

The POA-related interface is not supported in the IIOP Service (Java EE client).

```
//Java
package org.omg.PortableServer;
abstract public class Servant
{
    final public org.omg.CORBA.Object  _this_object(){...}
    final public org.omg.CORBA.Object  _this_object(org.omg.CORBA.ORB orb){...}
    final public org.omg.CORBA.ORB  _orb(){...}
    final public void  _orb(org.omg.CORBA.ORB orb){...}
    final public org.omg.PortableServer.POA  _poa(){...}
    final public byte[]  _object_id(){...}
    private synchronized void  _getPOACurrent(){...} //private method
    public org.omg.PortableServer.POA  _default_POA(){...}
    //method called by ORB
    abstruct public java.lang.String[] _all_interfaces(org.omg.PortableServer.POA
poa,
             byte[]  objectID);
}

public interface POA extends org.omg.CORBA.Object
{
    abstract public org.omg.PortableServer.POA  create_POA(String adapter_name,
          org.omg.PortableServer.POAManager a_POAManager,
          org.omg.CORBA.Policy[] policies)
       throws org.omg.PortableServer.POAPackage.AdapterAlreadyExists,
    org.omg.PortableServer.POAPackage.InvalidPolicy;
    abstract public org.omg.PortableServer.POA  find_POA(String adapter_name,
          boolean activate_it)
       throws org.omg.PortableServer.POAPackage.AdapterNonExistent;
    abstract public void  destroy(boolean etherealize_objects, boolean
wait_for_completion);
```

```
    abstract public org.omg.PortableServer.ThreadPolicy
       create_thread_policy(org.omg.PortableServer.ThreadPolicyValue value);
    abstract public org.omg.PortableServer.LifespanPolicy
       create_lifespan_policy(org.omg.PortableServer.LifespanPolicyValue value);
    abstract public org.omg.PortableServer.IdUniquenessPolicy
       create_id_uniqueness_policy(org.omg.PortableServer.IdUniquenessPolicyValue
value);
    abstract public org.omg.PortableServer.IdAssignmentPolicy
       create_id_assignment_policy(org.omg.PortableServer.IdAssignmentPolicyValue
value);
    abstract public org.omg.PortableServer.ImplicitActivationPolicy
       create_implicit_activation_policy(
            org.omg.PortableServer.ImplicitActivationPolicyValue value);
    abstract public org.omg.PortableServer.ServantRetentionPolicy
       create_servant_retention_policy(
            org.omg.PortableServer.ServantRetentionPolicyValue value);
    abstract public org.omg.PortableServer.RequestProcessingPolicy
       create_request_processing_policy(
          org.omg.PortableServer.RequestProcessingPolicyValue value);
    abstract public org.omg.PortableServer.ServantManager  get_servant_manager()
       throws org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public void  set_servant_manager(org.omg.PortableServer.ServantManager
imgr)
       throws org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public org.omg.PortableServer.Servant  get_servant()
       throws org.omg.PortableServer.POAPackage.NoServant,
         org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public void  set_servant(org.omg.PortableServer.Servant p_servant)
       throws org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public byte[]  activate_object(org.omg.PortableServer.Servant p_servant)
       throws org.omg.PortableServer.POAPackage.ServantAlreadyActive,
          org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public void  activate_object_with_id(byte[] id,
          org.omg.PortableServer.Servant p_servant)
       throws org.omg.PortableServer.POAPackage.ServantAlreadyActive,
          org.omg.PortableServer.POAPackage.ObjectAlreadyActive,
          org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public void  deactivate_object(byte[] oid)
       throws org.omg.PortableServer.POAPackage.ObjectNotActive,
          org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public org.omg.CORBA.Object  create_reference(String intf)
       throws org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public org.omg.CORBA.Object  create_reference_with_id(byte[] oid, String
intf)
       throws org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public byte[]  servant_to_id(org.omg.PortableServer.Servant p_servant)
       throws org.omg.PortableServer.POAPackage.ServantNotActive,
          org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public org.omg.CORBA.Object  servant_to_reference(
          org.omg.PortableServer.Servant p_servant)
       throws org.omg.PortableServer.POAPackage.ServantNotActive,
          org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public org.omg.PortableServer.Servant reference_to_servant(
          org.omg.CORBA.Object reference)
       throws org.omg.PortableServer.POAPackage.ObjectNotActive,
          org.omg.PortableServer.POAPackage.WrongAdapter,
          org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public byte[]  reference_to_id(org.omg.CORBA.Object reference)
       throws org.omg.PortableServer.POAPackage.WrongAdapter,
          org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public org.omg.PortableServer.Servant id_to_servant(byte[] oid)
       throws org.omg.PortableServer.POAPackage.ObjectNotActive,
          org.omg.PortableServer.POAPackage.WrongPolicy;
```

```
    abstract public org.omg.CORBA.Object  id_to_reference(byte[] oid)
       throws org.omg.PortableServer.POAPackage.ObjectNotActive,
          org.omg.PortableServer.POAPackage.WrongPolicy;
    abstract public org.omg.PortableServer.AdapterActivator  the_activator();
    abstract public void  the_activator(org.omg.PortableServer.AdapterActivator arg);
    abstract public String  the_name();
    abstract public void  the_name(String arg);
    abstract public org.omg.PortableServer.POA  the_parent();
    abstract public org.omg.PortableServer.POAManager  the_POAManager();
}
abstract public class POAManager
{
    abstract public void  activate()
       throws org.omg.PortableServer.POAManagerPackage.AdapterInactive;
    abstract public void  hold_requests(boolean wait_for_completion)
       throws org.omg.PortableServer.POAManagerPackage.AdapterInactive;
    abstract public void  discard_requests(boolean wait_for_completion)
       throws org.omg.PortableServer.POAManagerPackage.AdapterInactive;
    abstract public void  deactivate(boolean etherealize_objects, boolean
wait_for_completion)
       throws org.omg.PortableServer.POAManagerPackage.AdapterInactive;
}
public interface ThreadPolicy  extends org.omg.CORBA.Object, org.omg.CORBA.Policy
{
    org.omg.PortableServer.ThreadPolicyValue  value();
}
public interface LifespanPolicy  extends org.omg.CORBA.Object, org.omg.CORBA.Policy
{
    org.omg.PortableServer.LifespanPolicyValue  value();
}
public interface IdUniquenessPolicy  extends org.omg.CORBA.Object,
org.omg.CORBA.Policy
{
    org.omg.PortableServer.IdUniquenessPolicyValue  value();
}
public interface IdAssignmentPolicy  extends
org.omg.CORBA.Object,org.omg.CORBA.Policy
{
    org.omg.PortableServer.IdAssignmentPolicyValue  value();
}
public interface ImplicitActivationPolicy  extends org.omg.CORBA.Object,
org.omg.CORBA.Policy
{
    org.omg.PortableServer.ImplicitActivationPolicyValue  value();
}
public interface ServantRetentionPolicy  extends org.omg.CORBA.Object,
org.omg.CORBA.Policy
{
    org.omg.PortableServer.ServantRetentionPolicyValue  value();
}
public interface RequestProcessingPolicy extends org.omg.CORBA.Object,
org.omg.CORBA.Policy
{
    org.omg.PortableServer.RequestProcessingPolicyValue value();
}
public class Current implements org.omg.CORBA.Current
{
    public org.omg.PortableServer.POA  get_POA()
       throws org.omg.PortableServer.CurrentPackage.NoContext {...}
    public byte[] get_object_id()
              throws org.omg.PortableServer.CurrentPackage.NoContext {...}
}
```

3.15.1 Servant Interface

This section describes the Servant interface.

3.15.1.1 org.omg.PortableServer.Servant._this_object()

Name

org.omg.PortableServer.Servant._this_object

Format

```
public final org.omg.CORBA.Object _this_object();
```

Description

Returns the object reference assigned to the Servant object. It is necessary to relate the targeted Servant object to the POA to which the RETAIN policy is set.

Parameters

None

Return Values

Normal termination: Returns the org.omg.CORBA.Object type object that contains the Servant object's object reference.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.1.2 org.omg.PortableServer.Servant._orb()

Name

org.omg.PortableServer.Servant._orb()

Format

```
public final org.omg.CORBA.ORB _orb();
```

Description

Obtains the instance of ORB that is registered in the Servant object.

Parameters

None

Return Values

Normal termination: Returns the instance of ORB that is registered in the Servant object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.1.3 org.omg.PortableServer.Servant._poa()

Name

org.omg.PortableServer.Servant._poa()

Format

```
public final org.omg.PortableServer.POA _poa();
```

Description

Returns the POA object that controls the Servant object.

Return Values

Normal termination: Returns the POA object that controls the Servant object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.1.4 org.omg.PortableServer.Servant._object_id()

Name

org.omg.PortableServer.Servant._object_id

Format

```
public final byte[] _object_id();
```

Description

Returns the object ID assigned to the Servant object. This method has the same affect as org.omg.PortableServer.Current.get_Object_id().

Return Values

Normal termination: Returns the object ID assigned to the Servant object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.1.5 org.omg.PortableServer.Servant._default_POA()

Name

org.omg.PortableServer.Servant._default_POA

Format

```
public org.omg.PortableServer.POA _default_POA();
```

Description

When called from a Servant object not controlled by the POA, this method will return the default POA object. The root POA is based on the ORB to which the Servant object is connected. It will be returned as the default POA. However, you can return a different POA object. This is possible because you can override this method from this class's subclass.

Return Values

Normal termination: Returns the default POA object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2 POA Interface

This section describes the POA interface.

3.15.2.1 org.omg.PortableServer.POA.create_POA()

Name

org.omg.PortableServer.POA.create_POA

Format

```
public org.omg.PortableServer.POA create_POA(
       java.lang.String adapter_name,
       org.omg.PortableServer.POAManager a_POAManager,
       org.omg.CORBA.Policy[] policies)
throws org.omg.PortableServer.POAPackage.AdapterAlreadyExists,
       org.omg.PortableServer.POAPackage.InvalidPolicy;
```

Description

Generates a child POA object for the relevant POA object. The adapter_name parameter specifies a name that lets a child object identify a new child object that has the same parent.

Parameters

adapter_name

the generated child POA's name

a_POAManager

the associated POA manager's object.

policies

the PolicyList object that contains the policy that specifies the child POA.

Return Values

Normal termination: Returns the generated child POA object.

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.AdapterAlreadyExists

  The specified POA object already has a child POA object with this name.

- org.omg.PortableServer.POAPackage.InvalidPolicy

  Specified policy object construction is invalid.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Note

The OMG CORBA rule specifies that a "null" at the second parameter or "a_POAManager" creates a new POAManager object and links it to new POA objects. In the Interstage CORBA service, however, one process contains only one POAManager object. Only one CORBA service POAManager object is created when the RootPOA object is created.

3.15.2.2 org.omg.PortableServer.POA.find_POA()

Name

org.omg.PortableServer.POA.find_POA

Format

```
public org.omg.PortableServer.POA find_POA()(
       java.lang.String adapter_name,
       boolean activate_it)
throws org.omg.PortableServer.POAPackage.AdapterNonExistent;
```

Description

In a specified POA object, this returns the child POA objects that contain names specified by adapter_name. If there are none, and if the activate_it parameter is TRUE, and if an AdaperActivator exists in the POA object, it will be invoked.

The AdaptorActivator will generate a child POA containing the specified name. If this succeeds the main method will return the newly generated child POA object.

In other cases: Issues an AdapterNonExistent exception.

Parameters

adapter_name

the child POA's name.

activate_it

true

If the specified child POA name does not exist, this will activate a new child POA.

false

If the specified child POA name does not exist, this do nothing but issue an AdapterNonExistent exception.

Return Values

Normal termination: Returns a child POA object that contains the name specified in adapter_name. Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.AdapterNonExistent

  Refer to the Description.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.3 org.omg.PortableServer.POA.destroy()

Name

org.omg.PortableServer.POA.destroy

Format

```
public void destroy(boolean etherealize_objects,
       boolean wait_for_completion);
```

Description

Destroys the specified POA object as well as its descendent POA objects.

Parameters

etherealize_objects

true

If the POA has a RETAIN policy, its registered ServantManager issues an etherealize() method for the objects active in the AOM.

false

It does not do the above.

wait_for_completion

true

Returns only after all requests in the process are completed, and all etherealize() methods are completed.

false

Returns as soon as it destroys the POA object.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.4 org.omg.PortableServer.POA.create_thread_policy()

Name

org.omg.PortableServer.POA.create_thread_policy

Format

```
public org.omg.PortableServer.ThreadPolicy
create_thread_policy( org.omg.PortableServer.ThreadPolicyValue value);
```

Description

Generates a TheadPolicy object that contains the policy specified in value.

Parameters

value

You can specify the following values.

org.omg.PortableServer.ThreadPolicyValue.ORB_CTRL_MODEL

org.omg.PortableServer.ThreadPolicyValue.SINGLE_THREAD_MODEL

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Return Values

Normal termination: Returns the generated ThreadPolicy object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.5 org.omg.PortableServer.POA.create_lifespan_policy()

Name

org.omg.PortableServer.POA.create_lifespan_policy

Format

```
public org.omg.PortableServer.LifespanPolicy
create_lifespan_policy(
       org.omg.PortableServer.LifespanPolicyValue value);
```

Description

Generates a LifespanPolicy object that contains the policy specified in value.

Parameters

value

You can specify the following values.

org.omg.CORBA.PortableServer.LifespanPolicyValue.TRANSIENT

org.omg.CORBA.PortableServer.LifespanPolicyValue.PERSISTENT

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Return Values

Normal termination: Returns the generated LifespanPolicy object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.6 org.omg.PortableServer.POA.create_id_uniqueness_policy()

Name

org.omg.PortableServer.POA.create_id_uniqueness_policy

Format

```
public org.omg.PortableServer.IdUniquenessPolicy
create_id_uniqueness_policy(
    org.omg.PortableServer.IdUniquenessPolicyValue value);
```

Description

Generates an IdUniquenessPolicy object that contains the policy specified in value.

Parameters

value

You can specify the following values.

org.omg.CORBA.PortableServer.IdUniquenessPolicyValue.UNIQUE_ID,

org.omg.CORBA.PortableServer.IdUniquenessPolicyValue.MULTIPLE_ID

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Return Values

Normal termination: Returns the generated IdUniquenessPolicy object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.7 org.omg.PortableServer.POA.create_id_assignment_policy()

Name

org.omg.PortableServer.POA.create_id_assignment _policy

Format

```
public org.omg.PortableServer.IdAssignmentPolicy
create_id_assignment_policy(
       org.omg.PortableServer.IdAssignmentPolicyValue value);
```

Description

Generates an IdAssignmentPolicy object that contains the policy specified in value.

Parameters

value

You can specify the following values.

org.omg.PortableServer.IdAssignmentPolicyValue.USER_ID

org.omg.PortableServer.IdAssignmentPolicyValue.SYSTEM_ID

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Return Values

Normal termination: Returns the generated IdAssignmentPolicy object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.8 org.omg.PortableServer.POA.create_implicit_activation_policy()

Name

org.omg.PortableServer.POA.create_implicit_activation_policy

Format

```
public org.omg.PortableServer.ImplicitActivationPolicy
create_implicit_activation_policy(
       org.omg.PortableServer.ImplicitActivationPolicyValue value);
```

Description

Generates an ImplicitActivationPolicy object that contains the policy specified in value.

Parameters

value

You can specify the following values.

org.omg.PortableServer.ImplicitActivationPolicy.IMPLICIT_ACTIVATION

org.omg.PortableServer.ImplicitActivationPolicy.NO_IMPLICIT_ACTIVATION

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Return Values

Normal termination: Returns the generated ImplicitActivationPolicy object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.9 org.omg.PortableServer.POA.create_servant_retention_policy()

Name

org.omg.PortableServer.POA.create_servant_retention_policy

Format

```
public org.omg.PortableServer.ServantRetentionPolicy
create_servant_retention_policy(
       org.omg.PortableServer.ServantRetentionPolicyValue value);
```

Description

Generates a ServantRetentionPolicy object that contains the policy specified in value.

Parameters

value

You can specify the following values.

org.omg.PortableServer.ServantRetentionPolicyValue.RETAIN

org.omg.PortableServer.ServantRetentionPolicyValue.NON_RETAIN

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Return Values

Normal termination: Returns the generated ServantRetentionPolicy object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.10 org.omg.PortableServer.POA.create_request_processing_policy()

Name

org.omg.PortableServer.POA.create_request_processing_policy

Format

```
public org.omg.PortableServer.RequestProcessingPolicy
create_request_processing_policy(
org.omg.PortableServer.RequestProcessingPolicyValue value);
```

Description

Generates a RequestProcessingPolicy object that contains the policy specified in value.

Parameters

value

You can specify the following values.

org.omg.PortableServer.RequestProcessingPolicyValue.USE_ACTIVE_OBJECT_MAP_ ONLY

org.omg.PortableServer.RequestProcessingPolicyValue.USE_DEFAULT_SERVANT

org.omg.PortableServer.RequestProcessingPolicyValue.USE_SERVANT_MANAGER

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Return Values

Normal termination: Returns the generated RequestProcessingPolicy object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.11 org.omg.PortableServer.POA.get_servant_manager()

Name

org.omg.PortableServer.POA.get_servant_manager

Format

```
public org.omg.PortableServer.ServantManager get_servant_manager()
throws org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

Returns the ServantManager object that is associated with the POA. If there is none, it returns a null. A USE_SERVANT_MANAGER policy must be set in the specified POA.

Return Values

Normal termination: Returns the associated ServantManager object.

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.WrongPolicy

  USE_SERVANT_MANAGER policy not set up.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.12 org.omg.PortableServer.POA.set_servant_manager()

Name

org.omg.PortableServer.POA.set_servant_manager

Format

```
public void  org.omg.PortableServer.POA.set_servant_manager(
             org.omg.PortableServer.ServantManager imgr )
throws org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

Sets up a default ServantManager that is associated with the POA. A USE_SERVANT_MANAGER policy must be set in the specified POA. You can only issue this method once for any one POA.

Parameters

imgr

the associated default ServantManager object.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.WrongPolicy

  USE_SERVANT_MANAGER policy not set up.

- org.omg.CORBA.BAD_INV_ORDER

  Multiple issues for one POA

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.13 org.omg.PortableServer.POA.get_servant()

Name

org.omg.PortableServer.POA.get_servant

Format

```
public org.omg.PortableServer.Servant get_servant()
throws org.omg.PortableServer.POAPackage.NoServant,
       org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

Returns the default Servant associated with the POA. A USE_DEFAULT_SERVANT policy must be set in the specified POA.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.NoServant

  There is no Servant associated with the specified POA.

- org.omg.PortableServer.POAPackage.WrongPolicy

  USE_SERVANT_MANAGER policy not set up.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.14 org.omg.PortableServer.POA.set_servant()

Name

org.omg.PortableServer.POA.set_servant

Format

```
public void set_servant(org.omg.PortableServer.Servant p_servant)
throws org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

Registers the Servant specified by p_servant as the Default Servant for the specified POA. When the system cannot find a Servant that corresponds to a request in the active object map (AOM), it will use this one. A USE_DEFAULT_SERVANT policy must be set in the specified POA.

Parameters

p_servant

the Default Servant object you want to register

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.WrongPolicy

  USE_DEFAULT_SERVANT policy not set up.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.15 org.omg.PortableServer.POA.activate_object()

Name

org.omg.PortableServer.POA.activate_object

Format

```
public byte[]
activate_object(org.omg.PortableServer.Servant p_servant)
throws org.omg.PortableServer.POAPackage.ServantAlreadyActive,
      org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

Generates an object ID and stores it and the specified Servant in the specified POA's active object map (AOM). Returns the object ID. A SYSTEM_ID policy and a RETAIN policy must be set in the specified POA.

Parameters

p_servant

the Servant object you want to register in the AOM.

Return Values

Normal termination: Returns the object ID for the registered Servant.

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.ServantAlreadyActive

  The POA has a UNIQUE_ID policy, and the Servant specified by the p_servant parameter already exists in the AOM.

- org.omg.PortableServer.POAPackage.WrongPolicy

  SYSTEM_ID and RETAIN policies not set up.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.16 org.omg.PortableServer.POA.activate_object_with_id()

Name

org.omg.PortableServer.POA.activate_object_with_id

Format

```
public void activate_object_with_id(byte[] oid,
org.omg.PortableServer.Servant p_servant)
throws org.omg.PortableServer.POAPackage.ObjectAlreadyActive,
       org.omg.PortableServer.POAPackage.ServantAlreadyActive,
       org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

Stores the object ID specified by the oid parameter, and the Servant association specified by the p_servant parameter in the specified POA's AOM. A RETAIN policy must be set in the specified POA.

Parameters

oid

the object ID you want to store in the AOM

p_servant

the Servant object you want to associate with oid and register in the AOM

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.ObjectAlreadyActive

  The specified POA already has an active CORBA object specified by the oid parameter.

- org.omg.PortableServer.POAPackage.ServantAlreadyActive

  The POA has a UNIQUE_ID policy, and the Servant specified by the p_servant parameter already exists in the AOM.

- org.omg.PortableServer.POAPackage.WrongPolicy

  RETAIN policy not set up.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.17 org.omg.PortableServer.POA.deactivate_object()

Name

org.omg.PortableServer.POA.deactivate_object

Format

```
public void deactivate_object(byte[] oid)
throws org.omg.PortableServer.POAPackage.ObjectNotActive,
       org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

Deletes the object ID specified by the oid parameter, and the Servant association specified by the p_servant parameter from the specified POA's active object map (AOM). When ServerManager is associated with the specified POA, the org.omg.PortableServer.ServantLocator.etherealize() method invokes oid and Servant as arguments. A RETAIN policy must be set in the specified POA.

Parameters

oid

the object ID you want to remove from the AOM

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.ObjectNotActive

  There is no active object associated with the specified object ID.

- org.omg.PortableServer.POAPackage.WrongPolicy

  RETAIN policy not set up.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.18 org.omg.PortableServer.POA.create_reference()

Name

org.omg.PortableServer.POA.create_reference

Format

```
public org.omg.CORBA.Object create_reference(
    java.lang.String intf)
    throws org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

Generates an object reference that includes object IDs created by the POA, and interface repository IDs specified by the intf parameter. A SYSTEM_ID policy must be set in the specified POA.

Parameters

intf

Interface Repository ID for the newly generated object reference

Return Values

Normal termination: Returns the newly generated object reference.

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.WrongPolicy

  SYSTEM_ID policy not set up.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.19 org.omg.PortableServer.POA.create_reference_with_id()

Name

org.omg.PortableServer.POA.create_reference_with_id

Format

```
public org.omg.CORBA.Object create_reference_with_id(
    byte[] oid,
    java.lang.String intf);
```

Description

Generates an object reference that includes object IDs specified by the oid parameter, and interface repository IDs specified by the intf parameter.

Parameters

oid

the object ID to for the newly generated object reference

intf

interface repository ID for the newly generated object reference

Return Values

Normal termination: Returns the newly generated object reference.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.BAD_PARAM

  Specified oid value differs from system generated value. Occurs when a SYSTEM_ID policy is set in the relevant POA.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.20 org.omg.PortableServer.POA.servant_to_id()

Name

org.omg.PortableServer.POA.servant_to_id

Format

```
public byte[]
servant_to_id( org.omg.PortableServer.Servant p_servant)
throws org.omg.PortableServer.POAPackage.ServantNotActive,
       org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

Depending on the policy set in the POA, it will do either (a) or (b).

(a) If the POA has a UNIQUE_ID policy, and if the Servant specified by the p_servant parameter is active, it will return the repository ID associated with the Servant.

(b) If the POA has an IMPLICIT_ACTIVATION policy and a MULTIPLE_ID policy, or if it has an IMPLICIT_ACTIVATION policy and the specified Servant is not active, the Servant will be activated using an object ID generated by the POA, and the interface repository ID associated with the Servant. Then it will return the object ID.

A RETAIN policy and either a UNIQUE_ID policy, or an IMPLICIT_ACTIVATION policy must be set in the specified POA.

Parameters

p-servant

the Servant (Servant type object) that will obtain the object ID

Return Values

Normal termination: Returns the object ID for the specified Servant.

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.ServantNotActive

  Other than (a) or (b)

- org.omg.PortableServer.POAPackage.WrongPolicy

  One of the following combinations is not set: RETAIN policy and UNIQUE_ID policy, or RETAIN policy and IMPLICIT_ACTIVATION policy.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.21 org.omg.PortableServer.POA.servant_to_reference()

Name

org.omg.PortableServer.POA.servant_to_reference

Format

```
public org.omg.CORBA.Object servant_to_reference(
                            org.omg.PortableServer.Servant p_servant)
throws org.omg.PortableServer.POAPackage.ServantNotActive,
       org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

Depending on the policy set in the POA, it will do either (a) or (b).

(a) If the POA has a UNIQUE_ID policy, and if the Servant specified by the p_servant parameter is active, it will return the object reference that contains the information used to activate the Servant.

(b) If the POA has an IMPLICIT_ACTIVATION policy and a MULTIPLE_ID policy, or if it has an IMPLICIT_ACTIVATION policy and the specified Servant is not active, the Servant will be activated with an object ID generated by the POA, and the interface ID associated with the Servant. Then it will return the object reference. A RETAIN policy and either a UNIQUE_ID policy, or an IMPLICIT_ACTIVATION policy must be set in the specified POA.

Parameters

p-servant

the Servant (Servant type object) that will obtain the object reference

Return Values

Normal termination: Returns the object reference for the specified Servant.

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.ServantNotActive

  Other than (a) or (b)

- org.omg.PortableServer.POAPackage.WrongPolicy

  One of the following combinations is not set: RETAIN policy and UNIQUE_ID policy, or RETAIN policy and IMPLICIT_ACTIVATION policy.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.22 org.omg.PortableServer.POA.reference_to_servant()

Name

org.omg.PortableServer.POA.reference_to_servant

Format

```
org.omg.PortableServer.Servant
public org.omg.PortableServer.Servant reference_to_servant(
                                     org.omg.CORBA.Object reference)
throws org.omg.PortableServer.POAPackage.ObjectNotActive,
       org.omg.PortableServer.POAPackage.WrongAdapter,
       org.omg.PortableServer.POAPackage.WrongPolicy
```

Description

Depending on the policy set in the POA, it will do either (a) or (b).

(a) If the POA has a RETAIN policy, and if the AOM contains an object reference specified by the reference parameter, inside the AOM it will return the Servant associated with the object reference.

(b) If the POA has a USE_DEFAULT_SERVANT policy, and if a Default Servant is set in the POA, it will return the Default Servant. A RETAIN policy or a USE_DEFAULT_SERVANT policy must be set in the specified POA.

Parameters

reference

object reference for the Servant you want to obtain

Return Values

Normal termination: Returns the Servant referred to by the specified object reference.

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.ObjectNotActive

  Other than (a) or (b)

- org.omg.PortableServer.POAPackage.WrongAdapter

  This POA cannot generate the object reference.

- org.omg.PortableServer.POAPackage.WrongPolicy

  RETAIN or USE_DEFAULT_SERVANT policy not set up.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.23 org.omg.PortableServer.POA.reference_to_id()

Name

org.omg.PortableServer.POA.reference_to_id

Format

```
public byte[] reference_to_id(org.omg.CORBA.Object reference)
throws org.omg.PortableServer.POAPackage.WrongAdapter,
       org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

Returns a value for the object ID contained in the object reference specified by the reference parameter. Object references specified by the reference parameter are only valid when they have been generated by the POA object that invoked this method.

Parameters

reference

The object reference that contains the targeted object ID

Return Values

Normal termination: Returns the value for the object ID contained in the specified object reference.

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.WrongAdapter

  This POA cannot generate the object reference.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.24 org.omg.PortableServer.POA.id_to_servant()

Name

org.omg.PortableServer.POA.id_to_servant

Format

```
public Servant id_to_servant(byte[] oid)
throws org.omg.PortableServer.POAPackage.ObjectNotActive,
       org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

Returns the active Servant associated with the object ID specified by the oid parameter. A RETAIN policy must be set in the specified POA.

Parameters

oid

object ID associated with the Servant you want to obtain

Return Values

Normal termination: Returns the active Servant associated with the specified object ID.

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.ObjectNotActive

  The specified object ID is not active.

- org.omg.PortableServer.POAPackage.WrongPolicy

  RETAIN policy not set up.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.25 org.omg.PortableServer.POA.id_to_reference()

Name

org.omg.PortableServer.POA.id_to_reference

Format

```
public org.omg.CORBA.Object id_to_reference(byte[] oid)
throws org.omg.PortableServer.POAPackage.ObjectNotActive,
       org.omg.PortableServer.POAPackage.WrongPolicy;
```

Description

If the object with object ID values specified by the oid parameter is currently active, it returns the object reference that contains the information used to activate the object. A RETAIN policy must be set in the specified POA.

Parameters

oid

object ID associated with the Servant that will obtain the object reference

Return Values

Normal termination: Returns the object reference.

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAPackage.ObjectNotActive

  The specified object ID is not active.

- org.omg.PortableServer.POAPackage.WrongPolicy

  RETAIN policy not set up.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.26 org.omg.PortableServer.POA.the_activator()

Name

org.omg.PortableServer.POA.the_activator

Format

```
(1) public AdapterActivator the_activator();
(2) public void the_activator(org.omg.PortableServer.AdapterActivator arg);
```

Description

(1) Returns the AdapterActivator object that is set in the POA.

(2) Sets the AdapterActivator object in the POA.

Use an AdapterActivator object to dynamically generate child POAs. If a POA receives a request from ORB but cannot find a child POA with the specified name (POA-ID), it calls the AdapterActivator object's unknown_adapter() method. This method will generate a child POA with the name specified. You must implement the unknown_adapter method from an application.

Parameters

(1) None

(2)

arg

The AdapterActivator object to be set in the POA.

Return Values

(1) Normal termination: Returns the AdapterActivator object that is set in the POA.

(2) Normal termination: none

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.27 org.omg.PortableServer.POA.the_name()

Name

org.omg.PortableServer.POA.the_name

Format

```
public  java.lang.String the_name();
```

Description

Returns the POA object's name.

Return Values

Normal termination: Returns a String type object that shows the relevant POA object name.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.28 org.omg.PortableServer.POA.the_parent()

Name

org.omg.PortableServer.POA.the_parent

Format

```
public org.omg.PortableServer.POA the_parent();
```

Description

Returns the POA object's parent object.

Return Values

Normal termination: Returns the POA object's parent object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.2.29 org.omg.PortableServer.POA.the_POAManager()

Name

org.omg.PortableServer.POA.the_POAManager

Format

```
public abstract org.omg.PortableServer.POAManager the_POAManager();
```

Description

Returns the POAManager object that is associated with the POA object.

Return Values

Normal termination: Returns the POAManager object that is associated with the POA object.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.3 POAManager Interface

This section describes the POAManager interface.

3.15.3.1 org.omg.PortableServer.POAManager.activate()

Name

org.omg.PortableServer.POAManager.activate

Format

```
public void activate()
throws org.omg.PortableServer.POAManagerPackage.AdapterInactive;
```

Description

Converts the POA manager to active status. When the POA manager is active, the associated POA can process requests.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAManagerPackage.AdapterInactive

  Method issued when POA manager is inactive.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Note

Invoking this method causes the server application to stand by to receive requests.

3.15.3.2 org.omg.PortableServer.POAManager.hold_requests()

Name

org.omg.PortableServer.POAManager.hold_requests

Format

```
public void hold_requests(boolean wait_for_completion)
throws org.omg.PortableServer.POAManagerPackage.AdapterInactive;
```

Description

Converts the POA manager to holding status. When the POA manager is in holding status, the associated POA places incoming requests in a queue.

Parameters

wait_for_completion

true

This method will not return until one of the two following conditions is met: 1. the currently processing requests in all of the POAs associated with the POA manager are gone; 2. the POA manager is taken out of holding status.

false

This method will return as soon as the POA manager's status is changed.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAManagerPackage.AdapterInactive

  Method issued when POA manager is inactive.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.3.3 org.omg.PortableServer.POAManager.discard_requests()

Name

org.omg.PortableServer.POAManager.discard_requests

Format

```
public void hold_requests(boolean wait_for_completion)
throws org.omg.PortableServer.POAManagerPackage.AdapterInactive;
```

Description

Converts the POA manager to discarding status. When the POA manager is in discarding status, the associated POA discards incoming requests. It also discards any requests in the queue that are not currently being processed. When requests are discarded, it returns a TRANSIENT exception to the client.

Parameters

wait_for_completion

true

This method will not return until one of the two following conditions is met: 1. the currently processing requests in all of the POAs associated with the POA manager are gone; 2. the POA manager is taken out of discarding status.

false

This method will return as soon as the POA manager's status is changed.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAManagerPackage.AdapterInactive

  Method issued when POA manager is inactive.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.3.4 org.omg.PortableServer.POAManager.deactivate()

Name

org.omg.PortableServer.POAManager.deactivate

Format

```
public void deactivate(boolean etherealize_objects,
                       boolean wait_for_completion)
throws org.omg.PortableServer.POAManagerPackage.AdapterInactive;
```

Description

Converts the POA manager to inactive status. When the POA manager is in inactive status, the associated POA rejects both new and unprocessed requests.

Parameters

etherealize_objects

true

If POAs associated with the POA manager have a RETAIN policy and a USER_SERVANT_MANAGER policy, the servant manager associated with those POAs will invoke an org.omg.PortableServer.ServantActivator.etherealize() operation for active objects.

false

It will not invoke the etherealize() operation.

wait_for_completion

true

This method will not return until there are no more requests being processed in any of the POAs associated with the POA manager.

false

This method will return as soon as the POA manager's status is changed.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.PortableServer.POAManagerPackage.AdapterInactive

  Method issued when POA manager is inactive.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.4 ThreadPolicy Interface

This section describes the ThreadPolicy interface.

3.15.4.1 org.omg.PortableServer.ThreadPolicy.value()

Name

org.omg.PortableServer.ThreadPolicy.value

Format

```
org.omg.PortableServer.ThreadPolicyValue value();
```

Description

Returns the policy value that is set in the ThreadPolicy object.

Return Values

Normal termination: Returns one of the following values (ThreadPolicyValue objects):

- org.omg.PortableServer.ThreadPolicyValue.ORB_CTRL_MODEL
- org.omg.PortableServer.ThreadPolicyValue.SINGLE_THREAD_MODEL

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.5 LifespanPolicy Interface

This section describes the LifespanPolicy interface.

3.15.5.1 org.omg.PortableServer.LifespanPolicy.value()

Name

org.omg.PortableServer. LifespanPolicy. value

Format

```
org.omg.PortableServer.LifespanPolicyValue value();
```

Description

Returns the policy value that is set in the LifespanPolicy object.

Return Values

Normal termination: Returns one of the following values (LifespanPolicyValue objects):

- org.omg.PortableServer.LifespanPolicyValue.TRANSIENT
- org.omg.PortableServer.LifespanPolicyValue.PERSISTENT

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.6 IdUniquenessPolicy Interface

This section describes the IdUniquenessPolicy interface.

3.15.6.1 org.omg.PortableServer.IdUniquenessPolicy.value()

Name

org.omg.PortableServer. IdUniquenessPolicy. value

Format

```
org.omg.PortableServer.IdUniquenessPolicyValue value();
```

Description

Returns the policy value that is set in the IdUniquenessPolicy object.

Return Values

Normal termination: Returns one of the following values (IdUniquenessPolicyValue objects):

- org.omg.PortableServer.IdUniquenessPolicyValue.UNIQUE_ID
- org.omg.PortableServer. IdUniquenessPolicyValue.MULTIPLE_ID

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.7 IdAssignmentPolicy Interface

This section describes the IdAssignmentPolicy interface.

3.15.7.1 org.omg.PortableServer.IdAssignmentPolicy.value()

Name

org.omg.PortableServer. IdAssignmentPolicy. value

Format

```
org.omg.PortableServer.IdAssignmentPolicyValue value();
```

Description

Returns the policy value that is set in the IdAssignmentPolicy object.

Return Values

Normal termination: Returns one of the following values (IdAssignmentPolicyValue objects):

- org.omg.PortableServer.IdAssignmentPolicyValue.USER_ID
- org.omg.PortableServer.IdAssignmentPolicyValue.SYSTEM_ID

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.8 ImplicitActivationPolicy Interface

This section describes the ImplicitActivationPolicy interface.

3.15.8.1 org.omg.PortableServer.ImplicitActivationPolicy. value()

Name

org.omg.PortableServer. ImplicitActivationPolicy. value

Format

```
org.omg.PortableServer.ImplicitActivationPolicyValue value();
```

Description

Returns the policy value that is set in the ImplicitActivationPolicy object.

Return Values

Normal termination: Returns one of the following values (ImplicitActivationPolicyValue objects):

- org.omg.PortableServer.ImplicitActivationPolicyValue.IMPLICIT_ACTIVATION
- org.omg.PortableServer.ImplicitActivationPolicyValue.NO_IMPLICIT_ACTIVATION

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.9 ServantRetentionPolicy Interface

This section describes the ServantRetentionPolicy interface.

3.15.9.1 org.omg.PortableServer.ServantRetentionPolicy.value()

Name

org.omg.PortableServer. ServantRetentionPolicy. value

Format

```
org.omg.PortableServer.ServantRetentionPolicyValue value();
```

Description

Returns the policy value that is set in the ServantRetentionPolicy object.

Return Values

Normal termination: Returns one of the following values (ServantRetentionPolicyValue objects):

- org.omg.PortableServer.ServantRetentionPolicyValue.RETAIN
- org.omg.PortableServer.ServantRetentionPolicyValue.NON_RETAIN

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.15.10 RequestProcessingPolicy Interface

This section describes the RequestProcessingPolicy interface.

3.15.10.1 org.omg.PortableServer.RequestProcessingPolicy.value()

Name

org.omg.PortableServer. RequestProcessingPolicy.value

Format

```
org.omg.PortableServer.RequestProcessingPolicyValue value();
```

Description

Returns the policy value that is set in the RequestProcessingPolicy object.

Return Values

Normal termination: Returns one of the following values (RequestProcessingPolicyValue objects):

- org.omg.PortableServer.RequestProcessingPolicyValue.USE_ACTIVE_OBJECT_MAP_ ONLY
- org.omg.PortableServer.RequestProcessingPolicyValue.USE_DEFAULT_SERVANT
- org.omg.PortableServer.RequestProcessingPolicyValue.USE_SERVANT_MANAGER

For details on POA objects, refer to the Distributed Application Development Guide (CORBA Service Edition).

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.16 Naming Service Interface

This section will explain the methods available in the Naming service.

The APIs which are supported by the CORBA Service (ObjectDirector) and IIOP Service (Java EE client) in the Naming Service method are shown below.

API	CORBA Service (ObjectDirector)	IIOP Service (Java EE client)
org.omg.CosNaming.NamingContext.bind()	Y	Y
org.omg.CosNaming.NamingContext.rebind()	Y	Y
org.omg.CosNaming.NamingContext.bind_context()	Y	Y
org.omg.CosNaming.NamingContext.rebind_context()	Y	Y
org.omg.CosNaming.NamingContext.resolve()	Y	Y
org.omg.CosNaming.NamingContext.unbind()	Y	Y
org.omg.CosNaming.NamingContext.new_context()	Y	Y
org.omg.CosNaming.NamingContext.bind_new_context()	Y	Y
org.omg.CosNaming.NamingContext.destroy()	Y	Y
org.omg.CosNaming.NamingContext.list()	Y	Y
org.omg.CosNaming.BindingIterator.next_one()	Y	Y
org.omg.CosNaming.BindingIterator.next_n()	Y	Y
org.omg.CosNaming.BindingIterator.destroy()	Y	Y
org.omg.CosNaming.NameComponent.NameComponent()	Y	Y
org.omg.CosNaming.NamingContextExt.to_string()	Y	Y
org.omg.CosNaming.NamingContextExt.to_name()	Y	Y
org.omg.CosNaming.NamingContextExt.to_url()	Y	Y
org.omg.CosNaming.NamingContextExt.resolve_str()	Y	Y

```
//Java
package org.omg.CosNaming;
public interface NamingContext  extends org.omg.CORBA.Object
```

```
{
    public void  bind(org.omg.CosNaming.NameComponent[] n, org.omg.CORBA.Object obj)
       throws org.omg.CosNaming.NamingContextPackage.NotFound,
          org.omg.CosNaming.NamingContextPackage.CannotProceed,
          org.omg.CosNaming.NamingContextPackage.InvalidName,
          org.omg.CosNaming.NamingContextPackage.AlreadyBound;
    public void  rebind(org.omg.CosNaming.NameComponent[] n, org.omg.CORBA.Objectobj)
       throws org.omg.CosNaming.NamingContextPackage.NotFound,
          org.omg.CosNaming.NamingContextPackage.CannotProceed,
          org.omg.CosNaming.NamingContextPackage.InvalidName;
    public void  bind_context(org.omg.CosNaming.NameComponent[] n,
          org.omg.CosNaming.NamingContext nc)
       throws org.omg.CosNaming.NamingContextPackage.NotFound,
          org.omg.CosNaming.NamingContextPackage.CannotProceed,
          org.omg.CosNaming.NamingContextPackage.InvalidName,
          org.omg.CosNaming.NamingContextPackage.AlreadyBound;
    public void  rebind_context(org.omg.CosNaming.NameComponent[] n,
          org.omg.CosNaming.NamingContext nc )
       throws org.omg.CosNaming.NamingContextPackage.NotFound,
          org.omg.CosNaming.NamingContextPackage.CannotProceed,
          org.omg.CosNaming.NamingContextPackage.InvalidName;
    public org.omg.CORBA.Object  resolve(org.omg.CosNaming.NameComponent[] n)
       throws org.omg.CosNaming.NamingContextPackage.NotFound,
          org.omg.CosNaming.NamingContextPackage.CannotProceed,
          org.omg.CosNaming.NamingContextPackage.InvalidName;
    public void  unbind(org.omg.CosNaming.NameComponent[] n)
       throws org.omg.CosNaming.NamingContextPackage.NotFound,
          org.omg.CosNaming.NamingContextPackage.CannotProceed,
          org.omg.CosNaming.NamingContextPackage.InvalidName;
    public org.omg.CosNaming.NamingContext  new_context();
    public org.omg.CosNaming.NamingContext  bind_new_context(
          org.omg.CosNaming.NameComponent[] n)
       throws org.omg.CosNaming.NamingContextPackage.NotFound,
          org.omg.CosNaming.NamingContextPackage.CannotProceed,
          org.omg.CosNaming.NamingContextPackage.InvalidName,
          org.omg.CosNaming.NamingContextPackage.AlreadyBound;
    public void  destroy()
       throws org.omg.CosNaming.NamingContextPackage.NotEmpty;
    public void  list(int how_many, org.omg.CosNaming.BindingListHolder bl,
          org.omg.CosNaming.BindingIteratorHolder bi);
}
public interface BindingIterator  extends org.omg.CORBA.Object
{
    public boolean  next_one(org.omg.CosNaming.BindingHolder b);
    public boolean  next_n(int how_many, org.omg.CosNaming.BindingListHolder bl);
    public void  destroy();
}

public interface NamingContextExt  extends org.omg.CosNaming.NamingContext
{
    public java.lang.String  to_string(org.omg.CosNaming.NameComponent[] n)
       throws org.omg.CosNaming.NamingContextPackage.InvalidName;
    public org.omg.CosNaming.Name  to_name(java.lang.String sn)
       throws org.omg.CosNaming.NamingContextPackage.InvalidName;
    public java.lang.String  to_url(java.lang.String addrkey, java.lang.String sn)
       throws org.omg.CosNaming.NamingContextExtPackage.InvalidAddress;
          org.omg.CosNaming.NamingContextPackage.InvalidName;
    public org.omg.CORBA.Object  resolve_str(java.lang.String sn)
       throws org.omg.CosNaming.NamingContextPackage.NotFound,
          org.omg.CosNaming.NamingContextPackage.CannotProceed,
          org.omg.CosNaming.NamingContextPackage.InvalidName;
}
```

```
final public class NameComponent
{
   public java.lang.String id;
   public java.lang.String kind;
   public NameComponent( java.lang.String id, java.lang.String kind) {...}
}
```

3.16.1 Naming Context Interface

This section describes the Naming Context interface.

3.16.1.1 org.omg.CosNaming.NamingContext.bind()

Name

org.omg.CosNaming.NamingContext.bind

Format

```
public void bind(org.omg.CosNaming.NameComponent[] n,
       org.omg.CORBA.Object obj) throws
org.omg.CosNaming.NamingContextPackage.NotFound,
org.omg.CosNaming.NamingContextPackage.CannotProceed,
org.omg.CosNaming.NamingContextPackage.InvalidName,
org.omg.CosNaming.NamingContextPackage.AlreadyBound;
```

Description

Generates a name with n and an object reference with obj. Registers them in the specified naming context. If n is a compound name, it will be bound and finally registered in the naming context.

Parameters

n

name (specified as a NameComponet object's array)

obj

the object reference to bind with the specified name

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CosNaming.NamingContextPackage.NotFound

  Cannot find the naming context specified by n.

- org.omg.CosNaming.NamingContextPackage.CannotProceed

  There is no naming context.

- org.omg.CosNaming.NamingContextPackage.InvalidName

  Improper name specification

- org.omg.CosNaming.NamingContextPackage.AlreadyBound

  The specified name is already bound to the object.

- java.lang.NullPointerException

  A null is specified in the parameter.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.1.2 org.omg.CosNaming.NamingContext.rebind()

Name

org.omg.CosNaming.NamingContext.rebind

Format

```
public void rebind(org.omg.CosNaming.NameComponent[] n,
       org.omg.CORBA.Object obj) throws
org.omg.CosNaming.NamingContextPackage.NotFound,
org.omg.CosNaming.NamingContextPackage.CannotProceed,
org.omg.CosNaming.NamingContextPackage.InvalidName;
```

Description

Binds a name specified by n and an object reference specified by obj. Registers them in the specified naming context. Even if the specified name's binding already exists, no error will be generated. If n is a compound name, it will be bound and finally registered in the naming context.

Parameters

n

name (specified as a NameComponet object's array)

obj

the object reference to bind with the specified name

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CosNaming.NamingContextPackage.NotFound

  Cannot find the naming context specified by n.

- org.omg.CosNaming.NamingContextPackage.CannotProceed

  There is no naming context.

- org.omg.CosNaming.NamingContextPackage.InvalidName

  Improper name specification

- java.lang.NullPointerException

  A null is specified in the parameter.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.1.3 org.omg.CosNaming.NamingContext.bind_context()

Name

org.omg.CosNaming.NamingContext.bind_context

Format

```
public void bind_context(org.omg.CosNaming.NameComponent[] n,
       org.omg.CosNaming.NamingContext nc)throws
org.omg.CosNaming.NamingContextPackage.NotFound,
org.omg.CosNaming.NamingContextPackage.CannotProceed,
org.omg.CosNaming.NamingContextPackage.InvalidName,
org.omg.CosNaming.NamingContextPackage.AlreadyBound;
```

Description

Binds a name specified with n and a naming context object reference specified with nc. Registers it in the specified naming context. If n is a compound name, it will be bound and finally registered in the naming context.

Parameters

n

a name to bind to the naming context object (specified as a NameComponent object's array)

nc

the naming context object to register

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CosNaming.NamingContextPackage.NotFound

  Cannot find the naming context specified by n.

- org.omg.CosNaming.NamingContextPackage.CannotProceed

  There is no naming context.

- org.omg.CosNaming.NamingContextPackage.InvalidName

  Improper name specification

- org.omg.CosNaming.NamingContextPackage.AlreadyBound

  The specified name is already bound to the object.

- java.lang.NullPointerException

  A null is specified in the parameter.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

Note

Do not specify in nc the object reference to the naming service of the registration destination, or to the naming context that has already been included in the naming service of the registration destination.

3.16.1.4 org.omg.CosNaming.NamingContext.rebind_context()

Name

org.omg.CosNaming.NamingContext.rebind_context

Format

```
public void rebind_context(org.omg.CosNaming.NameComponent[] n,
       org.omg.CosNaming.NamingContext nc)throws
org.omg.CosNaming.NamingContextPackage.NotFound,
org.omg.CosNaming.NamingContextPackage.CannotProceed,
org.omg.CosNaming.NamingContextPackage.InvalidName;
```

Description

Binds a name specified with n and a naming context object reference specified with nc. Registers it in the specified naming context. Even if the specified name's binding already exists, no error will be generated. If n is a compound name, it will be bound and finally registered in the naming context.

Parameters

n

A name to bind to the naming context object (specified as a NameComponent object's array)

nc

The naming context object to register

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CosNaming.NamingContextPackage.NotFound

  Cannot find the naming context specified by n.

- org.omg.CosNaming.NamingContextPackage.CannotProceed

  There is no naming context.

- org.omg.CosNaming.NamingContextPackage.InvalidName

  Improper name specification

- java.lang.NullPointerException

  A null is specified in the parameter.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

Note

Do not specify in nc the object reference to the naming service of the registration destination, or to the naming context that has already been included in the naming service of the registration destination.

3.16.1.5 org.omg.CosNaming.NamingContext.resolve()

Name

org.omg.CosNaming.NamingContext.resolve

Format

```
public org.omg.CORBA.Object  resolve( org.omg.CosNaming.NameComponent[] n )
throws org.omg.CosNaming.NamingContextPackage.NotFound,
       org.omg.CosNaming.NamingContextPackage.CannotProceed,
       org.omg.CosNaming.NamingContextPackage.InvalidName;
```

Description

While in naming context, it returns the object reference that is bound to the name specified by n.

Parameters

n

a NameComponent object containing a set object name

Return Values

Normal termination: Notifies the object reference.

Abnormal termination: Issues the following exception:

- org.omg.CosNaming.NamingContextPackage.NotFound

  Cannot find the name specified by n.

- org.omg.CosNaming.NamingContextPackage.CannotProceed

  There is no naming context.

- org.omg.CosNaming.NamingContextPackage.InvalidName

  Improper name specification

- java.lang.NullPointerException

  A null is specified in the parameter.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.1.6 org.omg.CosNaming.NamingContext.unbind()

Name

org.omg.CosNaming.NamingContext.unbind

Format

```
public void unbind(org.omg.CosNaming.NameComponent[] n)throws
org.omg.CosNaming.NamingContextPackage.NotFound,
org.omg.CosNaming.NamingContextPackage.CannotProceed,
org.omg.CosNaming.NamingContextPackage.InvalidName;
```

Description

Unbinds the name specified by n from the naming context.

Parameters

n

name you want to unbind from the naming context (specified as a NameComponet object's array)

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CosNaming.NamingContextPackage.NotFound

  Cannot find the naming context specified by n.

- org.omg.CosNaming.NamingContextPackage.CannotProceed

  There is no naming context.

- org.omg.CosNaming.NamingContextPackage.InvalidName

  Improper name specification

- java.lang.NullPointerException

  A null is specified in the parameter.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.1.7 org.omg.CosNaming.NamingContext.new_context()

Name

org.omg.CosNaming.NamingContext.new_context

Format

```
public NamingContext new_context();
```

Description

Generates a new naming context in the naming server that controls it; it then returns the new one's naming context object reference.

Return Values

Normal termination: Returns the object reference for the new naming context.

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

Notes

- Fujitsu recommends using the org.omg.CosNaming.NamingContext.bind_new_context() method to create and register a naming context object.
- When you create a naming context object using this method, create the binding, then register it in the naming context using the org.omg.CosNaming.NamingContext.bind_context() method.

3.16.1.8 org.omg.CosNaming.NamingContext.bind_new_context()

Name

org.omg.CosNaming.NamingContext.bind_new_context

Format

```
public org.omg.CosNaming.NamingContext
bind_new_context(org.omg.CosNaming.NameComponent[] n)throws
org.omg.CosNaming.NamingContextPackage.NotFound,
org.omg.CosNaming.NamingContextPackage.AlreadyBound,
org.omg.CosNaming.NamingContextPackage.CannotProceed,
org.omg.CosNaming.NamingContextPackage.InvalidName;
```

Description

Generates a new naming context and binds its object reference with the name specified by n. Then it registers it in the relevant naming context.

The new naming context will be generated in the naming server that controls the naming context that registered its name.

Parameters

n

name you want to bind to the new naming context object reference (specified as a NameComponet object's array)

Return values

Normal termination: Returns the object reference for the new naming context. If n is a compound name, it will be bound and finally registered in the naming context.

Abnormal termination: Issues the following exception:

- org.omg.CosNaming.NamingContextPackage.NotFound

  Cannot find the naming context specified by n.

- org.omg.CosNaming.NamingContextPackage.CannotProceed

  There is no naming context.

- org.omg.CosNaming.NamingContextPackage.InvalidName

  Improper name specification

- org.omg.CosNaming.NamingContextPackage.AlreadyBound

  The specified name is already bound to the object.

- java.lang.NullPointerException

  A null is specified in the parameter.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.1.9 org.omg.CosNaming.NamingContext.destroy()

Name

org.omg.CosNaming.NamingContext.destroy

Format

```
public void  destroy()
       throws org.omg.CosNaming.NamingContextPackage.NotEmpty;
```

Description

Deletes the naming context.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CosNaming.NamingContextPackage.NotEmpty

  There is a binder in the naming context.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.1.10 org.omg.CosNaming.NamingContext.list()

Name

org.omg.CosNaming.NamingContext.list()

Format

```
public void  list( int how_many,
       org.omg.CosNaming.BindingListHolder bl,
       org.omg.CosNaming.BindingIteratorHolder bi );
```

Description

Returns binding lists made while in the naming context. The maximum number is specified by how_many. If the value specified in how_many exceeds the maximum binding number specified in the bl_how_many parameter of the nsconfig file, the Naming Service

returns the maximum binding number specified in the bl_how_many parameter. If 0 is specified in how_many, the client returns the sequence number of the binding in bi and the length in bl is set to 0.

The lists are set in org.omg.CosNaming.BindingListHolder, which is specified by bl. If the number of bindings during a naming context exceeds the value specified by how_many, an object that shows the current position will be generated in the naming context. Its object reference will return to bi. This object is called the binding iterator. The object reference returned to bi will be used when calling either org.omg.CosNaming.BindingIterator.next_one, or org.omg.CosNaming.BindingIterator.next_n.

Parameters

how_many

the number of binding lists to obtain

bl

binding lists' set area

bi

binding iterator's set area

Return Values

Normal termination: Sets the list object reference in bl.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.2 Binding Iterator Interface

This section describes the binding iterator interface.

3.16.2.1 org.omg.CosNaming.BindingIterator.next_one()

Name

org.omg.CosNaming.BindingIterator.next_one()

Format

```
public boolean  next_one( org.omg.CosNaming.BindingHolder b );
```

Description

The binding iterator shows the current position in the naming context. This takes the next binding after that and stores it in b.

Parameters

b

binding set area

Return Values

Normal termination: Returns true if one or more valid bindings are returned. However if there are no unreturned bindings, it returns a false.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.2.2 org.omg.CosNaming.BindingIterator.next_n()

Name

org.omg.CosNaming.BindingIterator.next_n()

Format

```
public boolean next_n( int how_many, org.omg.CosNaming.BindingListHolder bl );
```

Description

The binding iterator shows the current position in the naming context. This takes the next number of bindings, as specified in how_many, and stores it in bl. If the value specified in how_many exceeds the maximum binding number specified in the bl_how_many parameter in the nsconfig file, the Naming Service returns the binding number specified in the bl_how_many parameter. If 0 is specified in how_many, a BAD_PARAM system exception is issued.

Parameters

how_many

the number of binding lists to obtain

bl

binding lists' set area

Return values

Normal termination: Returns true if a valid binding is returned. However if there are no unlisted bindings, it returns a false.

Abnormal termination: Issues the following exception:

- java.lang.NullPointerException

  A null is specified in the parameter.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.2.3 org.omg.CosNaming.BindingIterator.destroy()

Name

org.omg.CosNaming.BindingIterator.destroy()

Format

```
public void destroy();
```

Description

Destroys the binding iterator.

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.3 NameComponent Class

This section describes the NameComponent class.

3.16.3.1 org.omg.CosNaming.NameComponent.NameComponent()

Name

org.omg.CosNaming.NameComponent.NameComponent()

Format

```
public NameComponent(java.lang.String id, java.lang.String kind);
```

Description

This method is a NameComponent class constructor. It stores the object reference name (String object) specified by id in a NameComponent object.

Parameters

id

the name that is either bound to, or that you want to bind to an object reference

kind

object type (specify a NULL string)

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.4 Naming Context Extension Interface

This section describes the Naming Context extension interface.

3.16.4.1 org.omg.CosNaming.NamingContextExt.to_string()

Name

org.omg.CosNaming.NamingContextExt.to_string

Format

```
public java.lang.String to_string(org.omg.CosNaming.NameComponent[] n) throws
  org.omg.CosNaming.NamingContextPackage.InvalidName;
```

Description

Converts a structural type binding name specified by n into a character string type binding name

Parameters

n

Structural type binding name

Return Values

Normal termination: Returns a string symbol binding name.

Abnormal termination: Issues the following exception:

- org.omg.CosNaming.NamingContextPackage.InvalidName

  There is an error in the specification of the name.

- java.lang.NullPointerException

  A null is specified in the parameter.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.4.2 org.omg.CosNaming.NamingContextExt.to_name()

Name

org.omg.CosNaming.NamingContextExt.to_name

Format

```
public org.omg.CosNaming.NameComponent[] to_name(java.lang.String sn) throws
  org.omg.CosNaming.NamingContextPackage.InvalidName;
```

Description

Converts a character string type binding name specified by sn into a structural type binding name

Parameters

sn

Character string type binding name

Return Values

Normal termination: Returns the structure type binding name.

Abnormal termination: Issues the following exception:

- org.omg.CosNaming.NamingContextPackage.InvalidName

  There is an error in the specification of the name.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.4.3 org.omg.CosNaming.NamingContextExt.to_url()

Name

org.omg.CosNaming.NamingContextExt.to_url

Format

```
public java.lang.String to_url(java.lang.String addrkey, java.lang.String sn) throws
  org.omg.CosNaming.NamingContextExtPackage.InvalidAddress,
  org.omg.CosNaming.NamingContextPackage.InvalidName;
```

Description

Generates a URL schema from the address specified in addrkey and the character string type binding name specified in sn.

Parameters

addrkey

Address that shows the naming context

sn

Character string type binding name

Return Values

Normal termination: Returns the created URL schema.

Abnormal termination: Issues the following exception:

- org.omg.CosNaming.NamingContextExtPackage.InvalidAddress

  There is an error in the address.

- org.omg.CosNaming.NamingContextPackage.InvalidName

  There is an error in the specification of the name.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.4.4 org.omg.CosNaming.NamingContextExt.resolve_str()

Name

org.omg.CosNaming.NamingContextExt.resolve_str

Format

```
public org.omg.CORBA.Object resolve_str(java.lang.String sn) throws
  org.omg.CosNaming.NamingContextPackage.NotFound,
  org.omg.CosNaming.NamingContextPackage.CannotProceed,
  org.omg.CosNaming.NamingContextPackage.InvalidName;
```

Description

Returns the object reference linked to the character string type binding name specified in sn in sn.

Parameters

sn

Character string type binding name

Return Values

Normal termination: Returns the object reference.

Abnormal termination: Issues the following exception:

- org.omg.CosNaming.NamingContextPackage.NotFound

  The name specified by sn was not found.

- org.omg.CosNaming.NamingContextPackage.CannotProceed

  The naming context does not exist.

- org.omg.CosNaming.NamingContextPackage.InvalidName

  There is an error in the specification of the name.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.16.4.5 org.omg.CosNaming.NamingContextExtHelper.narrow()

Name

org.omg.CosNaming.NamingContextExtHelper.narrow

Format

```
public org.omg.CosNaming.NamingContextExt
narrow( org.omg.CORBA.Object obj );
```

Description

The object reference acquired from the Naming Service is converted into the NamingContext class.

Parameters

obj

The object reference.

Return Values

For normal termination, the object reference is returned.

For abnormal termination: Issues the following exception:

- org.omg.CORBA.UserException
- org.omg.CORBA.SystemException

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information Minor Codes to be Reported from the CORBA Service in the Messages Manual.

When using the IIOP Service (Java EE client) feature, also refer to "Exception information output when Java EE is used" in the Java EE Operator's Guide.

3.17 Interface Repository Class

This section explains the methods available in the Interface Repository class.

Note

The Interface Repository class is not supported in the IIOP Service (Java EE client).

3.17.1 Type Definition

Format

```
package org.omg.CORBA;
public class DefinitionKind {                    // Object type

       public static final int dk_none       = (int)0;
       public static final int dk_all        = (int)1;

       public static final int dk_Attribute  = (int)2;
       public static final int dk_Constant   = (int)3;
       public static final int dk_Exception  = (int)4;
       public static final int dk_Interface  = (int)5;
       public static final int dk_Module     = (int)6;

       public static final int dk_Operation  = (int)7;
       public static final int dk_Typedef    = (int)8;
       public static final int dk_Alias      = (int)9;
       public static final int dk_Struct     = (int)10;

       public static final int dk_Union      = (int)11;
       public static final int dk_Enum       = (int)12;

       public static final int dk_Primitive  = (int)13;
       public static final int dk_String     = (int)14;
```

```
       public static final int dk_Sequence   = (int)15;
       public static final int dk_Array      = (int)16;

       public static final int dk_Repository = (int)17;
       public static final int dk_Wstring    = (int)18;
}
package org.omg.CORBA;
package org.omg.CORBA.ContainedPackage;
public class Description {                      // Object information class
       public org.omg.CORBA.DefinitionKind kind; // Object type

       public org.omg.CORBA.Any value;          // Object specific information
}
package org.omg.CORBA.ContainerPackage;
public class Description {                      // Object information class

       public org.omg.CORBA.Contained contained_object;  // Object
       public org.omg.CORBA.DefinitionKind kind;         // Object type
       public org.omg.CORBA.Any value;                   // Object specific
information
}
package org.omg.CORBA;
public class ModuleDescription {                // ModuleDef Information class
       public java.lang.String name;            // Identification name
       public java.lang.String id;              // Repository ID

       public java.lang.String defined_in;      // Parent project's repository ID
       public java.lang.String version;         // Version information
}
package org.omg.CORBA;
public class ConstantDescription {              // ConstantDef Information class
       public java.lang.String name;            // Identification name
       public java.lang.String id;              // Repository ID
       public java.lang.String defined_in;      // Parent project's repository ID

       public java.lang.String version;         // Version information
       public org.omg.CORBA.TypeCode type;      // Type code
       public org.omg.CORBA.Any value;          // Constant value
}
package org.omg.CORBA;
public class TypeDescription {                  // TypeDef Information class
       public java.lang.String name;            // Identification name

       public java.lang.String id;              // Repository ID
       public java.lang.String defined_in;      // Parent project's repository ID
       public java.lang.String version;         // Version information
       public org.omg.CORBA.TypeCode type;      // Type code
}
package org.omg.CORBA;
public class StructMember {                     // Structure member class
       public java.lang.String name;            // Identification name

       public org.omg.CORBA.TypeCode type;      // Type code
       public org.omg.CORBA.IDLType type_def;   // Member object reference
}
package org.omg.CORBA;
public class UnionMember {                      // UnionMember
       public java.lang.String name;            // Identification name

       public org.omg.CORBA.Any label;          // Discrimination value
       public org.omg.CORBA.TypeCode type;      // Type code
       public org.omg.CORBA.IDLType type_def;   // Member object reference
}
```

```
package org.omg.CORBA;
public class PrimitiveKind {                    // PrimitiveDef's Type
       public static final int pk_null       = (int)0;
       public static final int pk_void       = (int)1;

       public static final int pk_short      = (int)2;
       public static final int pk_long       = (int)3;
       public static final int pk_ushort     = (int)4;
       public static final int pk_ulong      = (int)5;

       public static final int pk_float      = (int)6;
       public static final int pk_double     = (int)7;
       public static final int pk_boolean    = (int)8;
       public static final int pk_char       = (int)9;

       public static final int pk_octet      = (int)10;
       public static final int pk_any        = (int)11;
       public static final int pk_TypeCode   = (int)12;
       public static final int pk_Principal  = (int)13;
       public static final int pk_string     = (int)14;
       public static final int pk_objref     = (int)15;

       public static final int pk_longlong   = (int)16;
       public static final int pk_ulonglong  = (int)17;

       public static final int pk_longdouble = (int)18;
       public static final int pk_wchar      = (int)19;
       public static final int pk_wstring    = (int)20;
}
package org.omg.CORBA;
public class ExceptionDescription {             // ExceptionDef Information class
       public java.lang.String name;            // Identification name

       public java.lang.String id;              // Repository ID
       public java.lang.String defined_in;      // Parent project's repository ID
       public java.lang.String version;         // Version
       public org.omg.CORBA.TypeCode type;      // Type code
}
package org.omg.CORBA;
public class AttributeMode {                    // Attribute's attribute
       public static final int ATTR_NORMAL   = (int)0;
       public static final int ATTR_READONLY = (int)1;
}
package org.omg.CORBA;
public class AttributeDescription {             // AttributeDef Information class
       public java.lang.String name;            // Identification name
       public java.lang.String id;              // Repository ID

       public java.lang.String defined_in;      // Parent project's repository ID
       public java.lang.String version;         // Version
       public org.omg.CORBA.TypeCode type;      // Type code
       public org.omg.CORBA.AttributeMode mode; // Attribute
}
package org.omg.CORBA;
public class ParameterMode {                    // Parameter's attribute type
       public static final int PARAM_IN = (int)0;

       public static final int PARAM_OUT = (int)1;
       public static final int PARAM_INOUT = (int)2;
}
package org.omg.CORBA;
public class ParameterDescription {             // Parameter information class
       public java.lang.String name;            // Identification name
```

```
       public org.omg.CORBA.TypeCode type;       // Type code
       public org.omg.CORBA.IDLType type_def;    // Member object reference
       public org.omg.CORBA.ParameterMode mode;  // Attribute
}
 package org.omg.CORBA;
public class OperationMode {                     // Operation's attribute type
       public static final int OP_NORMAL = (int)0;
       public static final int OP_ONEWAY = (int)1;
}
package org.omg.CORBA;
public class OperationDescription {              // OperationDef Information class
       public java.lang.String name;             // Identification name
       public java.lang.String id;               // Repository ID

       public java.lang.String defined_in;       // Parent project's repository ID
       public java.lang.String version;          // Version information
       public org.omg.CORBA.TypeCode result;     // Return value type code

       public org.omg.CORBA. OperationMode mode; // Attribute
       public java.lang.String[] contexts;       // Context

public org.omg.CORBA.ParameterDescription[] parameters;
                                                 // Parameter information
       public org.omg.CORBA.ExceptionDescription[] exceptions;
                                                 // Exception information
}
package org.omg.CORBA.InterfaceDefPackage;
public class FullInterfaceDescription {          // FullInterface Information class
       public java.lang.String name;             // Identification name
       public java.lang.String id;                 // Repository ID
       public java.lang.String defined_in;         // Parent project's repository ID

       public java.lang.String version;            // Version information
       public org.omg.CORBA.OperationDescription[] operations;   // Operation
                                                                    information
       public org.omg.CORBA.AttributeDescription[] attributes;   // Attribute
                                                                    information

       public java.lang.String[] base_interfaces;  // Inheritance interface
information
       public org.omg.CORBA.TypeCode type;         // Type code
       public boolean  is_abstract;                // Abstract information
}
package org.omg.CORBA;
public class InterfaceDescription {                // InterfaceDef Information class
       public java.lang.String name;               // Identification name
       public java.lang.String id;                 // Repository ID
       public java.lang.String defined_in;         // Parent project's repository ID

       public java.lang.String version;            // Version information
       public java.lang.String base_interfaces[];  // Inheritance interface
                                                      information
       public boolean is_abstract;                 // Abstract information
}
```

3.17.2 IRObject Common Interface

The methods described in this section are inherited by each Interface Repository object in the repository, and can be used as such.

3.17.2.1 org.omg.CORBA.IRObject.def_kind()

Name

org.omg.CORBA.IRObject.def_kind

Format

```
public interface  IRObject extends IRObjectOperations,
       org.omg.CORBA.portable.IDLEntity, org.omg.CORBA.Object
{
       public org.omg.CORBA.DefinitionKind def_kind();
}
```

Description

Returns the interface type of the interface repository object.

Return Values

Normal termination: Returns the interface type of the interface repository object.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.3 Contained Common Interface

The methods described in this section are inherited by each interface repository object that is contained in other repository objects, and can be used as such.

3.17.3.1 org.omg.CORBA.Contained.id()

Name

org.omg.CORBA.Contained.id

Format

```
public interface Contained extends org.omg.CORBA.IRObject,
       ContainedOperations, org.omg.CORBA.portable.IDLEntity,
       org.omg.CORBA.Object
{
       public java.lang.String id();
}
```

Description

Returns the repository ID of the interface repository object.

Return Values

Normal termination: Returns the repository ID.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.3.2 org.omg.CORBA.Contained.name()

Name

org.omg.CORBA.Contained.name

Format

```
public interface Contained extends org.omg.CORBA.IRObject,
       ContainedOperations, org.omg.CORBA.portable.IDLEntity,
       org.omg.CORBA.Object
{
       public java.lang.String name();
}
```

Description

Returns the interface repository object's name.

Return Values

Normal termination: Returns names (org.omg.CORBA.Identifier type).

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.3.3 org.omg.CORBA.Contained.defined_in()

Name

org.omg.CORBA.Contained.defined_in

Format

```
public interface Contained extends org.omg.CORBA.IRObject,
       ContainedOperations, org.omg.CORBA.portable.IDLEntity,
       org.omg.CORBA.Object
{
       public org.omg.CORBA.Container defined_in();
}
```

Description

Returns the object reference for the interface repository object's container object.

Return Values

Normal termination: Returns the object reference.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.3.4 org.omg.CORBA.Contained.describe()

Name

org.omg.CORBA.Contained.describe

Format

```
public interface Contained extends org.omg.CORBA.IRObject,
       ContainedOperations, org.omg.CORBA.portable.IDLEntity,
       org.omg.CORBA.Object
{
```

```
        public org.omg.CORBA.ContainedPackage.Description describe();
}
```

Description

Returns the org.omg.CORBA.ContainedPackage.Description for interface repository object information. This information will vary according to the interface repository object. Information specific to each repository object is stored in org.omg.CORBA.ContainedPackage.Description's value member. This value is a CORBA.any type construct. Refer to the table below for a list of interface repository object attributes and the classes that they set. For details on each construct, refer to 3.17.1 Type Definition.

Module	Class
org.omg.CORBA.ModuleDef object	ModuleDescription class
org.omg.CORBA.ConstantDef object	ConstantDescription class
org.omg.CORBA.StructDef object	TypeDescription class
org.omg.CORBA.UnionDef object	TypeDescription class
org.omg.CORBA.EnumDef object	TypeDescription class
org.omg.CORBA.AliasDef object	TypeDescription class
org.omg.CORBA.ExceptionDef object	ExceptionDescription class
org.omg.CORBA.AttributeDef object	AttributeDescription class
org.omg.CORBA.OperationDef object	OperationDescription class
org.omg.CORBA.InterfaceDef object	InterfaceDescription class

Return Values

Normal termination: Returns org.omg.CORBA.ContainedPackage.Description.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.3.5 Methods you can use through Inheritance

(1) org.omg.CORBA.Contained.def_kind()

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.Contained".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

3.17.4 Container Common Interface

The methods explained in this section are inherited by interface repository objects that contain other repository objects (containers), and can be used as such.

3.17.4.1 org.omg.CORBA.Container.lookup()

Name

org.omg.CORBA.Container.lookup

Format

```
public interface Container extends org.omg.CORBA.IRObject,
       ContainerOperations, org.omg.CORBA.portable.IDLEntity,
```

```
       org.omg.CORBA.Object
{
       public org.omg.CORBA.Contained lookup(java.lang.String search_name);
}
```

Description

It searches in the container for objects with specified names, and returns their references. If the object to be returned cannot be found, object reference NIL (null) is returned and operation ends normally.

Parameters

search_name

The name you are looking for

Return Values

Normal termination: Returns an object reference for the object with the specified name.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.4.2 org.omg.CORBA.Container.contents()

Name

org.omg.CORBA.Container.contents

Format

```
public interface Container extends org.omg.CORBA.IRObject,
       ContainerOperations, org.omg.CORBA.portable.IDLEntity,
       org.omg.CORBA.Object
{
       public org.omg.CORBA.Contained[] contents(
                     org.omg.CORBA.DefinitionKind limit_type,
                     boolean exclude_inherited);
}
```

Description

Returns a list of object references of directly contained interface repository objects and those it contains through inheritance.

If the object to be returned cannot be found, 0 is set to length of the return list and the object reference becomes undefined.

Parameters

limit_type

The contained Inclusion object of found interface type

When dk_all is specified for this parameter, and FALSE is specified for exclude_inherited, all contained and inherited object reference are targeted.

exclude_inherited

TRUE:

The inherited object is targeted in the search.

FALSE:

The inherited object is not targeted in the search.

Return Values

Normal termination: Returns a list of found object's object references.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.4.3 org.omg.CORBA.Container.lookup_name()

Name

org.omg.CORBA.Container.lookup_name

Format

```
public interface Container extends org.omg.CORBA.IRObject,
       ContainerOperations, org.omg.CORBA.portable.IDLEntity,
       org.omg.CORBA.Object
{
       public org.omg.CORBA.Contained[] lookup_name(
              java.lang.String search_name,
              int levels_to_search,
              org.omg.CORBA.DefinitionKind limit_type,
              boolean exclude_inherited);
}
```

Description

Returns a list of object references both of directly contained interface repository objects and those contained through inheritance. In addition it returns a list of the references that they contain or inherit.

If the object to be returned cannot be found, 0 is set to length of the return list and the object reference becomes undefined.

Parameters

search_name

The search key name (Identifier)

levels_to_search

The hierarchical depth of the search. -1 will cause a search on all levels; 1 will only search the objects immediately below the specified object.

limit_type

The contained Inclusion object of found interface type

When dk_all is specified for this parameter, and FALSE is specified for exclude_inherited, all contained and inherited object reference are targeted.

exclude_inherited

TRUE:

The inherited object is targeted in the search.

FALSE:

The inherited object is not targeted in the search.

Return Values

Normal termination: Returns a list of found object's object references.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.4.4 org.omg.CORBA.Container.describe_contents()

Name

org.omg.CORBA.Container.describe_contents

Format

```
public interface Container extends org.omg.CORBA.IRObject,
       ContainerOperations, org.omg.CORBA.portable.IDLEntity,
       org.omg.CORBA.Object
{
       public org.omg.CORBA.ContainerPackage.Description[]
              describe_contents(
                     org.omg.CORBA.DefinitionKind limit_type,
                     boolean exclude_inherited,
                     int max_returned_objs);
}
```

Description

Returns the object definition information of directly contained interface repository objects and those contained through inheritance, as a Description construct list (refer to 3.17.1 Type Definition).

If the object to be returned cannot be found, 0 is set to length of the return list and the object reference becomes undefined.

Parameters

limit_type

The contained Inclusion object of found interface type

When dk_all is specified for this parameter, and FALSE is specified for exclude_inherited, all contained and inherited object reference are targeted. However, the number will be limited by max_returned_objs; to get a complete list, set max_returned_objs to -1.

exclude_inherited

TRUE:

The inherited object is targeted in the search.

FALSE:

The inherited object is not targeted in the search.

max_returned_objs

The number of definition information

Return Values

Normal termination: Returns an object definition information list.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual

3.17.4.5 Methods you can use through Inheritance

(1) org.omg.CORBA.Container.def_kind

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.Container".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

3.17.5 IDL Type Common Interface

This section describes the IDL Type common interface.

3.17.5.1 org.omg.CORBA.IDLType.type()

Name

org.omg.CORBA.IDLType.type

Format

```
public interface IDLType extends org.omg.CORBA.IRObject,
       IDLTypeOperations, org.omg.CORBA.portable.IDLEntity,
       org.omg.CORBA.Object
{
       public org.omg.CORBA.TypeCode type();
}
```

Description

Returns the IDL object's type code.

Return Values

Normal termination: Returns the type code of the IDLType object.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.5.2 Methods you can use through Inheritance

(1) org.omg.CORBA.IDLType.def_kind

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.IDLType".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

3.17.6 Repository Interface

This section describes the Interface Repository interface.

3.17.6.1 org.omg.CORBA.Repository.lookup_id()

Name

org.omg.CORBA.Repository.lookup_id

Format

```
public interface Repository extends CORBA.Container, RepositoryOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.Contained lookup_id(java.lang.String search_id);
}
```

Description

Looks for an object with a repository ID specified by search_id and returns its object reference. If the object to be returned cannot be found, object reference NIL (null) is returned and operation ends normally.

Parameters

search_id

The object with a repository ID

Return Values

Normal termination: Returns found object's object references.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.6.2 Methods you can use through Inheritance

(1) org.omg.CORBA.Repository.def_kind

(2) org.omg.CORBA.Repository.lookup

(3) org.omg.CORBA.Repository.contents

(4) org.omg.CORBA.Repository.lookup_name

(5) org.omg.CORBA.Repository.describe_contents

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.Repository".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2) to (5), "3.17.4 Container Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Container" with "org.omg.CORBA.Repository".

For details on the function specifications, refer to "3.17.4 Container Common Interface".

3.17.7 ModuleDef Interface

This section describes the ModuleDef interface.

3.17.7.1 Methods you can use through Inheritance

(1) org.omg.CORBA.ModuleDef.def_kind

(2) org.omg.CORBA.ModuleDef.id

(3) org.omg.CORBA.ModuleDef.name

(4) org.omg.CORBA.ModuleDef.defined_in

(5) org.omg.CORBA.ModuleDef.describe

(6) org.omg.CORBA.ModuleDef.lookup

(7) org.omg.CORBA.ModuleDef.contents

(8) org.omg.CORBA.ModuleDef.lookup_name

(9) org.omg.CORBA.ModuleDef.describe_contents

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.ModuleDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2) to (5), "3.17.3 Contained Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Contained" with "org.omg.CORBA.ModuleDef".

For details on the function specifications, refer to "3.17.3 Contained Common Interface".

For (6) to (9), "3.17.4 Container Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Container" with "org.omg.CORBA.ModuleDef".

For details on the function specifications, refer to "3.17.4 Container Common Interface".

3.17.8 ConstantDef Interface

This section describes the ConstantDef interface.

3.17.8.1 org.omg.CORBA.ConstantDef.type()

Name

org.omg.CORBA.ConstantDef.type()

Format

```
public interface ConstantDef extends CORBA.Contained,
        ConstantDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.TypeCode type();
}
```

Description

Returns the constant definition object's TypeCode.

Return Values

Normal termination: Returns the TypeCode.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.8.2 org.omg.CORBA.ConstantDef.value()

Name

org.omg.CORBA.ConstantDef.value

Format

```
public interface ConstantDef extends CORBA.Contained,
        ConstantDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.Any value();
}
```

Description

Returns the ConstantDef object's constant value in any type.

Return Values

Normal termination: Returns constant values (org.omg.CORBA.Any type).

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.8.3 Methods you can use through Inheritance

(1) org.omg.CORBA.ConstantDef.def_kind

(2) org.omg.CORBA.ConstantDef.id

(3) org.omg.CORBA.ConstantDef.name

(4) org.omg.CORBA.ConstantDef.defined_in

(5) org.omg.CORBA.ConstantDef.describe

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.ConstantDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2) to (5), "3.17.3 Contained Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Contained" with "org.omg.CORBA.ConstantDef".

For details on the function specifications, refer to "3.17.3 Contained Common Interface".

3.17.9 StructDef Interface

This section describes the StructDef interface.

3.17.9.1 org.omg.CORBA.StructDef.members()

Name

org.omg.CORBA.StructDef.members

Format

```
public interface StructDef extends org.omg.CORBA.TypedefDef,
        StructDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.StructMember[] members();
}
```

Description

Returns StructDef object member information in StructMember list format (refer to 3.17.1 Type Definition).

Return Values

Normal termination: Returns StructDef object member information (StructMember[]).

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.9.2 Methods you can use through Inheritance

(1) org.omg.CORBA.StructDef.def_kind

(2) org.omg.CORBA.StructDef.id

(3) org.omg.CORBA.StructDef.name

(4) org.omg.CORBA.StructDef.defined_in

(5) org.omg.CORBA.StructDef.describe

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.StructDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2) to (5), "3.17.3 Contained Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Contained" with "org.omg.CORBA.StructDef".

For details on the function specifications, refer to "3.17.3 Contained Common Interface".

3.17.10 UnionDef Interface

This section describes the UnionDef interface.

3.17.10.1 org.omg.CORBA.UnionDef.discriminator_type()

Name

org.omg.CORBA.UnionDef.discriminator_type

Format

```
public interface UnionDef extends org.omg.CORBA.TypedefDef,
        UnionDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.TypeCode discriminator_type();
}
```

Description

Returns the UnionDef object's discrimination information definition TypeCode.

Return Values

Normal termination: Returns the discrimination information definition's TypeCode.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.10.2 org.omg.CORBA.UnionDef.members()

Name

org.omg.CORBA.UnionDef.members

Format

```
public interface UnionDef extends org.omg.CORBA.TypedefDef,
        UnionDefOperations,
```

```
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.UnionMember[] members();
}
```

Description

Returns the UnionDef object's member information in UnionMember list format (refer to 3.17.1 Type Definition).

Return Values

Normal termination: Returns the UnionDef object's member information (UnionMember[]).

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.10.3 Methods you can use through Inheritance

(1) org.omg.CORBA.UnionDef.def_kind

(2) org.omg.CORBA.UnionDef.id

(3) org.omg.CORBA.UnionDef.name

(4) org.omg.CORBA.UnionDef.defined_in

(5) org.omg.CORBA.UnionDef.describe

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.UnionDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2) to (5), "3.17.3 Contained Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Contained" with "org.omg.CORBA.UnionDef".

For details on the function specifications, refer to "3.17.3 Contained Common Interface".

3.17.11 EnumDef Interface

This section describes the EnumDef interface.

3.17.11.1 org.omg.CORBA.EnumDef.members()

Name

org.omg.CORBA.EnumDef.members

Format

```
public interface EnumDef extends org.omg.CORBA.TypedefDef,
        EnumDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public java.lang.String[] members();
}
```

Description

Returns the EnumDef object's member information in String list format.

Return Values

Normal termination: Returns the EnumDef object's member information.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.11.2 Methods you can use through Inheritance

(1) org.omg.CORBA.EnumDef.def_kind

(2) org.omg.CORBA.EnumDef.id

(3) org.omg.CORBA.EnumDef.name

(4) org.omg.CORBA.EnumDef.defined_in

(5) org.omg.CORBA.EnumDef.describe

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.EnumDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2) to (5), "3.17.3 Contained Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Contained" with "org.omg.CORBA.EnumDef".

For details on the function specifications, refer to "3.17.3 Contained Common Interface".

3.17.12 AliasDef Interface

This section describes the AliasDef interface.

3.17.12.1 org.omg.CORBA.AliasDef.original_type_def()

Name

org.omg.CORBA.AliasDef.original_type_def

Format

```
public interface AliasDef extends org.omg.CORBA.TypedefDef,
        AliasDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.IDLType original_type_def();
}
```

Description

Returns the AliasDef object's source data type's object reference.

Return Values

Normal termination: Returns the AliasDef object's source data type's object reference.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.12.2 Methods you can use through Inheritance

(1) org.omg.CORBA.AliasDef.def_kind

(2) org.omg.CORBA.AliasDef.id

(3) org.omg.CORBA.AliasDef.name

(4) org.omg.CORBA.AliasDef.defined_in

(5) org.omg.CORBA.AliasDef.describe

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.AliasDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2) to (5), "3.17.3 Contained Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Contained" with "org.omg.CORBA.AliasDef".

For details on the function specifications, refer to "3.17.3 Contained Common Interface".

3.17.13 StringDef Interface

This section describes the StringDef interface.

3.17.13.1 org.omg.CORBA.StringDef.bound()

Name

org.omg.CORBA.StringDef.bound

Format

```
public interface StringDef extends org.omg.CORBA.IDLType,
        StringDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public int bound();
}
```

Description

Returns the StringDef object's maximum string length.

Return Values

Normal termination: Returns the maximum string length.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.13.2 Methods you can use through Inheritance

(1) org.omg.CORBA.StringDef.def_kind

(2) org.omg.CORBA.StringDef.type

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.StringDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2), "3.17.5 IDL Type Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IDLType" with "org.omg.CORBA.StringDef".

For details on the function specifications, refer to "3.17.5 IDL Type Common Interface".

3.17.14 SequenceDef Interface

This section describes the SequenceDef interface.

3.17.14.1 org.omg.CORBA.SequenceDef.bound()

Name

org.omg.CORBA.SequenceDef.bound

Format

```
public interface SequenceDef extends org.omg.CORBA.IDLType,
        SequenceDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public int bound();
}
```

Description

Returns the maximum number of sequence elements, as defined by the SequenceDef object.

Return Values

Normal termination: Returns the maximum number of sequence elements.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.14.2 org.omg.CORBA.SequenceDef.element_type()

Name

org.omg.CORBA.SequenceDef.element_type

Format

```
public interface SequenceDef extends org.omg.CORBA.IDLType,
        SequenceDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.TypeCode element_type();
}
```

Description

Returns TypeCode, which shows the sequence elements' type, as defined by a SequenceDef object.

Return Values

Normal termination: Returns TypeCode, which shows the sequence elements' type.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.14.3 Methods you can use through Inheritance

(1) org.omg.CORBA.SequenceDef.def_kind

(2) org.omg.CORBA.SequenceDef.type

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.SequenceDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2), "3.17.5 IDL Type Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IDLType" with "org.omg.CORBA.SequenceDef".

For details on the function specifications, refer to "3.17.5 IDL Type Common Interface".

3.17.15 ArrayDef Interface

This section describes the ArrayDef interface.

3.17.15.1 org.omg.CORBA.ArrayDef.length()

Name

org.omg.CORBA.ArrayDef.length

Format

```
public interface ArrayDef extends org.omg.CORBA.IDLType,
        ArrayDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public int length();
}
```

Description

Returns the number of array elements contained in an ArrayDef object.

Return Values

Normal termination: Returns the number of array elements.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.15.2 org.omg.CORBA.ArrayDef.element_type()

Name

org.omg.CORBA.ArrayDef.element_type

Format

```
public interface ArrayDef extends org.omg.CORBA.IDLType,
        ArrayDefOperations,
        org.omg.CORBA.portable.IDLEntity,
```

```
        org.omg.CORBA.Object
{
       public org.omg.CORBA.TypeCode element_type();
}
```

Description

Returns TypeCode, which shows the type of the array elements in the ArrayDef object.

Return Values

Normal termination: Returns TypeCode, which shows the array elements' type.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.15.3 Methods you can use through Inheritance

(1) org.omg.CORBA.ArrayDef.def_kind

(2) org.omg.CORBA.ArrayDef.type

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.ArrayDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2), "3.17.5 IDL Type Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IDLType" with "org.omg.CORBA.ArrayDef".

For details on the function specifications, refer to "3.17.5 IDL Type Common Interface".

3.17.16 WstringDef Interface

This section describes the WstringDef interface.

3.17.16.1 org.omg.CORBA.WstringDef.bound()

Name

org.omg.CORBA.WstringDef.bound

Format

```
public interface WstringDef extends org.omg.CORBA.IDLType,
       WstringDefOperations,
       org.omg.CORBA.portable.IDLEntity,
       org.omg.CORBA.Object
{
       public int bound();
}
```

Description

Returns the WstringDef object's maximum string length.

Return Values

Normal termination: Returns the maximum string length.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.16.2 Methods you can use through Inheritance

(1) org.omg.CORBA.WstringDef.def_kind

(2) org.omg.CORBA.WstringDef.type

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.WstringDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2), "3.17.5 IDL Type Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IDLType" with "org.omg.CORBA.WstringDef".

For details on the function specifications, refer to "3.17.5 IDL Type Common Interface".

3.17.17 InterfaceDef Interface

This section describes the InterfaceDef interface.

3.17.17.1 org.omg.CORBA.InterfaceDef.describe_interface()

Name

org.omg.CORBA.InterfaceDef.describe_interface

Format

```
public interface InterfaceDef extends org.omg.CORBA.Container,
       org.omg.CORBA.Contained,
       org.omg.CORBA.IDLType,
       InterfaceDefOperations,
       org.omg.CORBA.portable.IDLEntity,
       org.omg.CORBA.Object
{
     public org.omg.CORBA.InterfaceDefPackage.FullInterfaceDescription describe_interface();
}
```

Description

Returns the InterfaceDef object's definition information, including inheritance related information. This is called the FullInterfaceDescription (refer to 3.17.1 Type Definition).

Return Values

Normal termination: Returns the InterfaceDef object's definition information (FullInterfaceDescription).

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.17.2 Methods you can use through Inheritance

(1) org.omg.CORBA.InterfaceDef.def_kind

(2) org.omg.CORBA.InterfaceDef.id

(3) org.omg.CORBA.InterfaceDef.name

(4) org.omg.CORBA.InterfaceDef.defined_in

(5) org.omg.CORBA.InterfaceDef.describe

(6) org.omg.CORBA.InterfaceDef.lookup

(7) org.omg.CORBA.InterfaceDef.contents

(8) org.omg.CORBA.InterfaceDef.lookup_name

(9) org.omg.CORBA.InterfaceDef.describe_contents

(10) org.omg.CORBA.InterfaceDef.type

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.InterfaceDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2) to (5), "3.17.3 Contained Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Contained" with "org.omg.CORBA.InterfaceDef".

For details on the function specifications, refer to "3.17.3 Contained Common Interface".

For (6) to (9), "3.17.4 Container Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Container" with "org.omg.CORBA.InterfaceDef".

For details on the function specifications, refer to "3.17.4 Container Common Interface".

For (10), "3.17.5 IDL Type Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IDLType" with "org.omg.CORBA.InterfaceDef".

For details on the function specifications, refer to "3.17.5 IDL Type Common Interface".

3.17.18 OperationDef Interface

This section describes the OperationDef interface.

3.17.18.1 org.omg.CORBA.OperationDef.result()

Name

org.omg.CORBA.OperationDef.result

Format

```
public interface OperationDef extends org.omg.CORBA.Contained,
        OperationDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.TypeCode result();
}
```

Description

Returns TypeCode, defined by the OperationDef object, which shows the operation's return value type.

Return Values

Normal termination: Returns TypeCode which shows the operation's return type.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.18.2 org.omg.CORBA.OperationDef.params()

Name

org.omg.CORBA.OperationDef.params

Format

```
public interface OperationDef extends org.omg.CORBA.Contained,
        OperationDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.ParameterDescription[] params();
}
```

Description

Returns a list of operation parameter information as defined by the OperationDef object. The list is in org.omg.CORBA.ParameterDescription format (refer to 3.17.1 Type Definition).

Return Values

Normal termination: Returns operation parameter information (org.omg.CORBA.ParameterDescription[]).

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.18.3 org.omg.CORBA.OperationDef.contexts()

Name

org.omg.CORBA.OperationDef.contexts

Format

```
public interface OperationDef extends org.omg.CORBA.Contained,
        OperationDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public java.lang.String[] contexts();
}
```

Description

Returns context identifiers for operations as defined by the OperationDef object. The identifiers are in list form (refer to 3.17.1 Type Definition).

Return Values

Normal termination: Returns the list of identifiers (String[]).

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.18.4 org.omg.CORBA.OperationDef.exceptions()

Name

org.omg.CORBA.OperationDef.exceptions

Format

```
public interface OperationDef extends org.omg.CORBA.Contained,
        OperationDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.ExceptionDef[] exceptions();
}
```

Description

Returns a list containing definition information for operation exception events as defined by the OperationDef object. For list format refer to 3.17.1 Type Definition.

Return Values

Normal termination: Returns a list containing definition information for operation exception events (org.omg.CORBA.ExceptionDef[]).

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.18.5 Methods you can use through Inheritance

(1) org.omg.CORBA.OperationDef.def_kind

(2) org.omg.CORBA.OperationDef.id

(3) org.omg.CORBA.OperationDef.name

(4) org.omg.CORBA.OperationDef.defined_in

(5) org.omg.CORBA.OperationDef.describe

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.OperationDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2) to (5), "3.17.3 Contained Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Contained" with "org.omg.CORBA.OperationDef".

For details on the function specifications, refer to "3.17.3 Contained Common Interface".

3.17.19 AttributeDef Interface

This section describes the AttributeDef interface.

3.17.19.1 org.omg.CORBA.AttributeDef.type()

Name

org.omg.CORBA.AttributeDef.type

Format

```
public interface AttributeDef extends org.omg.CORBA.Contained,
        AttributeDefOperations,
        org.omg.CORBA.portable.IDLEntity,
```

```
        org.omg.CORBA.Object
{
       public org.omg.CORBA.TypeCode type();
}
```

Description

Returns the AttributeDef object's TypeCode.

Return Values

Normal termination: Returns the AttributeDef object's TypeCode.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.19.2 Methods you can use through Inheritance

(1) org.omg.CORBA.AttributeDef.def_kind

(2) org.omg.CORBA.AttributeDef.id

(3) org.omg.CORBA.AttributeDef.name

(4) org.omg.CORBA.AttributeDef.defined_in

(5) org.omg.CORBA.AttributeDef.describe

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.AttributeDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2) to (5), "3.17.3 Contained Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Contained" with "org.omg.CORBA.AttributeDef".

For details on the function specifications, refer to "3.17.3 Contained Common Interface".

3.17.20 ExceptionDef Interface

This section describes the ExceptionDef interface.

3.17.20.1 org.omg.CORBA.ExceptionDef.type()

Name

org.omg.CORBA.ExceptionDef.type

Format

```
public interface ExceptionDef extends org.omg.CORBA.Contained,
        ExceptionDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.TypeCode type();
}
```

Description

Returns the ExceptionDef object's TypeCode.

Return Values

Normal termination: Returns the ExceptionDef object's TypeCode.

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.20.2 org.omg.CORBA.ExceptionDef.members()

Name

org.omg.CORBA.ExceptionDef.members

Format

```
public interface ExceptionDef extends org.omg.CORBA.Contained,
        ExceptionDefOperations,
        org.omg.CORBA.portable.IDLEntity,
        org.omg.CORBA.Object
{
       public org.omg.CORBA.StructMember[] members();
}
```

Description

Returns a list of definition information for exception events (org.omg.CORBA. StructMember[]) defined by the ExceptionDef object.

Return Values

Normal termination: Returns definition information for exception events (StructMember[]).

Abnormal termination: Issues the org.omg.CORBA.SystemException.

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.17.20.3 Methods you can use through Inheritance

(1) org.omg.CORBA.ExceptionDef.def_kind

(2) org.omg.CORBA.ExceptionDef.id

(3) org.omg.CORBA.ExceptionDef.name

(4) org.omg.CORBA.ExceptionDef.defined_in

(5) org.omg.CORBA.ExceptionDef.describe

For (1), "3.17.2 IRObject Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.IRObject" with "org.omg.CORBA.ExceptionDef".

For details on the function specifications, refer to "3.17.2 IRObject Common Interface".

For (2) to (5), "3.17.3 Contained Common Interface" is inherited.

It is possible to replace portions of "org.omg.CORBA.Contained" with "org.omg.CORBA.ExceptionDef".

For details on the function specifications, refer to "3.17.3 Contained Common Interface".

3.18 Other Interfaces

Here we will explain other interfaces not prescribed in CORBA.

Note

The other interfaces described in this section are not supported in the IIOP Service (Java EE client).

3.18.1 Instance Release Interface

This section explains the POAdisconnect class that handles the instance release interface.

To use this function you must use the CORBA service instance control function. In addition POA must use the ActiveObjectMap for instance control.

```
//Java
package com.fujitsu.ObjectDirector.PortableServer;
public class POAdisconnect
{
    static public void setDisconnect( POAdisconnect disconnect )
          throws org.omg.CORBA.BAD_OPERATION {
    }
    static public void resetDisconnect( POAdisconnect disconnect )
          throws org.omg.CORBA.BAD_OPERATION {
    }
    static public void resetDisconnect()
          throws org.omg.CORBA.BAD_OPERATION {
    }
    public boolean release_instance(
        org.omg.PortableServer.POA     poa,
        org.omg.PortableServer.Servant servant,
        byte[]                          oid ) {
    }
}
```

3.18.1.1 com.fujitsu.ObjectDirector.PortableServer.POAdisconnect.setDisconnect()

Name

com.fujitsu.ObjectDirector.PortableServer.POAdisconnect.setDisconnect

Format

```
public static void setDisconnect(
       com.fujitsu.ObjectDirector.PortableServer.POAdisconnect disconnect );
```

Description

Registers an instance of a class that will release instances when a client application terminates. Classes that invoke the instance release must inherit this one.

Parameters

disconnect

The class instance you want to register

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.BAD_PARAM

  A null is specified in the parameter.

  Multiple registration.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.18.1.2 com.fujitsu.ObjectDirector.PortableServer.POAdisconnect.resetDisconnect ()

Name

com.fujitsu.ObjectDirector.PortableServer.POAdisconnect.resetDisconnect

Format

```
(1) public static void resetDisconnect();
(2) public static void resetDisconnect(
        com.fujitsu.ObjectDirector.PortableServer.POAdisconnect disconnect );
```

Description

(1) Deletes any previously registered instances of the instance release process class. If there are no previously registered instances this method will fail.

(2) Using the specified parameters, it overwrites any previously registered instances of the instance release process class.

If there is a null in the parameter, and if there are no previously registered instances this method will fail.

Parameters

(1) None

(2)

disconnect

the class instance you want to change

Return Values

Normal termination: None

Abnormal termination: Issues the following exception:

- org.omg.CORBA.BAD_PARAM

  There is no previously registered instance.

  In the case of (2), the parameter specifies a null.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.18.1.3 com.fujitsu.ObjectDirector.PortableServer.POAdisconnect.release_instance()

Name

com.fujitsu.ObjectDirector.PortableServer.POAdisconnect.release_instance

Format

```
public boolean release_instance(
      org.omg.PortableServer.POA     poa,
```

```
    org.omg.PortableServer.Servant servant,
    byte[]                         oid );
```

Description

This is called when a client application terminates. When invoking instance release, it prepares a user definition class that inherits this one and overrides this method. Judges if a specified Servant instance or an object ID based instance is needed. Returns a true to release and a false to not release.

Super class release_instance() methods will always return a false.

Parameters

poa

Servant's POA.

servant

Servant's instance

oid

Servant's object ID

Return Values

false (default): Does not release the instance.

3.18.2 Communication Resource Release Interface

This section describes the communication resource release interface.

3.18.2.1 com.fujitsu.ObjectDirector.CORBA.ORB.net_disconnect()

Name

com.fujitsu.ObjectDirector.CORBA.ORB.net_disconnect

Format

```
public void net_disconnect( org.omg.CORBA.Object obj )
throws org.omg.CORBA.INV_OBJREF,
             org.omg.CORBA.COMM_FAILURE;
```

Description

Releases the communication resources being used by the object specified at obj.

Parameters

obj

Specify the ObjectReference to be disconnected.

Return Values

Normal termination: None

Abnormal termination: Issues the following system exception:

- org.omg.CORBA.INV_OBJREF

  Invalid object specified.

  The connection has already been disconnected.

- org.omg.CORBA.COMM_FAILURE

  There has been a communications failure.

For details on the minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Usage

The ORB class instance obtained by the ORB.init() method is cast in the com.fujitsu.ObjectDirector.CORBA.ORB form, then this method is called.

However, do not specify the com.fujitsu.ObjectDirector.CORBA package in the import declaration.

Coding example

```
 org.omg.CORBA.ORB orb=org.omg.CORBA.ORB.init();
 ((com.fujitsu.ObjectDirector.CORBA.ORB)orb).net_disconnect(obj);
```

Notes

- The following errors may be posted when this function is issued, depending on the application client status:

  If an application operating in thread mode is communicating with the server in a thread, other than the one where this function is issued, the thread communicating with the server is notified of org.omg.CORBA.COMM_FAILURE.

  If asynchronous requests are issued, asynchronous request results are not received.

  If this function is issued to the same object, org.omg.CORBA.INV_OBJREF is notified.

  If the connection information for the server does not exist, org.omg.CORBA.INV_OBJREF or org.omg.CORBA.COMM_FAILURE is posted.

- If the client process that issued this function shares the server where the object targeted for release (specified in the parameters) is located, the interface information belonging to the other objects is also released.

3.18.3 Server Method Standby Time Setting Interface

This section describes the server method standby time setting interface.

3.18.3.1 com.fujitsu.ObjectDirector.CORBA.ORB.set_client_timer()

Name

com.fujitsu.ObjectDirector.CORBA.ORB.set_client_timer

Format

```
public void set_client_timer ( org.omg.CORBA.Object obj, int time )
 throws org.omg.CORBA.INV_OBJREF,
        org.omg.CORBA.BAD_PARAM,
        org.omg.CORBA.NO_MEMORY;
```

Description

Specifies in time the standby time until the server method is returned to the server object specified in obj, in client applications. The standby time specified applies to all server objects operating on the host specified in obj.

Parameters

obj

Specify the object reference for the server object for which standby time is specified. A value from 0 to 100000000 can be specified as a response time.

time

Specify in seconds the standby time until the server method is returned. If 0 is specified, standby time until the server method is returned is not monitored.

Return Values

Normal termination: None

Abnormal termination: Issues the following system exception:

- org.omg.CORBA.INV_OBJREF

  An invalid object is specified in obj.

- org.omg.CORBA.BAD_PARAM

  An invalid value is specified in time.

- org.omg.CORBA.NO_MEMORY

  There is insufficient memory.

For details on the minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Usage

The ORB class instance obtained by the ORB.init() method is cast in the com.fujitsu.ObjectDirector.CORBA.ORB form, then this method is called.

However, do not specify the com.fujitsu.ObjectDirector.CORBA package in the import declaration.

Coding Example

```
org.omg.CORBA.ORB orb=org.omg.CORBA.ORB.init(app, props);
                    :
   ((com.fujitsu.ObjectDirector.CORBA.ORB)orb).set_client_timer(obj, time);
```

Notes

- This method is used when the standby time of the client processing must be modified from the standby time set in the config file (period_receive_timeout x 5 seconds).
- After obtaining the object references of a server object, issue this method to set the standby time in process units before sending a request to the server. If the standby time is to be changed during application operation, release the connection information with

  com.fujitsu.ObjectDirector.CORBA.ORB.net_disconnect()

  and then issue this method to set a new standby time.
- However, if a request has been issued to the same server machine before or when the object references of the server object are obtained, release the connection information with

  com.fujitsu.ObjectDirector.CORBA.ORB.net_disconnect() before using this method.

  This applies, for example, when org.omg.CosNaming.NamingContext.resolve() has been used to obtain the object references of a server object and the machine in which the Naming Service is operating is the same as the machine in which the server application is operating.
- This method cannot be used for the ORB instance created with ORB.init() without arguments.

3.18.3.2 com.fujitsu.ObjectDirector.CORBA.ORB.set_client_request_timer()

Name

com.fujitsu.ObjectDirector.CORBA.ORB.set_client_request_timer

Format

```
public void set_client_request_timer(int time)
       throws  org.omg.CORBA.BAD_PARAM,
       org.omg.CORBA.NO_MEMORY;
```

Description

Sets the response time until the server method returns in the client application, according to the time parameter. This response time applies to all future requests to be issued by the thread that issued the method. Before the thread exits the com.fujitsu.ObjectDirector.CORBA.ORB.clear_client_request_timer() method needs to be issued.

Parameters

time

Response time until the server method returns (seconds). The response time can be specified by a value from 0 to 100000000. If 0 is specified, the response time until the server method returns is not monitored.

Return Values

Normal termination: None

Abnormal termination: Issues the following system exception:

- org.omg.CORBA.BAD_PARAM

  An invalid value is specified in time.

- org.omg.CORBA.NO_MEMORY

  There is insufficient memory.

For details on the minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Usage

Cast the ORB class instance obtained by the ORB.init() method to the com.fujitsu.ObjectDirector.CORBA.ORB type before invoking this method.

However, do not specify the com.fujitsu.ObjectDirector.CORBA package in the import declaration.

Coding example

```
org.omg.CORBA.ORB orb=org.omg.CORBA.ORB.init(app, props);
                        :
 ((com.fujitsu.ObjectDirector.CORBA.ORB)orb).set_client_request_timer(time);
```

Notes

- Use this method to change the client process response time.
- To set the response time for each thread, issue this method before sending a request to the server.
- When this method is issued, issue com.fujitsu.ObjectDirector.CORBA.ORB.clear_client_request_timer() before exiting the thread, or a memory leak occurs.
- After this method is issued, do not change the thread name until com.fujitsu.ObjectDirector.CORBA.ORB.clear_client_request_timer() is issued, or a memory leak occurs.
- To set the time-out value to another value after this method is issued, reissue this method. In this case, com.fujitsu.ObjectDirector.CORBA.ORB.clear_client_request_timer() does not need to be issued in advance.
- The precedence of response time is shown below:

  1) Setting by com.fujitsu.ObjectDirector.CORBA.ORB.set_client_request_timer()

  2) Setting by com.fujitsu.ObjectDirector.CORBA.ORB.set_client_timer()

3) Response time set by the config file (period_receive_timeout x 5 seconds)

3.18.3.3 com.fujitsu.ObjectDirector.CORBA.ORB.get_client_request_timer()

Name

com.fujitsu.ObjectDirector.CORBA.ORB.get_client_request_timer

Format

```
public int get_client_request_timer(org.omg.CORBA.Object obj);
```

Description

This method obtains the server method response time (seconds).

The priority order of the standby time settings is given below. Note that if obj is specified by null, the value set in 1 or 3 below is obtained.

1. Setting by com.fujitsu.ObjectDirector.CORBA.ORB.set_client_request_timer()
2. Setting by com.fujitsu.ObjectDirector.CORBA.ORB.set_client_timer()
3. Response time set by the config file (period_receive_timeout x 5 seconds)

Use this function when you want to check the server method response time in the client application.

Parameters

obj

Specify the object reference to a server object when obtaining the response time set by com.fujitsu.ObjectDirector.CORBA.ORB.set_client_timer().

Return Values

Normal termination: Returns the server method response time.

Abnormal termination: Issues the following system exception:

- org.omg.CORBA.SystemException

For details on the minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Usage

Cast the ORB class instance obtained by the ORB.init() method to the com.fujitsu.ObjectDirector.CORBA.ORB type before invoking this method.

However, do not specify the com.fujitsu.ObjectDirector.CORBA package in the import declaration.

Coding example

```
 org.omg.CORBA.ORB orb=org.omg.CORBA.ORB.init(app, props);
                          :
 ((com.fujitsu.ObjectDirector.CORBA.ORB)orb).get_client_request_timer(obj);
```

3.18.3.4 com.fujitsu.ObjectDirector.CORBA.ORB.clear_client_request_timer()

Name

com.fujitsu.ObjectDirector.CORBA.ORB.clear_client_request_timer

Format

```
public void clear_client_request_timer();
```

Description

This method resets the server method response time set by com.fujitsu.ObjectDirector.CORBA.ORB.set_client_request_timer(). The server method response time of future requests to be issued by the thread that issued this function returns to the system default value.

If the server method response time is set by com.fujitsu.ObjectDirector.CORBA.ORB.set_client_request_timer(), this method must be issued before the thread is exited.

If this method is issued by a thread that does not set the server method response time by com.fujitsu.ObjectDirector.CORBA.ORB.set_client_request_timer(), no operation is performed.

Return Values

Normal termination: None.

Abnormal termination: Issues the following system exception:

- org.omg.CORBA.SystemException

For details on the minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Usage

Cast the ORB class instance obtained by the ORB.init() method to the com.fujitsu.ObjectDirector.CORBA.ORB type before invoking this method.

However, do not specify the com.fujitsu.ObjectDirector.CORBA package in the import declaration.

Coding Example

```
org.omg.CORBA.ORB orb=org.omg.CORBA.ORB.init(app, props);
                          :
  ((com.fujitsu.ObjectDirector.CORBA.ORB)orb).clear_client_request_timer();
```

3.19 Load Balance Function Interface Windows32 Solaris32 Linux32

This section describes the load balance function interface.

Notes

- This is not valid for Windows Server(R) x64 Editions and Linux for Intel64.
- The load balance function can only be used in the Interstage Application Server Enterprise Edition.
- The load balance function interface is not supported in the IIOP Service (Java EE client).

3.19.1 Load Balance Option Interface Windows32 Solaris32 Linux32

This section describes the load balance option interface.

3.19.1.1 com.fujitsu.ObjectDirector.ISOD.LBO.create_LBG() Windows32 Solaris32 Linux32

Name

com.fujitsu.ObjectDirector.ISOD.LBO.create_LBG

Format

```
public com.fujitsu.ObjectDirector.ISOD.LBG create_LBG(org.omg.CosNaming.NamingContext nc,
  org.omg.CosNaming.NameComponent[] name,
  com.fujitsu.ObjectDirector.ISOD.LBOPackage.LoadBalanceType loadbalancetype,
  org.omg.CORBA.Object defaultobjref) throws
```

```
com.fujitsu.ObjectDirector.ISOD.LBOPackage.NotFound,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.CannotProceed,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.InvalidName,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.AlreadyExist,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.InvalidType,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.BadObject,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.OperationBusy;
```

Description

This function creates a load balance object group and registers it in the naming service as the name specified for name under the naming context specified for nc.

Parameters

nc

The naming context object

name

Name of the load balance object group to be created

Specify this name as a NameComponent object array.

loadbalancetype

Specify roundrobin.

defaultobjref

Specify the default object reference returned by the naming service if the load balance function is not running.

When a null is specified with the load balance function disabled, the object reference is not returned from Naming Service.

This object reference cannot be registered and deleted by com.fujitsu.ObjectDirector.ISOD::LBG::bind() and com.fujitsu.ObjectDirector.ISOD::LBG::unbind().

After a load balance object group is created, the default object reference can be changed by using com.fujitsu.ObjectDirector.ISOD::LBG::rebind_default().

Return Values

Normal termination: Returns the object reference of the generated load balance object group.

Abnormal termination: Issues the following system exception:

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.NotFound

  Name specified at n not found

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.CannotProceed

  Naming context specified at nc does not exist

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.InvalidName

  Name specification is incorrect

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.AlreadyExist

  Specified name and object already bound

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.InvalidType

  Load balance type specification is incorrect

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.BadObject

  Object specified at defaultobjectref is not suitable

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.OperationBusy

  Upper-limit for number of request processed simultaneously has been reached. Please try again.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.19.1.2 com.fujitsu.ObjectDirector.ISOD.LBO.resolve_LBG() Windows32 Solaris32 Linux32

Name

com.fujitsu.ObjectDirector.ISOD.LBO.resolve_LBG

Format

```
public com.fujitsu.ObjectDirector.ISOD.LBG resolve_LBG(org.omg.CosNaming.NamingContext
 nc,
      org.omg.CosNaming.NameComponent[] n) throws
com.fujitsu.ObjectDirector.ISOD.LBOPackage.NotFound,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.CannotProceed,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.InvalidName,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.OperationBusy;
```

Description

This function searches the load balance object group that has the name specified for n under the naming context specified for nc.

Parameters

nc

The naming context object

n

Load balance object group name to be retrieved

Specify this name as a NameComponent object array.

Return Values

Normal termination: Returns the object reference of the detected load balance object group.

Abnormal termination: Issues the following system exception:

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.NotFound

  Name specified at n not found

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.CannotProceed

  Naming context specified at nc does not exist

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.InvalidName

  Name specification is incorrect

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.OperationBusy

  Upper-limit for number of request processed simultaneously has been reached. Please try again.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.19.1.3 com.fujitsu.ObjectDirector.ISOD.LBO.delete_LBG() Windows32 Solaris32 Linux32

Name

com.fujitsu.ObjectDirector.ISOD.LBO.delete_LBG

Format

```
public org.omg.CORBA.Object delete_LBG(org.omg.CosNaming.NamingContext nc,
      org.omg.CosNaming.NameComponent[] n) throws
com.fujitsu.ObjectDirector.ISOD.LBOPackage.NotFound,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.CannotProceed,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.InvalidName,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.NotEmpty,
com.fujitsu.ObjectDirector.ISOD.LBOPackage.OperationBusy;
```

Description

This function deletes a load balance object group from the naming service. The load balance object group that has the name specified for n under the naming context specified for nc is deleted. An exception occurs if any objects other than the default object are still registered in the load balance object group that is targeted for deletion.

Parameters

nc

The naming context object

n

Load balance object group name to be deleted

Specify this name as a NameComponent object array.

Return Values

Normal termination: Returns the object reference of the default object.

Abnormal termination: Issues the following system exception:

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.NotFound

  Name specified at n not found

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.CannotProceed

  Naming context specified at nc does not exist

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.InvalidName

  Name specification is incorrect

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.NotEmpty

  One or more objects, other than the default object, are registered in the object group.

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.OperationBusy

  Upper-limit for number of request processed simultaneously has been reached. Please try again.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.19.1.4 com.fujitsu.ObjectDirector.ISOD.LBO.list_LBG() Windows32 Solaris32 Linux32

Name

com.fujitsu.ObjectDirector.ISOD.LBO.list_LBG

Format

```
public void list_LBG(org.omg.CosNaming.NamingContext nc,
      com.fujitsu.ISOD.LBOPackage.LBGListHolder bl);
```

Description

This function returns a list of the load balance object groups under the naming context specified for nc. The returns the load balance object group names and the sequence of the structures (BindingObjectGroup) that hold load balance type data. Information can be retrieved for up to 256 load balance object groups.

Parameters

nc

The naming context object

bl

Area where the list of load balance objects is returned

Return Values

Normal termination: None.

Abnormal termination: Issues the following system exception:

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

Note

When 257 loading balance object groups, or more, are registered, the list of all the object groups cannot be retrieved.

3.19.1.5 com.fujitsu.ObjectDirector.ISOD.LBO.notify_down() Windows32 Solaris32 Linux32

Name

com.fujitsu.ObjectDirector.ISOD.LBO.notify_down

Format

```
public void notify_down(java.lang.String HostName ) throws
      com.fujitsu.ObjectDirector.ISOD.LBOPackage.InvalidArgument,
      com.fujitsu.ObjectDirector.ISOD.LBOPackage.CannotProceed2;
```

Description

This function notifies the load balance function if the server is down. Calling this function stops the return of object references for objects at the server address specified at HostName.

Parameters

HostName

Host name or IP address of the failed server

Return Values

Normal termination: None.

Abnormal termination: Issues the following system exception:

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.InvalidArgument

  HostName specification is incorrect

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.CannotProceed2

  Error occurred during processing of load balance function database.

3.19.1.6 com.fujitsu.ObjectDirector.ISOD.LBO.notify_recover() Windows32 Solaris32 Linux32

Name

com.fujitsu.ObjectDirector.ISOD.LBO.notify_recover

Format

```
public void notify_recover(java.jang.String HostName) throws
       com.fujitsu.ObjectDirector.ISOD.LBOPackage.InvalidArgument,
       com.fujitsu.ObjectDirector.ISOD.LBOPackage.CannotProceed2;
```

Description

This function notifies the load balance function of server start or recovery. Calling this function starts the return of object references for objects at the server address specified at HostName.

Parameters

HostName

Host name or IP address of the recovered server

Return Values

Normal termination: None.

Abnormal termination: Issues the following system exception:

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.InvalidArgument

  HostName specification is incorrect

- com.fujitsu.ObjectDirector.ISOD.LBOPackage.CannotProceed2

  Error occurred during processing of load balance function database

3.19.2 Load Balance Object Group Interface Windows32 Solaris32 Linux32

This section describes the load balance object group interface.

3.19.2.1 com.fujitsu.ObjectDirector.ISOD.LBG.bind() Windows32 Solaris32 Linux32

Name

com.fujitsu.ObjectDirector.ISOD.LBG.bind

Format

```
public void bind(org.omg.CORBA.Object objectref) throws
       com.fujitsu.ObjectDirector.ISOD.LBGPackage.AlreadyBound,
       com.fujitsu.ObjectDirector.ISOD.LBGPackage.CannotProceed2,
       com.fujitsu.ObjectDirector.ISOD.LBGPackage.BadObject;
```

Description

The object specified at objectref as the load balancing target is registered to this load balance object group by this function.

Parameters

objectref

Reference to the object to be added to the load balance object group

Return Values

Normal termination: None.

Abnormal termination: Issues the following system exception:

- com.fujitsu.ObjectDirector.ISOD.LBGPackage.AlreadyBound

  An object with the same data as the specified object is already registered.

- com.fujitsu.ObjectDirector.ISOD.LBGPackage.CannotProceed2

  Error occurred during processing of load balance function database. Or, the objectref specification is incorrect.

- com.fujitsu.ObjectDirector.ISOD.LBGPackage.BadObject

  Object specified at objectref is not suitable

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.19.2.2 com.fujitsu.ObjectDirector.ISOD.LBG.unbind() Windows32 Solaris32 Linux32

Name

com.fujitsu.ObjectDirector.ISOD.LBG.unbind

Format

```
public void unbind(org.omg.CORBA.Object objectref) throws
       com.fujitsu.ObjectDirector.ISOD.LBGPackage.NotFound,
       com.fujitsu.ObjectDirector.ISOD.LBGPackage.CannotProceed2,
       com.fujitsu.ObjectDirector.ISOD.LBGPackage.BadObject;
```

Description

The object specified at objectref as the load balancing target is deleted from this load balance object group by this function.

Parameters

objectref

Reference to the object to be deleted from the load balance object group

Return Values

Normal termination: None.

Abnormal termination: Issues the following system exception:

- com.fujitsu.ObjectDirector.ISOD.LBGPackage.NotFound

  Object specified at objectref not found

- com.fujitsu.ObjectDirector.ISOD.LBGPackage.CannotProceed2

  Error occurred during processing of load balance function database. Or, the objectref specification is incorrect.

- com.fujitsu.ObjectDirector.ISOD.LBGPackage.BadObject

  Object specified at objectref is not suitable

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.19.2.3 com.fujitsu.ObjectDirector.ISOD.LBG.rebind_default() Windows32 Solaris32 Linux32

Name

com.fujitsu.ObjectDirector.ISOD.LBG.rebind_default

Format

```
public org.omg.CORBA.Object rebind_default(org.omg.CORBA.Object objectref) throws
       com.fujitsu.ObjectDirector.ISOD.LBGPackage.CannotProceed,
       com.fujitsu.ObjectDirector.ISOD.LBGPackage.CannotProceed2,
       com.fujitsu.ObjectDirector.ISOD.LBGPackage.BadObject,
       com.fujitsu.ObjectDirector.ISOD.LBGPackage.OperationBusy;
```

Description

This function changes the default object of this load balance object group to the object specified at objectref.

Parameters

objectref

Reference to the object to be added as the default object

Return Values

Normal termination: Returns the object reference of the previous default object.

Abnormal termination: Issues the following system exception:

- com.fujitsu.ObjectDirector.ISOD.LBGPackage.CannotProceed

  Default object has not been set

- com.fujitsu.ObjectDirector.ISOD.LBGPackage.CannotProceed2

  Error occurred during processing of the load balance function database. Or, objectref specification is incorrect.

- com.fujitsu.ObjectDirector.ISOD.LBGPackage.BadObject

  Object specified at objectref is not suitable

- com.fujitsu.ObjectDirector.ISOD.LBGPackage.OperationBusy

  Upper-limit for number of request processed simultaneously has been reached. Please try again.

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.19.2.4 com.fujitsu.ObjectDirector.ISOD.LBG.list() Windows32 Solaris32 Linux32

Name

com.fujitsu.ObjectDirector.ISOD.LBG.list

Format

```
public void list(com.fujitsu.ISOD.LBGPackage.ObjectListHolder objectref) throws
       com.fujitsu.ObjectDirector.ISOD.LBGPackage.CannotProceed2;
```

Description

This function returns a list of the object references of objects already registered in this load balance object group.

Parameters

objectref

Area where the object list is returned

Return Values

Normal termination: None.

Abnormal termination: Issues the following system exception:

- com.fujitsu.ObjectDirector.ISOD.LBGPackage.CannotProceed2

  Error occurred during processing of the load balance function database

- org.omg.CORBA.SystemException

  Other causes

For details on the exception information and minor codes set when a system exception occurs, refer to Exception Information and Minor Codes to be Reported from the CORBA Service in the Messages Manual.

3.20 Event Service Interface

The event service event channel can be used for communication.

When communicating with the event service event channel, check that esjava4.jar has been specified in the CLASSPATH variable.

Note

The event service interface is not supported in the IIOP Service (Java EE client).

3.20.1 org.omg.CosEventComm Class

This section describes the org.omg.CosEventComm Class.

3.20.1.1 org.omg.CosEventComm.PushConsumer.push()

Name

org.omg.CosEventComm.PushConsumer.push

Format

```
import  org.omg.CosEventComm.*;
public  interface PushConsumer extends org.omg.CORBA.Object {
       public  void  push(
                        org.omg.CORBA.Any     data )
                        throws org.omg.CosEventComm.Disconnected;
  }
```

Description

Sends event data, specified by data, to a consumer.

Parameters

data

Event data to be transmitted to the consumer

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

In the event of a user error, issues the following exception:

- org.omg.CosEventComm.Disconnected

  Not connected to an event channel.

  If processing for automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

3.20.1.2 org.omg.CosEventComm.PushConsumer.disconnect_push_consumer()

Name

org.omg.CosEventComm.PushConsumer.disconnect_push_consumer

Format

```
import  org.omg.CosEventComm.*;
public  interface PushConsumer extends org.omg.CORBA.Object {
       public  void  disconnect_push_consumer();

  }
```

Description

Declares an end to event communication from the supplier.

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.20.1.3 org.omg.CosEventComm.PushSupplier.disconnect_push_supplier()

Name

org.omg.CosEventComm.PushSupplier.disconnect_push_supplier

Format

```
import  org.omg.CosEventComm.*;
public  interface PushSupplier extends org.omg.CORBA.Object {
       public  void  disconnect_push_supplier();

}
```

Description

Declares an end to event communication from the consumer.

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.20.1.4 org.omg.CosEventComm.PullSupplier.pull()

Name

org.omg.CosEventComm.PullSupplier.pull

Format

```
import  org.omg.CosEventComm.*;
public  interface PullSupplier extends org.omg.CORBA.Object {
       public org.omg.CORBA.Any  pull()
                        throws org.omg.CosEventComm.Disconnected;
}
```

Description

Requests event data from a supplier. It will be blocked until either it knows if it can fetch the data, or an exception is issued. If you want to return as soon as you find out that you cannot fetch the data, use org.omg.CosEventComm.PullSupplier.try_pull.

Return Values

Normal termination: Event data is returned from the supplier.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

In the event of a user exception, issues the following exception:

- org.omg.CosEventComm.Disconnected

  Not connected to an event channel.

If processing for automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

3.20.1.5 org.omg.CosEventComm.PullSupplier.try_pull()

Name

org.omg.CosEventComm.PullSupplier.try_pull

Format

```
import  org.omg.CosEventComm.*;
public  interface PullSupplier extends org.omg.CORBA.Object {
       public org.omg.CORBA.Any  try_pull(
                         org.omg.CORBA.BooleanHolder    has_event )
                         throws org.omg.CosEventComm.Disconnected;
}
```

Description

Requests event data from a supplier. Immediately returns if it cannot fetch the event data from the supplier. If you want to block until you can fetch the data, use org.omg.CosEventComm.PullSupplier.pull.

Parameters

has_event (out parameter)

When event data can be retrieved, "true" is set.

When event data cannot be retrieved, "false" is set.

Return Values

Normal termination: Returns event data from the supplier.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosEventComm.Disconnected

  Not connected to an event channel.

  If processing for automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

Notes

The cancellation need not be notified using com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_rollback() when CORBA_FALSE is set to has_event when a local transaction is operated.

Notifies the channel of completing using com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_commit().

The cancellation need not be notified using org.omg.CosTransactions.Current.rollback() when CORBA_FALSE is set to has_event when a globaltransaction is operated.

Notifies the channel of completing using org.omg.CosTransactions.Current.commit().

3.20.1.6 org.omg.CosEventComm.PullSupplier.disconnect_pull_supplier()

Name

org.omg.CosEventComm.PullSupplier.disconnect_pull_supplier

Format

```
import  org.omg.CosEventComm.*;
public  interface PullSupplier extends org.omg.CORBA.Object {
       public  void  disconnect_pull_supplier();

}
```

Description

Declares an end to event communication from the consumer.

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.20.1.7 org.omg.CosEventComm.PullConsumer.disconnect_pull_consumer()

Name

org.omg.CosEventComm.PullConsumer.disconnect_pull_consumer

Format

```
import  org.omg.CosEventComm.*;
public  interface PullConsumer extends org.omg.CORBA.Object {
       public  void  disconnect_pull_consumer();

}
```

Description

Declares an end to event communication from the consumer.

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.20.2 org.omg.CosEventChannelAdmin class

This section describes the org.omg.CosEventChannelAdmin class.

3.20.2.1 org.omg.CosEventChannelAdmin.EventChannel.for_consumers()

Name

org.omg.CosEventChannelAdmin.EventChannel.for_consumers

Format

```
import  org.omg.CosEventChannelAdmin.*;
public  interface EventChannel extends org.omg.CORBA.Object {
       public  org.omg.CosEventChannelAdmin.ConsumerAdmin  for_consumers();

}
```

Description

Obtains an event channel object reference that will allow the consumer to connect to the event channel.

Return values

Normal termination: Returns ConsumerAdmin's object reference.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.20.2.2 org.omg.CosEventChannelAdmin.EventChannel.for_suppliers()

Name

org.omg.CosEventChannelAdmin.EventChannel.for_suppliers

Format

```
import  org.omg.CosEventChannelAdmin.*;
public  interface EventChannel extends org.omg.CORBA.Object {
       public  org.omg.CosEventChannelAdmin.SupplierAdmin  for_suppliers();

}
```

Description

Obtains an event channel object reference that will allow the supplier to connect to the event channel.

Return values

Normal termination: Returns SupplierAdmin's object reference.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.20.2.3 org.omg.CosEventChannelAdmin.EventChannel.destroy()

Name

org.omg.CosEventChannelAdmin.EventChannel.destroy

Format

```
import  org.omg.CosEventChannelAdmin.*;
public  interface EventChannel extends org.omg.CORBA.Object {
       public  void  destroy();

}
```

Description

Destroys the specified event channel.

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.20.2.4 org.omg.CosEventChannelAdmin.ConsumerAdmin.obtain_push_supplier()

Name

org.omg.CosEventChannelAdmin.ConsumerAdmin.obtain_push_supplier

Format

```
import  org.omg.CosEventChannelAdmin.*;
public  interface ConsumerAdmin extends org.omg.CORBA.Object {
       public  org.omg.CosEventChannelAdmin.ProxyPushSupplier  obtain_push_supplier();

}
```

Description

Obtains an event channel object reference that will allow a Push model consumer to connect to the event channel.

Return values

Normal termination: Returns ProxyPushSupplier's object reference.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.20.2.5 org.omg.CosEventChannelAdmin.ConsumerAdmin.obtain_pull_supplier()

Name

org.omg.CosEventChannelAdmin.ConsumerAdmin.obtain_pull_supplier

Format

```
import  org.omg.CosEventChannelAdmin.*;
public  interface ConsumerAdmin extends org.omg.CORBA.Object {
       public  org.omg.CosEventChannelAdmin.ProxyPullSupplier  obtain_pull_supplier();

}
```

Description

Obtains an event channel object reference that will allow a Pull model consumer to connect to the event channel.

Return Values

Normal termination: Returns ProxyPullSupplier's object reference.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.20.2.6 org.omg.CosEventChannelAdmin.SupplierAdmin.obtain_push_consumer()

Name

org.omg.CosEventChannelAdmin.SupplierAdmin.obtain_push_consumer

Format

```
import  org.omg.CosEventChannelAdmin.*;
public  interface SupplierAdmin extends org.omg.CORBA.Object {
       public  org.omg.CosEventChannelAdmin.ProxyPushConsumer  obtain_push_consumer();

}
```

Description

Obtains an event channel object reference that will allow a Push model supplier to connect to the event channel.

Return Values

Normal termination: Returns ProxyPushConsumer's object reference.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.20.2.7 org.omg.CosEventChannelAdmin.SupplierAdmin.obtain_pull_consumer()

Name

org.omg.CosEventChannelAdmin.SupplierAdmin.obtain_pull_consumer

Format

```
import  org.omg.CosEventChannelAdmin.*;
public  interface SupplierAdmin extends org.omg.CORBA.Object {
       public  org.omg.CosEventChannelAdmin.ProxyPullConsumer  obtain_pull_consumer();

}
```

Description

Obtains an event channel object reference that will allow a Pull model supplier to connect to the event channel.

Return Values

Normal termination: Returns ProxyPullConsumer's object reference.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.20.2.8 org.omg.CosEventChannelAdmin.ProxyPushConsumer.connect_push_supplier()

Name

org.omg.CosEventChannelAdmin.ProxyPushConsumer.connect_push_supplier

Format

```
import  org.omg.CosEventChannelAdmin.*;
public  interface ProxyPushConsumer extends CosEventComm.PushConsumer {
       public  void  connect_push_supplier(
               org.omg.CosEventComm.PushSupplier    push_supplier )
               throws org.omg.CosEventChannelAdmin.AlreadyConnected;
}
```

Description

Connects a Push model supplier to an event channel.

Parameters

push_supplier

The supplier's object reference

If you do not need to give a disconnect notification when the event channel closes, set a null.

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosEventChannelAdmin.AlreadyConnected

  Event channel already connected.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

Note

To reconnect to an event channel, start again from org.omg.CosEventChannelAdmin.SupplierAdmin.obtain_push_consumer.

3.20.2.9 org.omg.CosEventChannelAdmin.ProxyPullSupplier.connect_pull_consumer( )

Name

org.omg.CosEventChannelAdmin.ProxyPullSupplier.connect_pull_consumer

Format

```
import  org.omg.CosEventChannelAdmin.*;
public  interface ProxyPullSupplier extends CosEventComm.PullSupplier {
       public  void  connect_pull_consumer(
               org.omg.CosEventComm.PullConsumer     pull_consumer )
               throws org.omg.CosEventChannelAdmin.AlreadyConnected;
  }
```

Description

Connects a Pull model consumer to an event channel.

Parameters

pull_consumer

The consumer's object reference

If you do not need to give a disconnect notification when the event channel closes, set a null.

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosEventChannelAdmin.AlreadyConnected

  Event channel already connected.

For system exceptions, refer to "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" in the Messages manual to correct the problem.

Note

To reconnect to an event channel, start again from org.omg.CosEventChannelAdmin.ConsumerAdmin.obtain_pull_supplier.

3.20.2.10 org.omg.CosEventChannelAdmin.ProxyPullConsumer.connect_pull_supplier()

Name

org.omg.CosEventChannelAdmin.ProxyPullConsumer.connect_pull_supplier

Format

```
import  org.omg.CosEventChannelAdmin.*;
public  interface ProxyPullConsumer extends CosEventComm.PullConsumer {
       public  void  connect_pull_supplier(
               org.omg.CosEventComm.PullSupplier    pull_supplier )
               throws org.omg.CosEventChannelAdmin.AlreadyConnected,
                      org.omg.CosEventChannelAdmin.TypeError;
}
```

Description

Connects a Pull model supplier to an event channel.

Parameters

pull_supplier

The supplier's object reference

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosEventChannelAdmin.AlreadyConnected

  Event channel already connected.

- org.omg.CosEventChannelAdmin.TypeError

  The specified object type is incorrect.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

Note

To reconnect to an event channel, start again from org.omg.CosEventChannelAdmin.SupplierAdmin.obtain_pull_consumer.

3.20.2.11 org.omg.CosEventChannelAdmin.ProxyPushSupplier.connect_push_consumer()

Name

org.omg.CosEventChannelAdmin.ProxyPushSupplier.connect_push_consumer

Format

```
import  org.omg.CosEventChannelAdmin.*;
  public  interface ProxyPushSupplier extends CosEventComm.PushSupplier {
       public  void  connect_push_consumer(
                       org.omg.CosEventComm.PushConsumer   push_consumer )
                       throws org.omg.CosEventChannelAdmin.AlreadyConnected,
                              org.omg.CosEventChannelAdmin.TypeError;
}
```

Description

Connects a Push model consumer to an event channel.

Parameters

push_consumer

The consumer's object reference

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosEventChannelAdmin.AlreadyConnected

  Event channel already connected.

- org.omg.CosEventChannelAdmin.TypeError

  The specified object type is incorrect.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

Note

To reconnect to an event channel, start again from org.omg.CosEventChannelAdmin.ConsumerAdmin.obtain_push_supplier.

3.20.2.12 Classes you can use when Inherited

You can inherit the following classes. For details refer to 3.20.1 org.omg.CosEventComm Class.

1. org.omg.CosEventChannelAdmin.ProxyPushConsumer.push
2. org.omg.CosEventChannelAdmin.ProxyPushConsumer.disconnect_push_consumer
3. org.omg.CosEventChannelAdmin.ProxyPushSupplier.disconnect_push_supplier
4. org.omg.CosEventChannelAdmin.ProxyPullSupplier.pull
5. org.omg.CosEventChannelAdmin.ProxyPullSupplier.try_pull

6. org.omg.CosEventChannelAdmin.ProxyPullSupplier.disconnect_pull_supplier
7. org.omg.CosEventChannelAdmin.ProxyPullConsumer.disconnect_pull_consumer

3.20.3 com.fujitsu.ObjectDirector.EventService.Event Factory Class

This section describes the com.fujitsu.ObjectDirector.EventService.Event Factory Class.

3.20.3.1 com.fujitsu.ObjectDirector.EventService.EventFactory.create()

Name

com.fujitsu.ObjectDirector.EventService.EventFactory.create

Format

```
import  com.fujitsu.ObjectDirector.EventService.*;
import  com.fujitsu.ObjectDirector.EventService.EventFactoryPackage.*;
public  interface EventFactory extends org.omg.CORBA.Object {
      public org.omg.CosEventChannelAdmin.EventChannel  create(
          String              key,
          com.fujitsu.ObjectDirector.EventService.EventFactoryPackage.Option  data );

  }
```

Description

Generates an event channel and returns its object reference.

The EventFactory object reference sets and obtains the following values in the org.omg.CORBA.ORB.resolve_initial_references() method's object_name parameters.

EventFactory.ObjectId_Factory

Parameters

key

The common keywords for consumer and supplier (maximum 64 characters).

If you use the same keyword, the same event channel's object reference will be returned.

data

EventFactory.Option class

Sets the values in the following table for each member of the EventFactory.Option class.

Table 3.1 EventFactory.Option Class Members

Member	Setting Value
max_queuing	When you set Event Factory.ES_DEFAULT_VALUE, it uses the "maximum value of event data number which can be stored in the event channel" set in the event service configuration information setting.
life_time	Data storage time (sec.) When you set EventFactory.ES_DEFAULT_VALUE, it uses the value of "existing time (sec.) of the stored event data" set in the event service configuration information setting. If 0 is specified, timeout monitoring does not take place.
model	Set the following connect model values: EventFactoryPackage.Model.ModelAny.....Determine when connected EventFactoryPackage.Model.ModelPush.....Push model EventFactoryPackage.Model.ModelPull.....Pull model EventFactoryPackage.Model.ModelMixed.....Mixed model

Return Values

Normal termination: Returns the object reference of the generated event channel.

Abnormal termination: Issues an exceptions.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.20.3.2 com.fujitsu.ObjectDirector.EventService.EventFactory.create_channel()

Name

com.fujitsu.ObjectDirector.EventService.EventFactory.create_channel

Format

```
import  com.fujitsu.ObjectDirector.EventService.*;
import  com.fujitsu.ObjectDirector.EventService.EventFactoryPackage.*;
public  interface EventFactory extends org.omg.CORBA.Object {
  public org.omg.CosEventChannelAdmin.EventChannel  create_channel(
    String   key,
    com.fujitsu.ObjectDirector.EventService.EventFactoryPackage.Option data,
    com.fujitsu.ObjectDirector.EventService.EventFactoryPackage.EventProperty property,
    org.omg.CORBA.boolean   create);
}
```

Description

Generates an event channel and returns its object reference. This method is used to change the host name and port number for the generated Channel.

The EventFactory object reference sets and obtains the following values in the org.omg.CORBA.ORB.resolve_initial_references() method's object_name parameters.

EventFactory.ObjectId_Factory

Parameters

key

The common keywords for consumer and supplier (maximum 64 characters).

If you use the same keyword, the same event channel's object reference will be returned.

data

EventFactory.Option class

Sets the values in the following table for each member of the EventFactory.Option class.

If the Event Channel has already been generated, the value for this parameter is ignored.

Table 3.2 EventFactory.Option Class Members

Member	Setting Value
max_queuing	When you set Event Factory.ES_DEFAULT_VALUE, it uses the "maximum value of event data number which can be stored in the event channel" set in the event service configuration information setting.
life_time	Data storage time (sec.) When you set EventFactory.ES_DEFAULT_VALUE, it uses the value of "existing time (sec.) of the stored event data" set in the event service configuration information setting. If 0 is specified, timeout monitoring does not take place.
model	Set the following connect model values: EventFactoryPackage.Model.ModelAny.....Determine when connected EventFactoryPackage.Model.ModelPush.....Push model EventFactoryPackage.Model.ModelPull.....Pull model EventFactoryPackage.Model.ModelMixed.....Mixed model

property

Specifies the value shown in the table below. If an invalid name is specified, the corresponding record is invalid. If either of HostName and PortNumber is set, the specified record is invalid.

If the Event Channel has already been generated, the value for this parameter is ignored.

Member (name)	Data type (value)	Set value
HostName	string	Specifies the host name or the IP address (maximum 64 characters),.
PortNumber	unsigned short	Specifies the port number.

create

If the Event Channel was generated, true is set.

If the Event Channel already exists, false is set.

Return Values

Normal termination: Returns the object reference of the generated event channel.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

Notes

This method cannot be used if the following host names are set, as the Event Channel will fail to start.

- The "Corba Host Name" Interstage operating environment definition
- "IIOP_hostname" in the config file (CORBA Service) (The host name used by the CORBA Service)

When specifying the port number, select one of the following host names specified for the CORBA Service port number:

- If SSL communication is enabled
  - The "SSL Port Number" Interstage operating environment definition
  - "UNO_IIOP_ssl_port" in the config file (CORBA Service)
- If SSL communication is disabled
  - The "Corba Port Number" Interstage operating environment definition
  - "IIOP_port" in the config file (CORBA Service)

3.20.3.3 com.fujitsu.ObjectDirector.EventService.EventFactory.get_event_channel()

Name

com.fujitsu.ObjectDirector.EventService.EventFactory.get_event_channel

Format

```
import  com.fujitsu.ObjectDirector.EventService.*;
import  com.fujitsu.ObjectDirector.EventService.EventFactoryPackage.*;
public  interface EventFactory extends org.omg.CORBA.Object {
     public  org.omg.CosEventChannelAdmin.EventChannel  get_event_channel(
               String    key)
               throws com.fujitsu.ObjectDirector.EventService.EventFactoryPackage.ChannelNotFound;
}
```

Description

The EventFactory object reference sets and obtains the following values in the org.omg.CORBA.ORB.resolve_initial_references() method's object_name parameters.

EventFactory.ObjectId_Factory

Parameters

key

The acquired Event Channel name

Return Values

Normal termination: Returns the object reference of the event channel specified with key.

Abnormal termination: Issues an exception.

In the event of a user exception: issues the following exception:

- com.fujitsu.ObjectDirector.EventService.EventFactoryPackage.ChannelNotFound

  The specified event channel could not be found.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21 Notification Service Interface

This function enables communication with the notification service event channel.

When communicating with the notification service event channel, check that the "esnotifyjava4.jar" have been specified in the CLASSPATH variable:

If the notification service function interface is used with the event service event channel, a BAD_OPERATION system exception occurs.

Notes

- The notification service function can only be used in Interstage Application Server Enterprise Edition.
- The notification service interface is not supported in the IIOP Service (Java EE client).

3.21.1 CosNotification Interface

Type Definitions

The formats of the data types used by this module are shown below.

```
package org.omg.CosNotification;
public class Property {
    public java.lang.String name;
    public org.omg.CORBA.Any value;
    public Property() {}
    public Property( java.lang.String name,
              org.omg.CORBA.Any value ) {}
}

package org.omg.CosNotification;
public class EventType {
    public java.lang.String domain_name;
    public java.lang.String type_name;
    public EventType() {}
    public EventType( java.lang.String domain_name,
              java.lang.String type_name ) {}
}
```

```
package org.omg.CosNotification;
public class PropertyRange {
    public org.omg.CORBA.Any low_val;
    public org.omg.CORBA.Any high_val;
    public PropertyRange() {}
    public PropertyRange( org.omg.CORBA.Any low_val,
              org.omg.CORBA.Any high_val ) {}
}

package org.omg.CosNotification;
public class NamedPropertyRange {
    public java.lang.String name;
    public org.omg.CosNotification.PropertyRange range;
    public NamedPropertyRange() {}
    public NamedPropertyRange( java.lang.String name,
              org.omg.CosNotification.PropertyRange range ) {}
}

package org.omg.CosNotification;
public class QoSError_code {
    public static final int _UNAVAILABLE_VALUE = (int)3;
    public static final QoSError_code UNAVAILABLE_VALUE = new
QoSError_code(_UNAVAILABLE_VALUE);
    public static final int _BAD_PROPERTY = (int)4;
    public static final QoSError_code BAD_PROPERTY = new
QoSError_code(_BAD_PROPERTY);
    public static final int _BAD_TYPE = (int)5;
    public static final QoSError_code BAD_TYPE = new QoSError_code(_BAD_TYPE);
    public int value() {}
    public static QoSError_code from_int( int i ) {}
}

package org.omg.CosNotification;
public class PropertyError {
    public org.omg.CosNotification.QoSError_code code;
    public java.lang.String name;
    public org.omg.CosNotification.PropertyRange available_range;
    public PropertyError() {}
    public PropertyError( org.omg.CosNotification.QoSError_code code,
              java.lang.String name,
              org.omg.CosNotification.PropertyRange available_range ) {}
}

package org.omg.CosNotification;
public class FixedEventHeader {
    public org.omg.CosNotification.EventType event_type;
    public java.lang.String event_name;
    public FixedEventHeader() {}
    public FixedEventHeader( org.omg.CosNotification.EventType event_type,
              java.lang.String event_name ) {}
}

package org.omg.CosNotification;
public class EventHeader {
    public org.omg.CosNotification.FixedEventHeader fixed_header;
    public org.omg.CosNotification.Property[] variable_header;
    public EventHeader() {}
    public EventHeader( org.omg.CosNotification.FixedEventHeader fixed_header,
               org.omg.CosNotification.Property[] variable_header ) {}
}

package org.omg.CosNotification;
public class StructuredEvent {
```

```
    public org.omg.CosNotification.EventHeader header;
    public org.omg.CosNotification.Property[] filterable_data;
    public org.omg.CORBA.Any remainder_of_body;
    public StructuredEvent() {}
    public StructuredEvent( org.omg.CosNotification.EventHeader header,
                            org.omg.CosNotification.Property[] filterable_data,
                            org.omg.CORBA.Any remainder_of_body ) {}
}
```

3.21.2 QoSAdmin Interface

This section describes the QoSAdmin Interface.

3.21.2.1 org.omg.CosNotification.QoSAdmin.get_qos()

Name

org.omg.CosNotification.QoSAdmin.get_qos

Format

```
import  org.omg.CosNotification.*;
  public  interface QoSAdmin extends org.omg.CORBA.Object {
       public  org.omg.CosNotification.Property[]  get_qos();

  }
```

Description

Fetches QoSProperties.

Return Values

Normal termination: Returns QoSProperties.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.2.2 org.omg.CosNotification.QoSAdmin.set_qos()

Name

org.omg.CosNotification.QoSAdmin.set_qos

Format

```
import  org.omg.CosNotification.*;
  public  interface QoSAdmin extends org.omg.CORBA.Object {
      public  void set_qos( org.omg.CosNotification.Property[] qos )
          throws org.omg.CosNotification.UnsupportedQoS;
  }
```

Description

Sets QoSProperties. Specify in qos the QoSProperties that have been set.

Parameters

qos

The QoS property item

Return Values

Normal termination: None.

Abnormal termination: Issues an exceptions.

In the event of a user exception, issues the following exception:

- org.omg.CosNotification.UnsupportedQoS

  There is an error in the specified QoS property item or value.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.3 CosNotifyComm Module

This section describes the CosNotifyComm Module.

3.21.3.1 org.omg.CosNotifyComm.StructuredPushConsumer.push_structured_event()

Name

org.omg.CosNotifyComm.StructuredPushConsumer.push_structured_event

Format

```
import  org.omg.CosNotifyComm.*;
 public  interface StructuredPushConsumer extends org.omg.CORBA.Object {
  public  void push_structured_event ( org.omg.CosNotification.StructuredEvent  data)
    throws org.omg.CosEventComm.Disconnected;
  }
```

Description

Sends the StructuredEvent type event data specified in data to the consumer.

Parameters

data

The StructuredEvent type event data sent to consumer

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

In the event of a user exception, Issues the following exception:

- org.omg.CosEventComm.Disconnected

  Not connected to the event channel.

  If processing for automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.3.2 org.omg.CosNotifyComm.StructuredPullSupplier.pull_structured_event()

Name

org.omg.CosNotifyComm.StructuredPullSupplier.pull_structured_event

Format

```
import  org.omg.CosNotifyComm.*;
public  interface StructuredPullSupplier extends org.omg.CORBA.Object {
    public  org.omg.CosNotification.StructuredEvent pull_structured_event ()
          throws org.omg.CosEventComm.Disconnected;
  }
```

Description

Requests the supplier for StructuredEvent type event data. This function is blocked until the event data can be fetched or an exception is issued. To return immediately if the event data cannot be fetched, use org.omg.CosNotifyComm.StructuredPullSupplier.try_pull_structured_event.

Return Values

Normal termination: Returns the event data from the supplier.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosEventComm.Disconnected

  Not connected to the event channel.

  If processing for automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.3.3 org.omg.CosNotifyComm.StructuredPullSupplier.try_pull_structured_event()

Name

org.omg.CosNotifyComm.StructuredPullSupplier.try_pull_structured_event

Format

```
import  org.omg.CosNotifyComm.*;
  public  interface StructuredPullSupplier extends org.omg.CORBA.Object {
      public  org.omg.CosNotification.StructuredEvent try_pull_structured_event (
          org.omg.CORBA.BooleanHolder    has_event )
          throws org.omg.CosEventComm.Disconnected;
  }
```

Description

Requests the supplier for StructuredEvent type event data. Immediately returns if event data cannot be fetched from the supplier. To block until the event data can be fetched, use org.omg.CosNotifyComm.StructuredPullSupplier.pull_structured_event.

Parameters

has_event (out parameter)

When event data can be retrieved, "true" is set.

When event data cannot be retrieved, "false" is set.

Return Values

Normal termination: Returns the event data from the supplier when the event data is fetched.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosEventComm.Disconnected

  Not connected to the event channel.

  If processing for automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

Notes

The cancellation need not be notified using com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_rollback() when CORBA_FALSE is set to has_event when a local transaction is operated.

Notifies the channel of completing using com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_commit().

The cancellation need not be notified using org.omg.CosTransactions.Current.rollback() when CORBA_FALSE is set to has_event when a globaltransaction is operated.

Notifies the channel of completing using org.omg.CosTransactions.Current.commit().

3.21.3.4 org.omg.CosNotifyComm.StructuredPushConsumer.disconnect_structured_push_consumer()

Name

org.omg.CosNotifyComm.StructuredPushConsumer.disconnect_structured_push_consumer

Format

```
import  org.omg.CosNotifyComm.*;
  public  interface StructuredPushConsumer extends org.omg.CORBA.Object {
      public  void disconnect_structured_push_consumer ();

  }
```

Description

Declares the end of event communication from the supplier.

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.3.5 org.omg.CosNotifyComm.StructuredPullSupplier.disconnect_structured_pull_supplier()

Name

org.omg.CosNotifyComm.StructuredPullSupplier.disconnect_structured_pull_supplier

Format

```
import  org.omg.CosNotifyComm.*;
  public  interface StructuredPullSupplier extends org.omg.CORBA.Object {
      public  void disconnect_structured_pull_supplier ();

  }
```

Description

Declares the end of event communication from the consumer.

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.3.6 Classes that can be Inherited

The following classes can be inherited.

Refer to 3.20.1 org.omg.CosEventComm Class for details.

1. org.omg.CosNotifyComm.PushConsumer.push
2. org.omg.CosNotifyComm.PushConsumer.disconnect_push_consumer
3. org.omg.CosNotifyComm.PullSupplier.pull
4. org.omg.CosNotifyComm.PullSupplier.try_pull
5. org.omg.CosNotifyComm.PullSupplier.disconnect_pull_supplier

3.21.4 CosNotifyChannelAdmin Module

This section describes the CosNotifyChannelAdmin module.

3.21.4.1 org.omg.CosNotifyChannelAdmin.ConsumerAdmin.obtain_notification_pull_supplier()

Name

org.omg.CosNotifyChannelAdmin.ConsumerAdmin.obtain_notification_pull_supplier

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
public  interface ConsumerAdmin extends org.omg.CORBA.Object {
      public org.omg.CosNotifyChannelAdmin.ProxySupplier obtain_notification_pull_supplier (
          org.omg.CosNotifyChannelAdmin.ClientType    ctype,
          org.omg.CORBA.IntHolder                     proxy_id )
              throws org.omg.CosNotifyChannelAdmin.AdminLimitExceeded;
  }
```

Description

Creates a ProxySupplier object with the client type specified in ctype. Sets in proxy_id the ProxySupplier ID that was created, and returns the ProxySupplier object reference

Parameters

ctype

The client type by which the object of ProxySupplier is made is specified as follows.

org.omg.CosNotifyChannelAdmin.ClientType.ANY_EVENT

Handles any type events

org.omg.CosNotifyChannelAdmin.ClientType.STRUCTURED_EVENT

Handles StructuredEvent type events

proxy_id (out parameter)

The identification ID of made ProxySupplier is set.

Return Values

Normal termination: Returns the object reference for the ProxySupplier.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosNotifyChannelAdmin.AdminLimitExceeded

  No more ProxySuppliers can be created because the upper limit has been reached.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.4.2 org.omg.CosNotifyChannelAdmin.SupplierAdmin.obtain_notification_push_consumer()

Name

org.omg.CosNotifyChannelAdmin.SupplierAdmin.obtain_notification_push_consumer

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface SupplierAdmin extends org.omg.CORBA.Object {
      public org.omg.CosNotifyChannelAdmin.ProxyConsumer obtain_notification_pull_supplier (
          org.omg.CosNotifyChannelAdmin.ClientType    ctype,
          org.omg.CORBA.IntHolder                     proxy_id )
              throws org.omg.CosNotifyChannelAdmin.AdminLimitExceeded;
  }
```

Description

Creates a ProxyConsumer object with the client type specified in ctype. Sets in proxy_id the ProxyConsumer ID that was created, and returns the ProxyConsumer object reference

Parameters

ctype

The client type by which the object of ProxyConsumer is made is specified as follows.

org.omg.CosNotifyChannelAdmin.ClientType.ANY_EVENT

Handles any type events

org.omg.CosNotifyChannelAdmin.ClientType.STRUCTURED_EVENT

Handles StructuredEvent type events

proxy_id (out parameter)

The identification ID of made ProxyConsumer is set.

Return Values

Normal termination: Returns the object reference for the ProxyConsumer.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exceptions:

- org.omg.CosNotifyChannelAdmin.AdminLimitExceeded

  No more ProxyConsumers can be created because the upper limit has been reached.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.4.3 org.omg.CosNotifyChannelAdmin.ConsumerAdmin.MyChannel()

Name

org.omg.CosNotifyChannelAdmin.ConsumerAdmin.MyChannel

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface ConsumerAdmin extends org.omg.CORBA.Object {
      public  org.omg.CosNotifyChannelAdmin.EventChannel MyChannel ();

  }
```

Description

Fetches the object reference for the event channel that generated ConsumerAdmin.

Return Values

Normal termination: Returns the object reference for the event channel that generated ConsumerAdmin.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.4.4 org.omg.CosNotifyChannelAdmin.SupplierAdmin.MyChannel()

Name

org.omg.CosNotifyChannelAdmin.SupplierAdmin.MyChannel

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface SupplierAdmin extends org.omg.CORBA.Object {
      public  org.omg.CosNotifyChannelAdmin.EventChannel MyChannel ();

  }
```

Description

Fetches the object reference for the event channel that generated SupplierAdmin.

Return Values

Normal termination: Returns the object reference for the event channel that generated SupplierAdmin.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.4.5 org.omg.CosNotifyChannelAdmin.EventChannel.default_consumer_admin()

Name

org.omg.CosNotifyChannelAdmin.EventChannel.default_consumer_admin

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface EventChannel extends org.omg.CORBA.Object {
      public  org.omg.CosNotifyChannelAdmin.ConsumerAdmin default_consumer_admin ();

  }
```

Description

Fetches the object reference for the ConsumerAdmin object that has the event channel as standard.

Return values

Normal termination: Returns the object reference for ConsumerAdmin.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.4.6 org.omg.CosNotifyChannelAdmin.EventChannel.default_supplier_admin()

Name

org.omg.CosNotifyChannelAdmin.EventChannel.default_supplier_admin

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface EventChannel extends org.omg.CORBA.Object {
      public  org.omg.CosNotifyChannelAdmin.SupplierAdmin default_supplier_admin ();

  }
```

Description

Fetches the object reference for the SupplierAdmin object that has the event channel as standard.

Return Values

Normal termination: Returns the object reference for SupplierAdmin.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.4.7 org.omg.CosNotifyChannelAdmin.ProxyPushConsumer.connect_any_push_supplier()

Name

org.omg.CosNotifyChannelAdmin.ProxyPushConsumer.connect_any_push_supplier

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface ProxyPushConsumer extends org.omg.CORBA.Object {
      public  void connect_any_push_supplier (
          org.omg.CosEventComm.PushSupplier   push_supplier)
              throws org.omg.CosEventChannelAdmin.AlreadyConnected
  }
```

Description

Connects any type supplier to the event channel.

Parameters

push_supplier

The own object reference

Specify null in push_supplier if no message is required when the event channel is closed.

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosEventChannelAdmin.AlreadyConnected

  The event channel is already connected.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

Note

When reconnecting to the event channel, start from org.omg.CosNotifyChannelAdmin.ConsumerAdmin.obtain_notification_push_consumer.

3.21.4.8 org.omg.CosNotifyChannelAdmin.ProxyPullSupplier.connect_any_pull_consumer()

Name

org.omg.CosNotifyChannelAdmin.ProxyPullSupplier.connect_any_pull_consumer

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface ProxyPullSupplier extends org.omg.CORBA.Object {
      public  void connect_any_pull_consumer (
          org.omg.CosEventComm.PullConsumer   pull_consumer )
              throws org.omg.CosEventChannelAdmin.AlreadyConnected;
}
```

Description

Connects any type consumer to the event channel.

Parameters

pull_consumer

The own object reference

Specify null in pull_consumer if no message is required when the event channel is closed.

Return values

Normal termination: None.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosEventChannelAdmin.AlreadyConnected

  The event channel is already connected.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

Note

When reconnecting to the event channel, start from org.omg.CosNotifyChannelAdmin.ConsumerAdmin.obtain_notification_pull_supplier.

3.21.4.9 org.omg.CosNotifyChannelAdmin.ProxyConsumer.MyType()

Name

org.omg.CosNotifyChannelAdmin.ProxyConsumer.MyType

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface ProxyConsumer extends org.omg.CORBA.Object {
      public  org.omg.CosNotifyChannelAdmin.ProxyType MyType ();

  }
```

Description

Returns the following values as Proxy object types:

org.omg.CosNotifyChannelAdmin.ProxyType.PUSH_ANY

any type PUSH model

org.omg.CosNotifyChannelAdmin.ProxyType.PULL_ANY

any type PULL model

org.omg.CosNotifyChannelAdmin.ProxyType.PUSH_STRUCTURED

StructuredEvent type PUSH model

org.omg.CosNotifyChannelAdmin.ProxyType.PULL_STRUCTURED

StructuredEvent type PULL model

Return Values

Normal termination: Returns the Proxy object type.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.4.10 org.omg.CosNotifyChannelAdmin.ProxySupplier.MyType()

Name

org.omg.CosNotifyChannelAdmin.ProxySupplier.MyType

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface ProxySupplier extends org.omg.CORBA.Object {
      public  org.omg.CosNotifyChannelAdmin.ProxyType MyType ();

  }
```

Description

Returns the following values as Proxy object types:

org.omg.CosNotifyChannelAdmin.ProxyType.PUSH_ANY

any type PUSH model

org.omg.CosNotifyChannelAdmin.ProxyType.PULL_ANY

any type PULL model

org.omg.CosNotifyChannelAdmin.ProxyType.PUSH_STRUCTURED

StructuredEvent type PUSH model

org.omg.CosNotifyChannelAdmin.ProxyType.PULL_STRUCTURED

StructuredEvent type PULL model

Return Values

Normal termination: Returns the Proxy object type.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.4.11 org.omg.CosNotifyChannelAdmin.StructuredProxyPushConsumer.connect_structured_push_supplier()

Name

org.omg.CosNotifyChannelAdmin.StructuredProxyPushConsumer.connect_structured_push_supplier

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface StructuredProxyPushConsumer extends org.omg.CORBA.Object {
      public  void connect_structured_push_supplier (
          org.omg.CosNotifyComm.StructuredPushSupplier    push_supplier  )
              throws org.omg.CosEventChannelAdmin.AlreadyConnected;
  }
```

Description

Connects StructuredEvent type supplier to the event channel.

Parameters

push_supplier

The own object reference

If no message is required when the event channel is closed, specify null in push_supplier.

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosEventChannelAdmin.AlreadyConnected

  The event channel is already connected.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

Note

When reconnecting to the event channel, start from org.omg.CosNotifyChannelAdmin.SupplierAdmin.obtain_notification_push_consumer.

3.21.4.12 org.omg.CosNotifyChannelAdmin.StructuredProxyPullSupplier.connect_structured_pull_consumer()

Name

org.omg.CosNotifyChannelAdmin.StructuredProxyPullSupplier.connect_structured_pull_consumer

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface StructuredProxyPullSupplier extends org.omg.CORBA.Object {
        public  void connect_structured_pull_consumer (
              org.omg.CosNotifyComm.StructuredPullConsumer    pull_consumer  )
                  throws org.omg.CosEventChannelAdmin.AlreadyConnected;
  }
```

Description

Connects StructuredEvent type consumer to the event channel.

Parameters

pull_consumer

The own object reference

If no message is required when the event channel is closed, specify null in pull_consumer.

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosEventChannelAdmin.AlreadyConnected

  The event channel is already connected.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

Note

When reconnecting to the event channel, start from org.omg.CosNotifyChannelAdmin.SupplierAdmin.obtain_notification_pull_supplier.

3.21.4.13 Interfaces that can be Inherited

The following interfaces can be inherited.

Refer to 3.21.3 CosNotifyComm Module for details.

1. org.omg.CosNotifyChannelAdmin.StructuredProxyPushConsumer.push_structured_event
2. org.omg.CosNotifyChannelAdmin.StructuredProxyPullSupplier.pull_structured_event
3. org.omg.CosNotifyChannelAdmin.StructuredProxyPullSupplier.try_pull_structured_event
4. org.omg.CosNotifyChannelAdmin.StructuredProxyPushConsumer.disconnect_structured_push_consumer
5. org.omg.CosNotifyChannelAdmin.StructuredProxyPullSupplier.disconnect_structured_pull_supplier
6. org.omg.CosNotifyChannelAdmin.ProxyPushConsumer.push
7. org.omg.CosNotifyChannelAdmin.ProxyPullSupplier.pull
8. org.omg.CosNotifyChannelAdmin.ProxyPullSupplier.try_pull
9. org.omg.CosNotifyChannelAdmin.ProxyPushConsumer.disconnect_push_consumer
10. org.omg.CosNotifyChannelAdmin.ProxyPullSupplier.disconnect_pull_supplier

3.21.5 EventChannelFactory Interface

org.omg.CosNotifyChannelAdmin.EventChannelFactory is a factory interface regulated by the OMG-NotificationService. When NotificationService APIs are used, the normal EventFactory interface cannot be used.

3.21.5.1 org.omg.CosNotifyChannelAdmin.EventChannelFactory.create_channel()

Name

org.omg.CosNotifyChannelAdmin.EventChannelFactory.create_channel

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface EventChannelFactory extends org.omg.CORBA.Object {
        public  org.omg.CosNotifyChannelAdmin.EventChannel create_channel (
              org.omg.CosNotification.Property[]     initial_qos,
              org.omg.CosNotification.Property[]     initial_admin,
              org.omg.CORBA.IntHolder                id )
                  throws org.omg.CosNotification.UnsupportedQoS,
                         org.omg.CosNotification.UnsupportedAdmin;
  }
```

Description

Generates an event channel with the QoS property item and Admin property item specified in initial_qos and initial_admin, and sets the event channel ID in id. Returns the event channel object reference as a return value.

Specify the following value in identifier method in org.omg.CORBA.ORB.resolve_initial_references() to fetch the factory object reference: NotificationService

Parameters

initial_qos

The QoS property item which generates event channel.

initial_admin

The Admin property item which generates event channel.

id (out parameter)

The identification ID of the event channel is set.

Return Values

Normal termination: Returns the object reference of event channel.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosNotification.UnsupportedQoS

  There is an error in the specified QoS property item or its value.

- org.omg.CosNotification.UnsupportedAdmin

  There is an error in the specified Admin property item or its value.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.5.2 org.omg.CosNotifyChannelAdmin.EventChannelFactory.get_all_channels()

Name

org.omg.CosNotifyChannelAdmin.EventChannelFactory.get_all_channels

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface EventChannelFactory extends org.omg.CORBA.Object {
      public  int[]  get_all_channels ();

  }
```

Description

Returns in sequence all IDs for managed event channels

Return Values

Normal termination: Returns in sequence all IDs for managed event channels.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.21.5.3 org.omg.CosNotifyChannelAdmin.EventChannelFactory.get_event_channel()

Name

org.omg.CosNotifyChannelAdmin.EventChannelFactory.get_event_channel

Format

```
import  org.omg.CosNotifyChannelAdmin.*;
  public  interface EventChannelFactory extends org.omg.CORBA.Object {
      public  org.omg.CosNotifyChannelAdmin.EventChannel get_event_channel (
          int  id )
          throws org.omg.CosNotifyChannelAdmin.ChannelNotFound;
  }
```

Description

Returns the object reference for the event channel with the ID specified in id.

Parameters

id

Event channel object reference ID to be acquired

Return Values

Normal termination: Returns the object reference for the event channel with the ID specified in id.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosNotifyChannelAdmin.ChannelNotFound

  The specified channel does not exist.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.22 Connection Information Fetch Function Interfaces

These interfaces are used to fetch connection information for event channels.

Note

The connection information fetch function interface is not supported in the IIOP Service (Java EE client).

3.22.1 org.omg.CosEventChannelAdmin Class

This section describes the org.omg.CosEventChannelAdmin class.

3.22.1.1 org.omg.CosEventChannelAdmin.EventChannel.create_util()

Name

org.omg.CosEventChannelAdmin.EventChannel.create_util

Format

```
  import  org.omg.CosEventChannelAdmin.*;
  import  com.fujitsu.ObjectDirector.EventService.ES.*;
  public interface org.omg.CORBA.Object{
      org.omg.CORBA.object create_util(
      org.omg.CosEventChannelAdmin.EventChannel.Utiltype     type  );

  }
```

Description

Creates an object reference for the ES_ChannelUtil interface.

Parameters

type

The interface type

Value Specified in Type	Return Value Information
org.omg.CosEventChannelAdmin.EventChannel.Utiltype. CHANNEL_UTIL	Fetches the object reference for the ChannelUtil interface

Return Values

Normal termination: Returns the object reference for the ES_ChannelUtil interface.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.22.1.2 Interfaces that can be Inherited

The following interface can be inherited:

org.omg.CosNotifyChannelAdmin.EventChannel.create_util.

3.22.2 com.fujitsu.ObjectDirector.EventService.ES Class

This section describes the com.fujitsu.ObjectDirector.EventService.ES class.

3.22.2.1 com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.get_proxys()

Name

com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.get_proxys

Format

```
import  com.fujitsu.ObjectDirector.EventService.ES.*;
  import  com.fujitsu.ObjectDirector.EventService.ES.ChannelUtilPackage.*;
  public interface ChannelUtil extends org.omg.CORBA.Object{
      public ProxyData[] get_proxys(
                        ProxyKind  kind);

  }

  class ProxyData {
        org.omg.CORBA.Object   proxy;
        int                    time;
        int                    ipaddress;
        ProxyKind              kind;
  };
```

Description

Fetches connection information for consumers/suppliers connected to the event channel.

Parameters

kind

Thee type of Proxy object to be fetched.

The connection information fetched depends on the value specified in kind. The following table lists the values specified in type and the return values.

Table 3.3 Values Specified in Type and Return Values

Value Specified in Type	Return Value Information
ChannelUtilPackage.Proxykind.ALL_PROXYS	Fetches all connection information
ChannelUtilPackage.Proxykind.PROXY_CONSUMER	Fetches all ProxyConsumer lists
ChannelUtilPackage.Proxykind.PROXY_SUPPLIER	Fetches all ProxySupplier lists
ChannelUtilPackage.Proxykind.PROXY_PULL_CONSUMER_EVENT	Fetches the CosEventChannelAdmin ProxyPullConsumer list
ChannelUtilPackage.Proxykind.PROXY_PULL_SUPPLIER_EVENT	Fetches the CosEventChannelAdmin ProxyPullSupplier list
ChannelUtilPackage.Proxykind.PROXY_PUSH_CONSUMER_EVENT	Fetches the CosEventChannelAdmin ProxyPushConsumer list
ChannelUtilPackage.Proxykind.PROXY_PUSH_SUPPLIER_EVENT	Fetches the CosEventChannelAdmin ProxyPushSupplier list
ChannelUtilPackage.Proxykind.PROXY_PULL_CONSUMER_NOTIFY	Fetches the CosNotifyChannelAdmin ProxyPullConsumer list
ChannelUtilPackage.Proxykind.PROXY_PULL_SUPPLIER_NOTIFY	Fetches the CosNotifyChannelAdmin ProxyPullSupplier list
ChannelUtilPackage.Proxykind.PROXY_PUSH_CONSUMER_NOTIFY	Fetches the CosNotifyChannelAdmin ProxyPushConsumer list
ChannelUtilPackage.Proxykind.PROXY_PUSH_SUPPLIER_NOTIFY	Fetches the CosNotifyChannelAdmin ProxyPushSupplier list
ChannelUtilPackage.Proxykind.STRUCTURED.PROXY_PULL_CONSUMER	Fetches the CosNotifyChannelAdmin StructuredProxyPullConsumer list
ChannelUtilPackage.Proxykind.STRUCTURED.PROXY_PULL_SUPPLIER	Fetches the CosNotifyChannelAdmin StructuredProxyPullSupplier list
ChannelUtilPackage.Proxykind.STRUCTURED.PROXY_PUSH_CONSUMER	Fetches the CosNotifyChannelAdmin StructuredProxyPushConsumer list
ChannelUtilPackage.Proxykind.STRUCTURED.PROXY_PUSH_SUPPLIER	Fetches the CosNotifyChannelAdmin StructuredProxyPushSupplier list

Set the values shown in the following table in the ProxyData structure members:

Table 3.4 ProxyData Structure Members and Values Set

Member	Value Set
Proxy	Object references for consumers/suppliers connected to the event channel
Time	Number of times connected to the event channel
ipaddress	IP addresses of consumers/suppliers connected to the event channel
kind	Proxy types for consumers/suppliers connected to the event channel: ChannelUtilPackage.Proxykind.PROXY_PULL_CONSUMER_EVENT ChannelUtilPackage.Proxykind.PROXY_PULL_SUPPLIER_EVENT ChannelUtilPackage.Proxykind.PROXY_PUSH_CONSUMER_EVENT

Member	Value Set
	ChannelUtilPackage.Proxykind.PROXY_PUSH_SUPPLIER_EVENT
	ChannelUtilPackage.Proxykind.PROXY_PULL_CONSUMER_NOTIFY
	ChannelUtilPackage.Proxykind.PROXY_PULL_SUPPLIER_NOTIFY
	ChannelUtilPackage.Proxykind.PROXY_PUSH_CONSUMER_NOTIFY
	ChannelUtilPackage.Proxykind.PROXY_PUSH_SUPPLIER_NOTIFY
	ChannelUtilPackage.Proxykind.STRUCTURED_PROXY_PULL_CONSUMER
	ChannelUtilPackage.Proxykind.STRUCTURED_PROXY_PULL_SUPPLIER
	ChannelUtilPackage.Proxykind.STRUCTURED_PROXY_PUSH_CONSUMER
	ChannelUtilPackage.Proxykind.STRUCTURED_PROXY_PUSH_SUPPLIER

Return Values

Normal termination: Returns the connection information for the consumer/supplier connected to the event channel.

Abnormal termination: Issues an exception.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

Notes

The 'Ipaddress' member is set to 0 when using the IP address in an IPv6 environment. In an IPv6 environment, use com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.get_proxys6().

3.22.2.2 com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.get_proxys6()

Name

com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.get_proxys6()

Synopsis

```
import  com.fujitsu.ObjectDirector.EventService.ES.*;
import  com.fujitsu.ObjectDirector.EventService.ES.ChannelUtilPackage.*;
public interface ChannelUtil extends org.omg.CORBA.Object{
    public ProxyData6[] get_proxys6(
                                ProxyKind  kind);
}

class ProxyData6 {
      org.omg.CORBA.Object   proxy;
      int                    time;
      byte[]                 ipaddress;
      int                    ip_format;
      ProxyKind              kind;
};
```

Description

Fetches consumers/suppliers connected to the event channel, and connection information.

When 'ipaddress' is set via the IPv6 form, 'ip_format' is set to 1. When 'ipaddress' is set via the IPv4 form, 'ip_format' is set to 0.

Parameters

kind

The type of Proxy object fetched.

The connection information fetched depends on the value specified in *kind*. For details of values that can be specified in *kind*, and the corresponding return inforrmation, refer to the Table 3.3 Values Specified in Type and Return Values lists under 3.22.2.1 com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.get_proxys().

Return Values

For normal termination, the connection information for the consumer/supplier connected to the event channel is returned.

Abnormal termination: Issues an exception.

For system exceptions, refer to "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" in the Messages manual to correct the problem.

Notes

The IPv4 client member 'Ipaddress' is set to the parallel IPv4 address when operating using the IP address in an IPv6 environment. For details of settings when operating in an IPv6 environment, refer to 'Operating in an IPv6 Environment' and 'config'(IP-version parameter) under 'CORBA Service Environment Definition' of the Tuning Guide.

3.22.2.3 com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.get_consumer_count()

Name

com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.get_consumer_count

Format

```
import  com.fujitsu.ObjectDirector.EventService.ES.*;
  public interface ChannelUtil extends org.omg.CORBA.Object{
      int get_consumer_count();

  }
```

Description

Fetches the number of consumers connected to the event channel.

Return Values

Normal termination: Returns the number of consumers connected to the event channel.

Abnormal termination: Issues an exception.

For system exceptions, refer to "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" in the Messages manual to correct the problem.

3.22.2.4 com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.get_supplier_count()

Name

com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.get_supplier_count

Format

```
import  com.fujitsu.ObjectDirector.EventService.ES.*;
  public interface ChannelUtil extends org.omg.CORBA.Object{
      int get_supplier_count();
```

```
  }
```

Description

Fetches the number of suppliers connected to the event channel.

Return Values

Normal termination: Returns the number of suppliers connected to the event channel.

Abnormal termination: Issues an exception.

For system exceptions, refer to "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" in the Messages manual to correct the problem.

3.22.2.5 com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.get_queue_length()

Name

com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.get_queue_length

Format

```
import  com.fujitsu.ObjectDirector.EventService.ES.*;
  public interface ChannelUtil extends org.omg.CORBA.Object{
      int get_queue_length();

  }
```

Description

Fetches the number of event data items queued on the event channel.

Return Values

Normal termination: Returns the number of event data items queued on the event channel.

Abnormal termination: Issues an exception.

For system exceptions, refer to "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" in the Messages manual to correct the problem.

3.22.2.6 com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_begin()

Name

com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_begin

Format

```
  import  com.fujitsu.ObjectDirector.EventService.ES.*;
  import  com.fujitsu.ObjectDirector.EventService.ES.ChannelUtilPackage.*;
  public  interface ChannelUtil extends org.omg.CORBA.Object {
      public  void local_begin(org.omg.CORBA.Object  proxy);
  }
```

Description

Notifies the channel of starting the local transaction.

For proxy, specifies the object reference of the channel fetched by obtain.

In the case of consumer, the following methods can be issued before com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_commit() or com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_rollback() is issued.

- org.omg.CosEventComm.PullSupplier.pull()
- org.omg.CosEventComm.PullSupplier.try_pull()
- org.omg.CosNotifyComm.StructuredPullSupplier.pull_structured_event()
- org.omg.CosNotifyComm.StructuredPullSupplier.try_pull_structured_event()

In the case of supplier, the following methods can be issued before com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_commit() or com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_rollback() is issued.

- org.omg.CosEventComm.PushConsumer.push()
- org.omg.CosNotifyComm.StructuredPushConsumer.push_structured_event()

If com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_commit() or com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_rollback() is not issued after issuance of this method, they are automatically rolled back by the notification service immediately after the local transaction timeout time passes.

This method cannot be issued twice for a proxy.

Parameters

proxy

Channel object reference acquired by the org.omg.CosNotifyChannelAdmin.ConsumerAdmin.obtain_notification_pull_supplier() or the org.omg.CosNotifyChannelAdmin.SupplierAdmin.obtain_notification_push_consumer().

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

For system exceptions, refer to "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" in the Messages manual to correct the problem.

3.22.2.7 com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_commit()

Name

com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_commit

Format

```
import  com.fujitsu.ObjectDirector.EventService.ES.*;
import  com.fujitsu.ObjectDirector.EventService.ES.ChannelUtilPackage.*;
public  interface ChannelUtil extends org.omg.CORBA.Object {
    public  void local_commit(org.omg.CORBA.Object  proxy);
}
```

Description

Notifies the channel of completing the local transaction.

In the case of consumer, message reception from the event channel completes as this method completes.

In the case of supplier, sending of data from the event channel completes as this method completes.

Parameters

proxy

Channel object reference acquired by the org.omg.CosNotifyChannelAdmin.ConsumerAdmin.obtain_notification_pull_supplier() or the org.omg.CosNotifyChannelAdmin.SupplierAdmin.obtain_notification_push_consumer().

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

For system exceptions, refer to "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" in the Messages manual to correct the problem.

3.22.2.8 com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_rollback()

Name

com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_rollback

Format

```
import  com.fujitsu.ObjectDirector.EventService.ES.*;
import  com.fujitsu.ObjectDirector.EventService.ES.ChannelUtilPackage.*;
public  interface ChannelUtil extends org.omg.CORBA.Object {
    public  void local_rollback(org.omg.CORBA.Object  proxy);
}
```

Description

Notifies the channel of cancellation of the local transaction. Issuance of this method allows the event channel status to return to that before com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.local_begin() is issued.

Parameters

proxy

Channel object reference acquired by the org.omg.CosNotifyChannelAdmin.ConsumerAdmin.obtain_notification_pull_supplier() or the org.omg.CosNotifyChannelAdmin.SupplierAdmin.obtain_notification_push_consumer().

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

For system exceptions, refer to "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" in the Messages manual to correct the problem.

3.22.2.9 com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.pull_cancel()

Name

com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.pull_cancel

Format

```
import  com.fujitsu.ObjectDirector.EventService.ES.*;
import  com.fujitsu.ObjectDirector.EventService.ES.ChannelUtilPackage.*;
public  interface ChannelUtil extends org.omg.CORBA.Object {
    public  void pull_cancel(org.omg.CORBA.Object  proxy);
}
```

Description

This method cancels the pull waiting status (waiting status by issuance of pull and com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.pull_wait()) of the target proxy.

Parameters

proxy

Channel object reference acquired by the org.omg.CosNotifyChannelAdmin.ConsumerAdmin.obtain_notification_pull_supplier().

Note

Specify the object reference that was specified in pull and com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.pull_wait().

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

For system exceptions, refer to "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" in the Messages manual to correct the problem.

3.22.2.10 com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.pull_wait()

Name

com.fujitsu.ObjectDirector.EventService.ES.ChannelUtil.pull_wait

Format

```
import  com.fujitsu.ObjectDirector.EventService.ES.*;
import  com.fujitsu.ObjectDirector.EventService.ES.ChannelUtilPackage.*;
public  interface ChannelUtil extends org.omg.CORBA.Object {
    public  void pull_wait(org.omg.CORBA.Object  proxy)
        throws org.omg.CosEventComm.Disconnected;
}
```

Description

This method waits until the status of the target proxy becomes enabled to allow the proxy to fetch data with the pull method or the try_pull method. Returns to the invoking side when the status becomes enabled to allow fetching data after execution of this method.

Returns immediately if the target proxy has already been enabled to fetch data with the pull method. If the status does not become enabled to allow fetching data within the waiting time, this method returns with timeout.

Parameters

proxy

Channel object reference acquired by the org.omg.CosNotifyChannelAdmin.ConsumerAdmin.obtain_notification_pull_supplier().

Return Values

Normal termination: None.

Abnormal termination: Issues an exception.

In the event of a user exception, issues the following exception:

- org.omg.CosEventComm.Disconnected

  Not connected to an event channel.

  If processing for automatic collection of connection information is enabled when the event channel is generated (the -autodiscon option is specified when the esmkchnl command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

For system exceptions, refer to 'Exception Information Minor Codes Reported from the Event Service' and 'Exception Information Minor Codes Reported from the CORBA Service' in the Messages manual, and correct the problem.

3.23 Current Class Windows32/64 Solaris32 Linux32/64

This section describes the current class.

Note

The current class is not supported in the IIOP Service (Java EE client).

3.23.1 org.omg.CosTransactions.Current.begin Windows32/64 Solaris32 Linux32/64

Name

org.omg.CosTransactions.Current.begin

Format

```
public interface Current extends org.omg.CORBA.Object {
public void begin()
 throws org.omg.CORBA.Exception;
}
```

Description

Generates a new transaction. Obtain the current interface object reference with the org.omg.CORBA.ORB.resolve_initial_references method (org.omg.CORBA.ORB.ObjectId_TransactionCurrent).

If you do not set timeout with org.omg.CosTransactions.Current.set_timeout, the system will use the value specified in the operation environment file's TRAN_TIME_OUT.

Return Values

Abnormal termination: Issues a org.omg.CORBA.Exception.

User exception: Issues the following exception:

- org.omg.CosTransactions.SubtransactionsUnavailable

  You tried to generate a nested transaction.

Systemexception: Issues the following exception:

- org.omg.CORBA.NO_IMPLEMENT

  The Database Linkage Service system is not implemented.

- org.omg.CORBA.COMM_FAILURE

  The following causes are possible.

  There has been a communications failure.

  Database Linkage Service host information is not set. Check the CORBA service environment file: initial_hosts (for PCs inithost).

- org.omg.CORBA.NO_RESOURCES

  Insufficient resource error. Possibly exceeded the maximum number of transactions.

- org.omg.CORBA.NO_MEMORY

  Fetching of dynamic memory failed.

3.23.2 org.omg.CosTransactions.Current.commit Windows32/64 Solaris32 Linux32/64

Name

org.omg.CosTransactions.Current.commit

Format

```
public interface Current extends org.omg.CORBA.Object {
public void  commit(
            boolean           report_heuristics)
   throws   org.omg.CORBA.Exception;
}
```

Description

Commits a transaction. Obtain the current interface object reference with the org.omg.CORBA.ORB.resolve_initial_references method (org.omg.CORBA.ORB.ObjectId_TransactionCurrent).

Set report_heuristics to true to get a heuristic query.

Parameters

report_heuristics

If "true" is specified, a heuristic error is indicated.

Return Values

Abnormal termination: Issues a org.omg.CORBA.Exception.

Userexception: Issues the following exception:

- org.omg.CosTransactions.NoTransaction

  No transaction has been generated.

- org.omg.CosTransactions.HeuristicMixed

  In transaction complete status, some are in commit and some are in rollback status.

- org.omg.CosTransactions.HeuristicHazard

  Transaction complete status is unknown.

Systemexception: Issues the following exception:

- org.omg.CORBA.NO_PERMISSION

  No permission to commit a transaction.

- org.omg.CORBA.TRANSACTION_ROLLEDBACK

  Transaction has been rolled back.

- org.omg.CORBA.NO_IMPLEMENT

  Either the Database Linkage Service system or the resource management program is not implemented.

- org.omg.CORBA.COMM_FAILURE

  There has been a communications failure.

- org.omg.CORBA.NO_RESOURCES

  Insufficient resource error. Possibly exceeded the maximum number of transactions.

- org.omg.CORBA.NO_MEMORY

  Fetching of dynamic memory failed.

3.23.3 org.omg.CosTransactions.Current.rollback 

Name

org.omg.CosTransactions.Current.rollback

Format

```
public interface Current extends org.omg.CORBA.Object {
public void rollback()
        throws   org.omg.CORBA.Exception;
}
```

Description

Rolls back a transaction. Obtain the current interface object reference with the org.omg.CORBA.ORB.resolve_initial_references method (org.omg.CORBA.ORB.ObjectId_TransactionCurrent).

Return Values

Abnormal termination: Issues a org.omg.CORBA.Exception.

Userexception: Issues the following exception:

- org.omg.CosTransactions.NoTransaction

  No transaction has been generated.

Systemexception: Issues the following exception:

- org.omg.CORBA.NO_PERMISSION

  No permission to rollback a transaction.

- org.omg.CORBA.TRANSACTION_ROLLEDBACK

  Transaction has been rolled back.

- org.omg.CORBA.NO_IMPLEMENT

  Either the Database Linkage Service system or the resource management program is not implemented.

- org.omg.CORBA.COMM_FAILURE

  There has been a communications failure.

- org.omg.CORBA.NO_RESOURCES

  There are insufficient resources.

- org.omg.CORBA.NO_MEMORY

  Fetching of dynamic memory failed.

3.23.4 org.omg.CosTransactions.Current.rollback_only Windows32/64 Solaris32 Linux32/64

Name

org.omg.CosTransactions.Current.rollback_only

Format

```
public interface Current extends org.omg.CORBA.Object {
public void rollback_only()
     throws   org.omg.CORBA.Exception;
}
```

Description

Only gives permission to rollback a transaction. Obtain the current interface object reference with the org.omg.CORBA.ORB.resolve_initial_references method (org.omg.CORBA.ORB.ObjectId_TransactionCurrent).

Return Values

Abnormal termination: Issues a org.omg.CORBA.Exception.

Userexception: Issues the following exception:

- org.omg.CosTransactions.NoTransaction

  No transaction has been generated.

Systemexception: Issues the following exception:

- org.omg.CORBA.TRANSACTION_ROLLEDBACK

  Transaction has been rolled back.

- org.omg.CORBA.NO_IMPLEMENT

  The Database Linkage Service system is not implemented.

- org.omg.CORBA.COMM_FAILURE

  There has been a communications failure.

- org.omg.CORBA.NO_RESOURCES

  There are insufficient resources.

- org.omg.CORBA.NO_MEMORY

  Fetching of dynamic memory failed.

3.23.5 org.omg.CosTransactions.Current.get_status Windows32/64 Solaris32 Linux32/64

Name

org.omg.CosTransactions.Current.get_status

Format

```
public interface Current extends org.omg.CORBA.Object {
public CosTransactions.Status get_status()
      throws   org.omg.CORBA.Exception;
}
```

Description

Returns transaction status. Obtain the current interface object reference with the org.omg.CORBA.ORB.resolve_initial_references method (org.omg.CORBA.ORB.ObjectId_TransactionCurrent).

Return Values

Normal termination: Notifies org.omg.CosTransaction.Status.

- org.omg.CosTransactions.Status.StatusActive

  Transaction is still being processed.

- org.omg.CosTransactions.Status.StatusMarkedRollback

  Marked for rollback.

- org.omg.CosTransactions.Status.StatusPrepared

  prepared
- org.omg.CosTransactions.Status.StatusCommited

  committed
- org.omg.CosTransactions.Status.StatusMarkedRollback

  rolled back
- org.omg.CosTransactions.Status.StatusUnknown

  Transaction status is unknown.
- org.omg.CosTransactions.Status.StatusNoTransaction

  Transaction not being processed.
- org.omg.CosTransactions.Status.StatusPreparing

  Preparing
- org.omg.CosTransactions.Status.StatusCommitting

  Committing
- org.omg.CosTransactions.Status.StatusRollingBack

  Rolling back

Abnormal termination: Issues a org.omg.CORBA.Exception.

Systemexception: Issues the following exception:

- org.omg.CORBA.TRANSACTION_ROLLEDBACK

  Transaction has been rolled back.
- org.omg.CORBA.NO_IMPLEMENT

  The Database Linkage Service system is not implemented.
- org.omg.CORBA.COMM_FAILURE

  There has been a communications failure.
- org.omg.CORBA.NO_RESOURCES

  There are insufficient resources.
- org.omg.CORBA.NO_MEMORY

  Fetching of dynamic memory failed.

3.23.6 org.omg.CosTransactions.Current.get_transaction_name Windows32/64 Solaris32 Linux32/64

Name

org.omg.CosTransactions.Current.get_transaction_name

Format

```
public interface Current extends org.omg.CORBA.Object {
public java.lang.string get_transaction_name()
        throws   org.omg.CORBA.Exception;
}
```

Description

Returns a character string that identifies a transaction. Obtain the current interface object reference with the org.omg.CORBA.ORB.resolve_initial_references method (org.omg.CORBA.ORB.ObjectId_TransactionCurrent).

Return Values

Normal termination: Returns a string that identifies a transaction. Returns a NULL string if no transaction has been generated.

Abnormal termination: Issues a org.omg.CORBA.Exception.

SystemException: Issues the following exception:

- org.omg.CORBA.TRANSACTION_ROLLEDBACK

  Transaction has been rolled back.

- org.omg.CORBA.NO_IMPLEMENT

  The Database Linkage Service system is not implemented.

- org.omg.CORBA.COMM_FAILURE

  There has been a communications failure.

- org.omg.CORBA.NO_RESOURCES

  There are insufficient resources.

- org.omg.CORBA.NO_MEMORY

  Fetching of dynamic memory failed.

3.23.7 org.omg.CosTransactions.Current.set_timeout Windows32/64 Solaris32 Linux32/64

Name

org.omg.CosTransactions.Current.set_timeout

Format

```
public interface Current extends org.omg.CORBA.Object {
public void  set_timeout(int            seconds)
       throws   org.omg.CORBA.Exception;
}
```

Description

Specifies the time (seconds) for transaction timeout monitoring. If a transaction is not completed in the time specified by seconds the transaction will be rolled back.

If called during a transaction, the specified timeout value will become valid when the next transaction begins.

Obtain the current interface object reference with the org.omg.CORBA.ORB.resolve_initial_references method (org.omg.CORBA.ORB.ObjectId_TransactionCurrent).

Parameters

seconds

Specify a transaction time-out monitoring time. If 0 is set in the seconds parameter, time-out monitoring does not take place.

Return Values

Normal termination: None.

Abnormal termination: Issues a org.omg.CORBA.Exception.

System exception: Issues the following exception:

- org.omg.CORBA.NO_MEMORY

  Fetching of dynamic memory failed.

- org.omg.CORBA.INTERNAL

  CORBA service error.

3.23.8 org.omg.CosTransactions.Current.get_control Windows32/64 Solaris32 Linux32/64

Name

org.omg.CosTransactions.Current.get_control

Format

```
public interface Current extends org.omg.CORBA.Object {
public CosTransactions.Control.get_control()
      throws   org.omg.CORBA.Exception;
}
```

Description

Returns the Control object that manages the transaction text.

The Control object can be used to complete transactions in server applications.

Obtain the current interface object reference with the org.omg.CORBA.ORB.resolve_initial_references method (org.omg.CORBA.ORB.ObjectId_TransactionCurrent).

Return Values

Normal termination: Posts org.omg.CosTransactions.Control.

Abnormal termination: Issues an org.omg.CORBA.Exception.

System exception: Issues the following exception:

- org.omg.CORBA.NO_IMPLEMENT

  Database Linkage Service system could not be activated.

- org.omg.CORBA.COMM.FAILURE

  There was a communication error.

- org.omg.CORBA.NO_RESOURCES

  There are insufficient resources.

- org.omg.CORBA.NO_MEMORY

  Fetching of dynamic memory failed.

3.23.9 org.omg.CosTransactions.Current.suspend Windows32/64 Solaris32 Linux32/64

Name

org.omg.CosTransactions.Current.suspend

Format

```
public interface Current extends org.omg.CORBA.Object {
public CosTransactions.Control suspend()
```

```
      throws   org.omg.CORBA.Exception;
}
```

Description

Immediately cuts the relationship between the transaction generated by the application and the thread, and stops. Returns the Control object that manages the transaction context that manages the transaction.

When this function terminates normally, the transaction remains invalid until the resume function is run.

Obtain the current interface object reference with the org.omg.CORBA.ORB.resolve_initial_references method (org.omg.CORBA.ORB.ObjectId_TransactionCurrent).

Return Values

Normal termination: Posts org.omg.CosTransactions.Control.

Abnormal termination: Issues an org.omg.CORBA.Exception.

System exception: Issues the following exception:

- org.omg.CORBA.NO_MEMORY

  Fetching of dynamic memory failed.

- org.omg.CORBA.UNKNOWN

  A CORBA Service initialization error occurred.

- org.omg.CORBA.INTERNAL

  CORBA service error.

3.23.10 org.omg.CosTransactions.Current.resume

Name

org.omg.CosTransactions.Current.resume

Format

```
public interface Current extends org.omg.CORBA.Object {
public void  resume(org.omg.CosTransactions.Cntrol control)
       throws   org.omg.CORBA.Exception;
}
```

Description

Reestablishes the relationship between the transaction and the current thread.

Parameters

control

Specify the Control object to be linked. Obtain the reference to the Control object using the org.omg.CosTransactions.Current.get_control method. Alternatively use the Control object returned by the suspend function.

If a null is specified, the linkage between the transaction and current thread is cut off.

Return Values

Normal termination: None.

Abnormal termination: Issues an org.omg.CORBA.Exception.

User exception: Issues the following exception:

- org.omg.CosTransactions.InvalidControl

  An object other than a control object was specified.

System exception: Issues the following exception:

- org.omg.CORBA.INTERNAL

  CORBA service error.

3.24 Transaction Initialization Class Windows32/64 Solaris32 Linux32/64

This section describes the transaction initialization class that provides the interface for initializing Database Linkage Service transactions.

The transaction initialization class is not supported in the IIOP Service (Java EE client).

```
//Java
package com.fujitsu.interstage.ots;
public abstract class OTS
{
  public void
       init(org.omg.CORBA.ORB orb, java.lang.String impl, java.lang.String libname)
       throws com.fujitsu.interstage.ots.Bad_Description;
       throws com.fujitsu.interstage.ots.No_DefFile;
       throws com.fujitsu.interstage.ots.Rm_Error;
       throws com.fujitsu.interstage.ots.Permission_Denied;
       throws com.fujitsu.interstage.ots.IOError;
       throws com.fujitsu.interstage.ots.Not_OTS_Started;
  }
```

3.24.1 com.fujitsu.interstage.ots.OTS.init Windows32/64 Solaris32 Linux32/64

Name

com.fujitsu.interstage.ots.OTS.init

Format

```
public void  init(
                org.omg.CORBA.ORB orb,
                java.lang.String      impl,
                java.lang.String      libname )
                throws org.omg.CORBA.SystemException;
```

Description

Connects server applications to the Database Linkage Service system.

The org.omg.CORBA.ORB.resolve.initial.references method (org.omg.CORBA.ORB.ObjectId_TransactionServerInit) fetches the Database Linkage Service interface object reference.

Parameters

orb

Specifies the ORB object obtained by the org.omg.CORBA.ORB.init method

impl

Specifies the server application implementation repository ID string.

libname

Specifies the name of the XA linkage program that the server application uses when accessing the database. Use the file name excluding extension.

Return Values

Normal termination: None.

Abnormal termination: Issues an org.omg.CORBA.SystemException.

User exception: Issues the following exception:

- com.fujitsu.interstage.OTS.Bad_Description

  There is an error in the coding of the resource definition file.

- com.fujitsu.interstage.OTS.No_DefFile

  There is no resource definition file. Alternatively, the resource management program for the resource definition file is not registered in the Implementation Repository.

- com.fujitsu.interstage.OTS.Rm_Error

  An error occurred when the database was opened.

- com.fujitsu.interstage.OTS.Permission_Denied

  There is no permission set up for the resource definition file.

- com.fujitsu.interstage.OTS.IO_Error

  An I/O error occurred during reading of the resource definition file.

- com.fujitsu.interstage.OTS.Not_OTS_Started

  The Database Linkage Service system or the resource definition file has not been started.

Note

Do not call this function more than once in one process.

3.25 Transaction Termination Class

Windows32/64 Solaris32 Linux32/64

This section describes the transaction termination class that provides the interface for terminating Database Linkage Service transactions.

Note

The transaction termination class is not supported in the IIOP Service (Java EE client).

```
//Java
package com.fujitsu.interstage.ots;
public abstract class OTS
{
  public void
      term()
        throws com.fujitsu.interstage.ots.Rm_Error;
        throws com.fujitsu.interstage.ots.Permission_Denied;
        throws com.fujitsu.interstage.ots.IOError;
        throws com.fujitsu.interstage.ots.Occurred_File;
        throws com.fujitsu.interstage.ots.Not_OTS_Started;
   }
```

3.25.1 com.fujitsu.interstage.ots.OTS.term

Windows32/64 Solaris32 Linux32/64

Name

com.fujitsu.interstage.ots.OTS.term

Format

```
public void .term()
        throws org.omg.CORBA.SystemException;
```

Description

Disconnects server applications from the Database Linkage Service system.

The org.omg.CORBA.ORB.resolve.initial.references method (org.omg.CORBA.ORB.ObjectId_TransactionServerInit) fetches the Database Linkage Service interface object reference.

Return Values

Normal termination: None.

Abnormal termination: Issues an org.omg.CORBA.SystemException.

User exception: Issues the following exception:

- com.fujitsu.interstage.OTS.Rm_Error:

  An error occurred when the database was opened.

- com.fujitsu.interstage.OTS.Permission_Denied:

  There is no permission set up for the resource definition file.

- com.fujitsu.interstage.OTS.IO_Error:

  An I/O error occurred during reading of the resource definition file.

- com.fujitsu.interstage.OTS.Occurred_File:

  Therre is an error in the resource definition file.

- com.fujitsu.interstage.OTS.Not_OTS_Started:

  The Database Linkage Service system or the resource definition file has not been started.

Note

Do not call this function more than once in one process.

3.26 Continuation Transaction Initialization Class Windows32/64 Solaris32 Linux32/64

This section describes the continuation transaction initialization class that provides the interface for initializing Database Linkage Service continuation transactions.

Note

The continuation transaction initialization class is not supported in the IIOP Service (Java EE client).

```
//Java
package com.fujitsu.interstage.ots;
public abstract class OTScnt
{
  public void
          init(org.omg.CORBA.ORB orb, java.lang.String impl)
        throws com.fujitsu.interstage.ots.Bad_Description;
        throws com.fujitsu.interstage.ots.No_DefFile;
        throws com.fujitsu.interstage.ots.Rm_Error;
        throws com.fujitsu.interstage.ots.Permission_Denied;
        throws com.fujitsu.interstage.ots.IOError;
        throws com.fujitsu.interstage.ots.Not_OTS_Started;
  }
```

3.26.1 com.fujitsu.interstage.ots.OTScnt.init Windows32/64 Solaris32 Linux32/64

Name

com.fujitsu.interstage.ots.OTScnt.init

Format

```
public void  init(
                org.omg.CORBA.ORB orb,
                java.lang.String       impl)
             throws org.omg.CORBA.SystemException;
```

Description

Connects server applications to the Database Linkage Service system.

The org.omg.CORBA.ORB.resolve.initial.references method (org.omg.CORBA.ORB.ObjectId_TransactionServerInit) fetches the Database Linkage Service interface object reference.

Parameters

orb

Specifies the ORB object obtained by the org.omg.CORBA.ORB.init method

impl

Specifies the server application implementation repository ID string.

Return Values

Normal termination: None.

Abnormal termination: Issues an org.omg.CORBA.SystemException.

User exception: Issues the following exception:

- com.fujitsu.interstage.OTS.Bad_Description:

  There is an error in the coding of the resource definition file.

- com.fujitsu.interstage.OTS.No_DefFile:

  There is no resource definition file. Alternatively, the resource management program for the resource definition file is not registered in the Implementation Repository.

- com.fujitsu.interstage.OTS.Rm_Error:

  An error occurred when the database was opened.

- com.fujitsu.interstage.OTS.Permission_Denied:

  There is no permission set up for the resource definition file.

- com.fujitsu.interstage.OTS.IO_Error:

  An I/O error occurred during reading of the resource definition file.

- com.fujitsu.interstage.OTS.Not_OTS_Started:

  The Database Linkage Service system or the resource definition file has not been started.

Note

Do not call this function more than once in one process.

3.27 Continuation Transaction Termination Class Windows32/64 Solaris32 Linux32/64

This section describes the continuation transaction termination class that provides the interface for terminating Database Linkage Service continuation transactions.

Note

The continuation transaction termination class is not supported in the IIOP Service (Java EE client).

```
//Java
package com.fujitsu.interstage.ots;
public abstract class OTScnt
{
  public void
      term()
        throws com.fujitsu.interstage.ots.Rm_Error;
        throws com.fujitsu.interstage.ots.Permission_Denied;
        throws com.fujitsu.interstage.ots.IOError;
        throws com.fujitsu.interstage.ots.Occurred_File;
        throws com.fujitsu.interstage.ots.Not_OTS_Started;
   }
```

3.27.1 com.fujitsu.interstage.ots.OTScnt.term Windows32/64 Solaris32 Linux32/64

Name

com.fujitsu.interstage.ots.OTScnt.term

Format

```
public void .term()
        throws org.omg.CORBA.SystemException;
```

Description

Disconnects server applications from the Database Linkage Service system.

The org.omg.CORBA.ORB.resolve.initial.references method (org.omg.CORBA.ORB.ObjectId_TransactionServerInit) specifies the Database Linkage Service interface object reference.

Return Values

Abnormal termination: Issues an org.omg.CORBA.SystemException.

User exception: Issues the following exception:

- com.fujitsu.interstage.OTS.Rm_Error:

  An error occurred when the database was opened.

- com.fujitsu.interstage.OTS.Permission_Denied:

  There is no permission set up for the resource definition file.

- com.fujitsu.interstage.OTS.IO_Error:

  An I/O error occurred during reading of the resource definition file.

- com.fujitsu.interstage.OTS.Occurred_File:

  Therre is an error in the resource definition file.

- com.fujitsu.interstage.OTS.Not_OTS_Started:

  The Database Linkage Service system or the resource definition file has not been started.

Note

Do not call this function more than once in one process.

Chapter 4 COBOL Interface

This chapter describes the COBOL interface.

4.1 Libraries

This section describes the libraries required while developing client server programs in the ObjectDirector (OD).

Synopsis

```
CORBA-SHORT PIC S9(4) COMP-5.
CORBA-LONG PIC S9(9) COMP-5.
CORBA-UNSIGNED-SHORT PIC S9(4) COMP-5.
CORBA-UNSIGNED-LONG PIC 9(9) COMP-5.
CORBA-FLOAT COMP-1.
CORBA-DOUBLE COMP-2.
CORBA-CHAR PIC X(1).
CORBA-OCTET PIC S9(4) COMP-5.
CORBA-ENUM PIC 9(9) COMP-5.
CORBA-STRING USAGE IS POINTER.
CORBA-OBJECT USAGE IS POINTER.
CORBA-BOOLEAN PIC S9(9) COMP-5.

CORBA-SEQUENCE.
03  SEQ-MAXIMUM  PIC  S9(9) COMP-5.
03  SEQ-LENGTH   PIC  S9(9) COMP-5.
03  SEQ-BUFFER   USAGE POINTER.

CORBA-ANY.
49  ANY-TYPE   USAGE POINTER.
49  ANY-VALUE  USAGE POINTER.

CORBA-FLAGS PIC S9(9) COMP-5.

CORBA-FALSE IS 0
CORBA-TRUE  IS 1

CORBA-ENVIRONMENT.
03  MAJOR PIC 9(9) COMP-5.
88  CORBA-NO-EXCEPTION    VALUE 0.
88  CORBA-USER-EXCEPTION  VALUE 1.
88  CORBA-SYSTEM-EXCEPTION  VALUE 2.
03  IDL-ID       USAGE POINTER.
03  MINOR        PIC 9(9) COMP-5.
03  IDL-STATUS   PIC 9(9) COMP-5.
03  PARAM        USAGE POINTER.
03  MAGIC        PIC 9(9) COMP-5.
```

4.2 ORB Interface

This section describes the ORB (Object Request Broker) interface, which is the basic interface used to initialize ORB. The ORB function mediates communications between clients and servers.

4.2.1 CORBA-ORB-INIT

Name

CORBA-ORB-INIT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
ARGUMENT-NUMBER IS ARG-C
ARGUMENT-VALUE  IS ARG-V.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 ORB-IDENTIFIER USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ORB IN CORBA REPLACING CORBA-ORB BY ORB.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY C-ARG-C.
01 C-ARG-V.
02 FILLER OCCURS n.
03 C-ARG-V-VALUE USAGE POINTER.
01 APLI-NAME PIC X(M) VALUE "APPLICATION NAME".
```

PROCEDURE DIVISION.

```
CALL "CORBA-ORB-INIT" USING
C-ARG-C
C-ARG-V
ORB-IDENTIFIER
ENV
ORB.
```

Description

This function initializes ORB, and returns its object reference.

The object reference is used as a parameter when calling another ORB interface. Calling this function makes it possible to use the functions described in later sections. Use the object references returned in parameters when calling other ORB interfaces. To match interfaces when using COBOL and other languages, ARG-C and ARG-V (passed to the program as C-ARG-C and C-ARG-V) must be processed as shown below.

Setting the number of parameters

```
ACCEPT CURRENT-ARG-C FROM ARG-C.
COMPUTE CURRENT-ARG-C = CURRENT-ARG-C + 1.
```

Fetching and setting parameters

```
PERFORM VARYING ARG-COUNT FROM 1 BY 1 UNTIL ARG-COUNT > C-ARG-C
IF ARG-COUNT = 1
MOVE APLI-NAME TO TMP-STRING-BUF
ELSE
ACCEPT TMP-STRING-BUF FROM ARG-V
END-IF
MOVE FUNCTION LENG (TMP-STRING-BUF) TO STRING-LENGTH
CALL "CORBA-STRING-SET" USING
C-ARG-V-VALUE (ARG-COUNT)
STRING-LENGTH
TMP-STRING-BUF
END-PERFORM.
SET C-ARG-V-VALUE (ARG-COUNT) TO NULL.
```

ORB-IDENTIFIER is a name that identifies the ORB. Specify FJ-OM-CRBID in the OD name.

To fetch initial object references specific to clients with the CORBA-ORB-RESOLVE-INITIAL-REFERENCES function, values must be set in ARG_C and ARG-V as described in "Fetching Naming Service Initial References" in the Distributed Application Development Guide (CORBA Service Edition).

The functions described in the following sections can be used by invoking this function.

Return Values

For normal termination, the ORB object reference is set.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

Notes

This function must not be called by more than one process.

"n" and "m" in the format are explained below:

n: Number of startup parameters

m: Length of program name

For shared server and unshared server, the server application startup parameters (defined with param in the definition file specified by the *OD_impl_inst* command -ax option) are set after the following ORB control parameters in C-ARG-V:

First parameter:	"-ORB_FJ_PROC_ID"
Second parameter:	Process number
Third parameter:	"-ORB_FJ_HOME_DIR"
Fourth parameter:	Value set in $OD_HOME

4.2.2 CORBA-ORB-BOA-INIT

Name

CORBA-ORB-BOA-INIT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
ARGUMENT-NUMBER IS ARG-C
ARGUMENT-VALUE  IS ARG-V.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ORB IN CORBA REPLACING CORBA-ORB BY ORB.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 BOA-IDENTIFIER USAGE POINTER.
01 COPY BOA IN CORBA REPLACING CORBA-BOA BY BOA.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ORB-BOA-INIT" USING
ORB
```

```
ARG-C
ARG-V
BOA-IDENTIFIER
ENV
BOA.
```

Description

This function initializes BOA (Basic Object Adapter) that manages server applications, and returns the BOA object reference. Specify the object reference of the ORB returned by the CORBA-ORB-INIT function.

ARG-C and ARG-V are parameters used when executing the program, and are passed to the main entry point.

Specify CORBA-BOA-OA-ID in BOA-IDENTIFIER with the CORBA-STRING-SET function.

Return Values

For normal termination, the BOA object reference is returned, and CORBA-NO- EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If NULL is specified in ORB, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.2.3 CORBA-ORB-RESOLVE-INITIAL-REFERENCES

Name

CORBA-ORB-RESOLVE-INITIAL-REFERENCES

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY REPOSITORYID IN CORBA REPLACING CORBA-REPOSITORYID  BY REPOSITORY-ID.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ORB IN CORBA REPLACING CORBA-ORB BY ORB.
01 IDENTIFIER USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ORB-RESOLVE-INITIAL-REFERENCES" USING
ORB
IDENTIFIER
ENV
REPOSITORY-ID.
```

Description

This function returns an interface reference corresponding to an identifier (ID) indicating the object specified in IDENTIFIER.

The object reference is returned as described in "Fetching Naming Service Initial References" in the Distributed Application Development Guide (CORBA Service Edition).

The following can be used as IDENTIFIER:

CORBA-ORB-OBJECTID-LIGHTINTFR:

Interface Repository (static skeleton interface)

CORBA-ORB-OBJECTID-INTFREP:

Interface Repository (dynamic skeleton interface)

CORBA-ORB-OBJECTID-IMPLREP:

Implementation Repository

CORBA-ORB-OBJECTID-NAMESERVICE:

Naming Service

Specify the ORB reference returned by the CORBA-ORB-INIT function in ORB.

Invoke CORBA-STRING-SET to set the ID in IDENTIFIER.

Return Values

For normal termination, an object reference is set.

If the specified object does not exist, NULL is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If NULL is specified in ORB, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.2.4 CORBA-ORB-RESOLVE-INITIAL-REFERENCES-REMOTE

Name

CORBA-ORB-RESOLVE-INITIAL-REFERENCES-REMOTE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ORB IN CORBA REPLACING CORBA-ORB BY ORB.
01 COPY OBJECTID IN CORBA REPLACING CORBA-ORB-OBJECTID BY IDENTIFIER.
01 COPY REMOTEMODIFIER IN CORBA REPLACING CORBA-ORB-REMOTEMODIFIER BY M.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ORB-RESOLVE-INITIAL-REFERENCES-REMOTE" USING
         ORB
         IDENTIFIER
         M
         ENV
         OBJ.
```

Description

Returns the object reference for the object specified in IDENTIFIER. Object reference allocates the initial_services for the host defined in M in URL format.

The following can be used in IDENTIFIER:

CORBA-OBJECTID-LIGHTINTER

Interface Repository (static skeleton interface)

CORBA-OBJECTID-INTERREP

Interface Repository (dynamic skeleton interface)

CORBA-OBJECTID-IMPLREP

Implementation Repository

CORBA-ORB-OBJECTID-NAMESERVICE

Naming Service

Multiple URLs can be specified for M. In this case, the URLs are searched for in the order specified. When the object reference is found, the search is canceled.

The URL specification format is as follows:

iiop//<address>[:<port>]

<address>

The host name, DNS name, and IP address can be specified. They cannot be omitted.

<port>

Specify the port number of the connection destination ORB.

Return Values

On normal termination, returns the object reference of CORBA-OBJECT. If the specified object does not exist, NULL is returned.

On abnormal termination, the following exception is generated:

CORBA-ORB-INVALIDNAME

The object specified for the IDENTIFIER cannot be found.

CORBA-SYSTEM-EXCEPTION

Other causes

For details on the exception information and minor codes that are set, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.2.5 CORBA-ORB-LIST-INITIAL-SERVICES

Name

CORBA-ORB-LIST-INITIAL-SERVICES

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ORB IN CORBA REPLACING CORBA-ORB BY ORB.
01 COPY OBJECTIDLIST IN CORBA REPLACING CORBA-ORB-OBJECTIDLIST BY OBJECT-ID-LIST.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ORB-LIST-INITIAL-SERVICES" USING
ORB
ENV
OBJECT-ID-LIST.
```

Description

This function returns the OBJECT-ID-LIST list of usable objects. This list is stored in SEQ- BUFFER in the CORBA-ORB-OBJECTIDLIST structure by using character strings.

CORBA-SEQUENCE-STRING.

03 SEQ-MAXIMUM PIC S9(9) COMP-5.

03 SEQ-LENGTH PIC S9(9) COMP-5.

03 SEQ-BUFFER USAGE POINTER.

Specify the list returned by OBJECT-ID-LIST as a parameter to CORBA-ORB-RESOLVE-INITIAL-REFERENCES IDENTIFIER, to obtain an object reference corresponding to the list.

Specify the ORB object reference returned by the CORBA-ORB-INIT function in ORB.

Return Values

For normal termination, the list of OBJECT-ID-LIST is set.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If NULL is specified in ORB, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.2.6 CORBA-ORB-OBJECT-TO-STRING

Name

CORBA-ORB-OBJECT-TO-STRING

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY ORB IN CORBA REPLACING CORBA-ORB BY ORB.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 O-STRING USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ORB-OBJECT-TO-STRING" USING
ORB
OBJ
ENV
O-STRING.
```

Description

This function converts the object reference of an object specified by OBJ into a character string.

Specify the ORB reference returned by the CORBA-ORB-INIT function in ORB.

Since this function acquires area to store converted character strings, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the converted character string is set.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If NULL is specified in ORB or OBJ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.2.7 CORBA-ORB-STRING-TO-OBJECT

Name

CORBA-ORB-STRING-TO-OBJECT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ORB IN CORBA REPLACING CORBA-ORB BY ORB.
01 STR USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ORB-STRING-TO-OBJECT" USING
ORB
STR
ENV
OBJ.
```

Description

This function converts a character string specified by STR into an object reference.

All character string types listed in "corbaloc URL Schema" in the Distributed Application Development Guide (CORBA Service Edition) can be converted.

Specify the ORB reference returned by the CORBA-ORB-INIT function in ORB.

Since this function acquires area to store object references, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the converted object reference is set.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and details are set in ID and MINOR. The meaning and value of ID are as follows:

IDL:CORBA/BAD_PARAM:1.0

There is an error in the format of the specified URL schema.

For details on MINOR, refer to "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If NULL is specified in ORB, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.2.8 CORBA-ORB-CREATE-LIST

Name

CORBA-ORB-CREATE-LIST

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ORB IN CORBA REPLACING CORBA-ORB BY O01 COPY LONG IN CORBA
REPLACING CORBA-LONG BY C-COUNT.
01 NEW-LIST USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ORB-CREATE-LIST" USING
ORB
C-COUNT
NEW-LIST
ENV
ORB-STATUS.
```

Description

This function creates a list object (CORBA-NVLIST) to store parameters for the number of items specified by C-COUNT.

When a parameter is to be added to the list object created, invoke the CORBA-NVLIST-ADD-ITEM function (NVList interface).

Specify the ORB object reference returned by the CORBA-ORB-INIT function in ORB.

Return Values

For normal termination, CORBA-OK is returned, and the list object created is set in NEW- LIST.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in ORB or C-COUNT, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.2.9 CORBA-ORB-CREATE-OPERATION-LIST

Name

CORBA-ORB-CREATE-OPERATION-LIST

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ORB IN CORBA REPLACING CORBA-ORB BY ORB.
```

```
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OPER.
01 NEW-LIST USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ORB-CREATE-OPERATION-LIST" USING
ORB
OPER
NEW-LIST
ENV
ORB-STATUS.
```

Description

This function creates a list object (CORBA-NVLIST) initialized with parameter information for the specified operation in OPER.

The function returns list objects created in the same order as the parameters defined for the operation.

ORB acquires detailed operation information from the Interface Repository, based on the specified operation information.

Specify the ORB object reference returned by the CORBA-ORB-INIT function in ORB.

Return Values

For normal termination, CORBA-OK is returned, and the list object created is set in NEW- LIST.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in ORB or OPER, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.2.10 CORBA-ORB-GET-DEFAULT-CONTEXT

Name

CORBA-ORB-GET-DEFAULT-CONTEXT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ORB IN CORBA REPLACING CORBA-ORB BY ORB.
01 COPY CONTEXT IN CORBA REPLACING CORBA-CONTEXT BY CTX.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ORB-GET-CONTEXT-DEFAULT-CONTEXT" USING
ORB
CTX
ENV
ORB-STATUS.
```

Description

This function returns the default Context object reference.

Specify the ORB object reference returned by the CORBA-ORB-INIT function in ORB.

Return Values

For normal termination, CORBA-OK is returned, and the default Context object reference is set in CTX.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in ORB, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.2.11 CORBA-ORB-TYPECODE-FROM-CGEN-TC

Name

CORBA-ORB-TYPECODE-FROM-CGEN-TC

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 IMPL-SEQ-POINTER USAGE IS POINTER.
01 TYPE-POINTER USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ORB-TYPECODE-FROM-CGEN-TC" USING
IMPL-SEQ-POINTER
TYPE-POINTER.
```

Description

This function creates a TypeCode object from a character string (refer to IMPL-SEQ-POINTER, in the example above) indicating the TypeCode created in the IDL file (_h.cb1) created by the IDL compiler.

Return Values

The created TypeCode object is returned.

Notes

The TypeCode object specified for the TypeCode interface must be the TypeCode object created by this function, or a TypeCode obtained from OUT/INPUT after returning from the server.

4.3 Object Interface

This section describes the object interfaces that manage object references.

4.3.1 CORBA-OBJECT-IS-NIL

Name

CORBA-OBJECT-IS-NIL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY BOOLEAN IN CORBA REPLACING CORBA-BOOLEAN BY RESULT.
```

PROCEDURE DIVISION.

```
CALL "CORBA-OBJECT-IS-NIL" USING
OBJ
ENV
RESULT.
```

Description

This function checks whether an object reference indicates a valid object.

Return Values

If the object reference indicates a valid object, CORBA-TRUE is returned, otherwise CORBA-FALSE is returned.

4.3.2 CORBA-OBJECT-DUPLICATE

Name

CORBA-OBJECT-DUPLICATE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY RET-OBJ.
```

PROCEDURE DIVISION.

```
CALL "CORBA-OBJECT-DUPLICATE" USING
OBJ
ENV
RET-OBJ.
```

Description

This function copies and returns an object reference.

Since this function acquires area to store object references, use the CORBA-OBJECT-RELEASE function to release the area as soon as it is no longer needed.

Return Values

For normal termination, the copied object reference is returned, and CORBA-NO- EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If NULL is specified in OBJ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.3.3 CORBA-OBJECT-RELEASE

Name

CORBA-OBJECT-RELEASE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "CORBA-OBJECT-RELEASE" USING
OBJ
ENV.
```

Description

This function releases the object specified by OBJ. It does not delete an object reference created by the CORBA-BOA-CREATE function; the BOA interface CORBA-BOA-DISPOSE function deletes such object references.

Return Values

If NULL is specified in OBJ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.3.4 CORBA-OBJECT-CREATE-REQUEST

Name

CORBA-OBJECT-CREATE-REQUEST

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY CONTEXT IN CORBA REPLACING CORBA-CONTEXT BY CTX.
01 OPERATION USAGE POINTER.
01 COPY NVLIST IN CORBA REPLACING CORBA-NVLIST BY ARG-LIST.
01 COPY NAMEDVALUE IN CORBA REPLACING CORBA-NAMEDVALUE BY RESULT.
01 COPY REQUEST IN CORBA REPLACING  CORBA-REQUEST BY REQUEST.
01 COPY FLAGS IN CORBA REPLACING CORBA-FLAGS BY REQ-FLAGS.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-OBJECT-CREATE-REQUEST" USING
OBJ
CTX
OPERATION
ARG-LIST
RESULT
REQUEST
REQ-FLAGS
ENV
ORB-STATUS.
CORBA-NAMEDVALUE.
07  NAME  USAGE POINTER.
07  ARGUMENT.
09  ANY-TYPE  USAGE POINTER.
09  ANY-VALUE USAGE POINTER.
07  LEN PIC S9(9) COMP-5.
07  ARG-MODES PIC 9(9) COMP-5.
88  CORBA-ARG-IN                   VALUE   0.
88  CORBA-ARG-OUT                  VALUE   1.
88  CORBA-ARG-INOUT                VALUE   2.
88  CORBA-OUT-LIST-MEMORY          VALUE   3.
88  CORBA-IN-COPY-VALUE            VALUE   4.
88  CORBA-INV-NO-RESPONSE          VALUE   5.
88  CORBA-INV-TERM-ON-ERR          VALUE   6.
88  CORBA-RESP-NO-WAIT             VALUE   7.
88  CORBA-DEPENDENT-LIST           VALUE   8.
88  CORBA-CTX-RESTRICT-SCOPE       VALUE   9.
88  CORBA-CTX-DELETE-DESCENDENTS   VALUE   10.
```

Description

This function creates an object reference from a request object. Specify the object reference with an OD client when invoking the Request interface CORBA-REQUEST-INVOKE or CORBA-REQUEST-SEND functions.

Specify CORBA-OBJECT-NIL or the Context object returned by the CORBA-CONTEXT- CREATE-CHILD function in CTX, and the operation name with CORBA-STRING-SET in OPERATION.

Specify a list object returned by the CORBA-ORB-CREATE-LIST or CORBA-ORB- CREATE-OPERATION-LIST functions in ARG-LIST.

Specify TC-NULL if the parameter does not exist.

Specify the NAMEDVALUE object returned by CORBA-NVLIST-ADD-ITEM in RESULT.

If no value is returned, specify TC-NULL.

For the NamedValue object TypeCode (any type), specify the TypeCode from the return value of the method called by CORBA-REQUEST-INVOKE or CORBA-REQUEST-SEND.

The following flag can be specified in REQ-FLAG:

CORBA-OUT-LIST-MEMORY

As a list object (NVLIST) is associated with a request object, ARG-LIST must always be specified. If a request is deleted by CORBA-REQUEST-DELETE, the corresponding list object is also deleted.

If CORBA-OUT-LIST-MEMORY is not specified, the list object is usable until the program is released. Use the CORBA-FREE function to release it.

Return Values

For normal termination, CORBA-OK is returned, and the request object reference is set in REQUEST.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in OBJ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.3.5 CORBA-OBJECT-GET-IMPLEMENTATION

Name

CORBA-OBJECT-GET-IMPLEMENTATION

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY FJ-IMPLEMENTATIONDEF IN CORBA REPLACING FJ- IMPLEMENTATIONDEF BY
IMPL-DEF.
```

PROCEDURE DIVISION.

```
CALL "CORBA-OBJECT-GET-IMPLEMENTATION" USING
OBJ
ENV
IMPL-DEF.
```

Description

This function returns the object reference of the Implementation Repository that manages implementation information using the specified OBJ.

Return Values

The Implementation Repository object reference is returned.

If NULL is specified in OBJ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.3.6 CORBA-OBJECT-GET-INTERFACE

Name

CORBA-OBJECT-GET-INTERFACE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY INTERFACEDEF IN CORBA REPLACING CORBA-INTERFACEDEF BY INTF- DEF.
```

PROCEDURE DIVISION.

```
CALL "CORBA-OBJECT-GET-INTERFACE" USING
OBJ
ENV
INTF-DEF.
```

Description

This function returns the object reference of the Interface Repository that manages interface information using the specified OBJ.

Return Values

The Interface Repository object reference is returned.

If NULL is specified in OBJ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.4 BOA Interface

Windows32/64 Solaris32/64

This section describes BOA interfaces that manage server applications.

4.4.1 CORBA-BOA-CREATE

Name

CORBA-BOA-CREATE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY BOA IN CORBA REPLACING CORBA-BOA BY BOA.
01 COPY REFERENCEDATA IN CORBA REPLACING CORBA-REFERENCEDATA BY REF.
01 COPY INTERFACEDEF IN CORBA REPLACING CORBA-INTERFACEDEF BY INTF.
01 COPY FJ-IMPLEMENTATIONDEF IN CORBA REPLACING FJ- IMPLEMENTATIONDEF BY IMPL.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
```

PROCEDURE DIVISION.

```
CALL "CORBA-BOA-CREATE" USING
BOA
REF
INTF
IMPL
ENV
OBJ.
CORBA-REFERENCEDATA.
03 SEQ-MAXIMUM  PIC S9(9) COMP-5.
03 SEQ-LENGTH   PIC S9(9) COMP-5.
03 CORBA-OCTET  USAGE  POINTER.
```

Description

This function creates an object reference.

Specify the object reference returned by the CORBA-ORB-BOA-INIT function in BOA, and the object identification information (set by the server application when the object is created) in REF.

The identification information is set in the CORBA-OCTET area in structure CORBA- REFERENCEDATA (CORBA-SEQUENCE-OCTET).

This value does not change unless the object is deleted.

Specify the object reference of the Interface Repository that manages interface information about the object to be created in INTF.

Specify the object reference of the Implementation Repository that manages installation information about the object in IMPL.

Code type information is appended to the generated object reference if the code information is set by the *OD_impl_inst* or *OD_set_env* commands.

When creating an object reference in the server application, the data conversion code type must be what would have been set using the *OD_or_adm* command -L option (now unused).

To dispose of the object reference created here, use the CORBA-BOA-DISPOSE function. Alternatively, to release the object reference memory, use the CORBA-OBJECT-RELEASE function.

Return Values

For normal termination, the created object reference is returned, and CORBA-NO- EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If NULL is specified in BOA, INTF or IMPL, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.4.2 CORBA-BOA-DISPOSE

Name

CORBA-BOA-DISPOSE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY BOA IN CORBA REPLACING CORBA-BOA BY BOA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "CORBA-BOA-DISPOSE" USING
BOA
OBJ
ENV.
```

Description

This function discards the object reference specified in OBJ.

Specify the object reference returned by the CORBA-ORB-BOA-INIT function.

Return Values

If NULL is specified in BOA or OBJ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.4.3 CORBA-BOA-GET-ID

Name

CORBA-BOA-GET-ID

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY BOA IN CORBA REPLACING CORBA-BOA BY BOA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY REFERENCEDATA IN CORBA REPLACING CORBA-REFERENCEDATA BY REF.
```

PROCEDURE DIVISION.

```
CALL "CORBA-BOA-GET-ID" USING
BOA
OBJ
ENV
```

```
REF.
CORBA-REFERENCEDATA.
03 SEQ-MAXIMUM  PIC S9(9) COMP-5.
03 SEQ-LENGTH   PIC S9(9) COMP-5.
03 CORBA-OCTET  USAGE  POINTER.
```

Description

This function returns identification information about the object reference specified in OBJ. A value specified by the CORBA-BOA-CREATE function is returned as this information, and does not change unless the object is discarded.

The identification information is set in the CORBA-OCTET area in structure CORBA- REFERENCEDATA (CORBA-SEQUENCE-OCTET).

Specify the object reference returned by the CORBA-ORB-BOA-INIT function in BOA.

Return Values

For normal termination, the identification information (octet type) is returned and CORBA- NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If NULL is specified in BOA or OBJ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.4.4 CORBA-BOA-SET-EXCEPTION

Name

CORBA-BOA-SET-EXCEPTION

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY BOA IN CORBA REPLACING CORBA-BOA BY BOA.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY FLAGS.
01 USER-ID USAGE POINTER.
01 PARAM USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "CORBA-BOA-SET-EXCEPTION" USING
BOA
FLAGS
USER-ID
PARAM
ENV.
```

Description

This function sets exception information. A server application can be forcibly terminated as an error by calling this function, prior to return.

Specify the object reference returned by the CORBA-ORB-BOA-INIT function in BOA, and the ENV passed to the operation function in ENV.

The following values can be specified in FLAGS:

CORBA-SYSTEM-EXCEPTION

Standard exception

CORBA-USER-EXCEPTION

User exception

Specify an ID for exception identification in USER-ID.

If parameters are specified with an exception, set additional information in PARAM. An example of setting exception information is shown below.

DATA DIVISION.
WORKING-STORAGE SECTION.

```
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY USER-ID-LENGTH.
01 EXCEP-ID-V PIC X(20) VALUE "IDL:INTF_A/EXC_A:1.0".
```

LINKAGE SECTION.

```
01 EXCEP.
03 COPY LONG IN CORBA REPLACING CORBA-LONG BY EXCEP-L.
```

PROCEDURE DIVISION.

```
MOVE FUNCTION LENG(EXCEP-ID-V) TO USER-ID-LENGTH.
CALL "CORBA-STRING-SET" USING
USER-ID
USER-ID-LENGTH
EXCEP-ID-V.

CALL "INTF-A-EXC-A-ALLOC" USING PARAM.
SET ADDRESS OF EXCEP TO PARAM.

MOVE 0 TO EXCEP-L OF EXCEP.
```

Return Values

If NULL is specified in BOA, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.4.5 CORBA-BOA-IMPL-IS-READY

Name

CORBA-BOA-IMPL-IS-READY

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY BOA IN CORBA REPLACING CORBA-BOA BY BOA.
01 COPY FJ-IMPLEMENTATIONDEF IN CORBA REPLACING FJ- IMPLEMENTATIONDEF BY IMPL.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "CORBA-BOA-IMPL-IS-READY" USING
BOA
IMPL
ENV.
```

Description

This function notifies ORB that the server specified at IMPL is ready to receive a request.

The server is one of the following types; shared or persistent.

Return Values

If NULL is specified in BOA or IMPL, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.4.6 CORBA-BOA-DEACTIVATE-IMPL

Name

CORBA-BOA-DEACTIVATE-IMPL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY BOA IN CORBA REPLACING CORBA-BOA BY BOA.
01 COPY FJ-IMPLEMENTATIONDEF IN CORBA REPLACING FJ- IMPLEMENTATIONDEF BY IMPL.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "CORBA-BOA-DEACTIVATE-IMPL" USING
BOA
IMPL
ENV.
```

Description

This function notifies ORB that the shared server service specified at IMPL has stopped.

Return Values

If NULL is specified in BOA or IMPL, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.4.7 CORBA-BOA-OBJ-IS-READY

Name

CORBA-BOA-OBJ-IS-READY

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY BOA IN CORBA REPLACING CORBA-BOA BY BOA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY FJ-IMPLEMENTATIONDEF IN CORBA REPLACING FJ- IMPLEMENTATIONDEF BY IMPL.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "CORBA-BOA-OBJ-IS-READY" USING
BOA
OBJ
IMPL
ENV.
```

Description

This function notifies ORB that the object indicated by OBJ in the unshared server specified at IMPL is ready to receive a request.

Return Values

If NULL is specified in BOA or OBJ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.4.8 CORBA-BOA-DEACTIVATE-OBJ

Name

CORBA-BOA-DEACTIVATE-OBJ

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY BOA IN CORBA REPLACING CORBA-BOA BY BOA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "CORBA-BOA-DEACTIVATE-OBJ" USING
BOA
OBJ
ENV.
```

Description

This function notifies ORB that the object specified in OBJ in the shared or unshared server has stopped.

Return Values

If NULL is specified in BOA or OBJ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.5 NVList Interface

This section describes the NVList interfaces used in dynamic invocation interfaces.

4.5.1 CORBA-NVLIST-ADD-ITEM

Name

CORBA-NVLIST-ADD-ITEM

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY NVLIST IN CORBA REPLACING CORBA-NVLIST BY N-LIST.
01 ITEM-NAME USAGE POINTER.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY ITEM-TYPE.
01 N-VALUE USAGE POINTER.
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY VALUE-LEN.
01 COPY FLAGS IN CORBA REPLACING CORBA-FLAGS BY ITEM-FLAGS.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-NVLIST-ADD-ITEM" USING
N-LIST
ITEM-NAME
ITEM-TYPE
N-VALUE
VALUE-LEN
ITEM-FLAGS
ENV
ORB-STATUS.
```

Description

This function adds parameter information about the server application to be called to the list object specified in N-LIST.

N-LIST is acquired using CORBA-ORB-CREATE-LIST.

Specify the parameter name in ITEM-NAME, the parameter attribute in ITEM-TYPE, the parameter value in N-VALUE, and the parameter length in VALUE-LEN.

Specify one of the following values in ITEM-FLAGS:

CORBA-ARG-IN

Input-only parameter

CORBA-ARG-OUT

Output-only parameter

CORBA-ARG-INOUT

Input-output parameter

CORBA-IN-COPY-VALUE

The value is copied and used.

DEPENDENT-LIST

When the parent list is released, sublists are also released.

These parameters are added after previously added parameters.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

Note

If the client specifies CORBA-ARG-OUT in ITEM-FLAGS, set NULL in N-VALUE.

4.5.2 CORBA-NVLIST-FREE

Name

CORBA-NVLIST-FREE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY NVLIST IN CORBA REPLACING CORBA-NVLIST BY N-LIST.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-NVLIST-FREE" USING
N-LIST
ENV
ORB-STATUS.
```

Description

This function releases the list object specified in N-LIST and related memory. N-LIST is acquired using CORBA-ORB-CREATE-LIST.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

4.5.3 CORBA-NVLIST-FREE-MEMORY

Name

CORBA-NVLIST-FREE-MEMORY

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY NVLIST IN CORBA REPLACING CORBA-NVLIST BY N-LIST.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-NVLIST-FREE-MEMORY" USING
N-LIST
ENV
ORB-STATUS.
```

Description

This function releases the memory related to the list object specified in N-LIST.

N-LIST is acquired using CORBA-ORB-CREATE-LIST. This function does not release the list object.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

4.5.4 CORBA-NVLIST-GET-COUNT

Name

CORBA-NVLIST-GET-COUNT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY NVLIST IN CORBA REPLACING CORBA-NVLIST BY N-LIST.
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY N-COUNT.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-NVLIST-GET-COUNT" USING
N-LIST
N-COUNT
ENV
ORB-STATUS.
```

Description

This function returns the total number of parameters set for the list object specified in N- LIST.

N-LIST is acquired using CORBA-ORB-CREATE-LIST.

Return Values

For normal termination, CORBA-OK is returned, and the total number of parameters is set in N-COUNT.

For abnormal termination, CORBA-FAILED is returned.

4.6 Context Interface

This section describes the context interfaces used to control context objects.

4.6.1 CORBA-CONTEXT-CREATE-CHILD

Name

CORBA-CONTEXT-CREATE-CHILD

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY CONTEXT IN CORBA REPLACING CORBA-CONTEXT BY CTX.
01 CTX-NAME USAGE POINTER.
01 CHILD-CTX USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTEXT-CREATE-CHILD" USING
CTX
CTX-NAME
CHILD-CTX
ENV
ORB-STATUS.
```

Description

This function generates a context object associated with the object specified in CTX.

Return Values

For normal termination, CORBA-OK is returned, and a context object reference is set in CHILD-CTX.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in CTX, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.6.2 CORBA-CONTEXT-SET-ONE-VALUE

Name

CORBA-CONTEXT-SET-ONE-VALUE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY CONTEXT IN CORBA REPLACING CORBA-CONTEXT BY CTX.
01 PROP-NAME USAGE POINTER.
01 C-VALUE USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTEXT-SET-ONE-VALUE" USING
CTX
PROP-NAME
C-VALUE
ENV
ORB-STATUS.
```

Description

This function sets the attribute value specified in PROP-NAME in the context object. Specify the context object reference acquired using CORBA-ORB-GET-DEFAULT- CONTEXT or CORBA-CONTEXT-CREATE-CHILD in CTX.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in CTX, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.6.3 CORBA-CONTEXT-SET-VALUES

Name

CORBA-CONTEXT-SET-VALUES

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY CONTEXT IN CORBA REPLACING CORBA-CONTEXT BY CTX.
01 COPY NVLIST IN CORBA REPLACING CORBA-NVLIST BY C-VALUE.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTEXT-SET-VALUES" USING
CTX
C-VALUE
ENV
ORB-STATUS.
```

Description

This function sets multiple attribute values. Specify the context object reference acquired using CORBA-ORB-GET-DEFAULT-CONTEXT or CORBA-CONTEXT-CREATE-CHILD.

The context object only supports character strings. To use this function, specify CORBA- ARG-IN in the ITEM-FLAGS field, and TC-STRING in the ITEM-TYPE field of the NVLIST object set in C-VALUE.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in CTX or C-VALUE, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.6.4 CORBA-CONTEXT-GET-VALUES

Name

CORBA-CONTEXT-GET-VALUES

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY CONTEXT IN CORBA REPLACING CORBA-CONTEXT BY CTX.
01 START-SCOPE USAGE POINTER.
01 COPY FLAGS IN CORBA REPLACING CORBA-FLAGS BY OP-FLAGS.
01 PROP-NAME USAGE POINTER.
01 COPY NVLIST IN CORBA REPLACING CORBA-NVLIST BY C-VALUE.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTEXT-GET-VALUES" USING
CTX
START-SCOPE
OP-FLAGS
PROP-NAME
C-VALUE
ENV
ORB-STATUS.
```

Description

This function searches for and retrieves attribute values from the context object reference (acquired using CORBA-ORB-DEFAULT-CONTEXT or CORBA-CONTEXT-CREATE- CHILD) CTX.

If a wildcard character "*" is specified in PROP-NAME, all matching attribute names and their values are returned.

The following flags can be specified in OP-FLAGS:

CORBA-CTX-RESTRICT-SCOPE

Only the search scope or context object specified in START-SCOPE can be searched.

START-SCOPE indicates the level of the context object when starting the search for the specified attribute.

If CORBA-CTX-RESTRICT-SCOPE is not set in OP-FLAGS, going up the context tree can continue the search if the desired attribute cannot be found in the specified scope. If the scope name is omitted (" "), the search can begin with the specified context object. If the specified scope name cannot be found, an exception is returned.

Return Values

For normal termination, CORBA-OK is returned, and the attribute value is set in C-VALUE.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in CTX, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.6.5 CORBA-CONTEXT-DELETE-VALUES

Name

CORBA-CONTEXT-DELETE-VALUES

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY CONTEXT IN CORBA REPLACING CORBA-CONTEXT BY CTX.
01 PROP-NAME USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTEXT-DELETE-VALUES" USING
CTX
PROP-NAME
ENV
ORB-STATUS.
```

Description

This function deletes the attribute value specified in PROP-NAME from the context object indicated by CTX. If a wildcard character "*" is specified in PROP-NAME, all matching attribute names and their values are deleted.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in CTX, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.6.6 CORBA-CONTEXT-DELETE

Name

CORBA-CONTEXT-DELETE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY CONTEXT IN CORBA REPLACING CORBA-CONTEXT BY CTX.
01 COPY FLAGS IN CORBA REPLACING CORBA-FLAGS BY DEL-FLAGS.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTEXT-DELETE" USING
CTX
DEL-FLAGS
```

```
ENV
ORB-STATUS.
```

Description

This function deletes the context object specified in CTX. The function discards the object reference using the CORBA-BOA-DISPOSE function.

The following flag can be specified in DEL-FLAGS:

CORBA-CTX-DELETE-DESCENDENTS

When child context objects exist, descendent context objects are also deleted. When descendent context objects exist, this function terminates abnormally if CORBA- CTX-DELETE-DESCENDENTS is omitted.

Return Values

For normal termination, CORBA-OK is returned. For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in CTX, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.7 Request Interface

This section describes the request interfaces used in dynamic invocation interfaces.

4.7.1 CORBA-REQUEST-ADD-ARG

Name

CORBA-REQUEST-ADD-ARG

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY REQUEST IN CORBA REPLACING CORBA-REQUEST BY REQ.
01 NAME USAGE POINTER.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY ARG-TYPE
01 N-VALUE USAGE POINTER.
01 COPY LONGS IN CORBA REPLACING CORBA-LONG BY LEN.
01 COPY FLAGS IN CORBA REPLACING CORBA-FLAGS BY ARG-FLAGS.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-REQUEST-ADD-ARG" USING
REQ
Name
ARG-TYPE
N-VALUE
LEN
ARG-FLAGS
```

```
ENV
ORB-STATUS.
```

Description

This function adds parameter information for the server application function to be called to the request object specified in REQ.

Specify the request object reference returned using CORBA-OBJECT-CREATE-REQUEST in REQ.

Always specify the N-VALUE and LEN parameters. The ARG-TYPE, Name, and ARG- FLAGS parameters can be omitted.

Specify the parameter data type (type code) in ARG-TYPE, and the name in Name.

This function associates parameters with the request object. To associate parameters with the request object, specify them when CORBA-OBJECT-CREATE-REQUEST is called.

The CORBA-OBJECT-CREATE-REQUEST method cannot be called at the same time as this method is called.

One of the following flags can be specified in ARG-FLAG:

CORBA-ARG-IN

Input-only parameter

CORBA-ARG-OUT

Output-only parameter

CORBA-ARG-INOUT

Input-output parameter

CORBA-IN-COPY-VALUE

The argument value is copied and used instead. This flag is ignored when CORBA-ARG- INOUT and CORBA-ARG-OUT are specified.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in REQ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

Note

If CORBA-ARG-OUT is specified in ARG-FLAGS by the client, set NULL in N- VALUE.

4.7.2 CORBA-REQUEST-INVOKE

Name

CORBA-REQUEST-INVOKE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY REQUEST IN CORBA REPLACING CORBA-REQUEST BY REQ.
```

```
01 COPY FLAGS IN CORBA REPLACING CORBA-FLAGS BY INVOKE-FLAGS.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-REQUEST-INVOKE" USING
REQ
INVOKE-FLAGS
ENV
ORB-STATUS.
```

Description

This function calls the server application function specified when the request object specified in REQ is generated. Specify the request object reference returned using CORBA-OBJECT-CREATE- REQUEST in REQ.

The result is assigned to the RESULT argument specified in CORBA-OBJECT-CREATE-REQUEST. The calling application must wait until server application processing terminates. Set NULL in INVOKE-FLAGS.

Before using the method in which user exception is defined, register the contents of the IDL file in an Interface Repository.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in REQ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.7.3 CORBA-REQUEST-SEND

Name

CORBA-REQUEST-SEND

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY REQUEST IN CORBA REPLACING CORBA-REQUEST BY REQ.
01 COPY FLAGS IN CORBA REPLACING CORBA-FLAGS BY INVOKE-FLAGS.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-REQUEST-SEND" USING
REQ
INVOKE-FLAGS
ENV
ORB-STATUS.
```

Description

This function calls the server application function specified when the request object specified in REQ is generated.

Specify the request object reference returned using CORBA-OBJECT-CREATE-REQUEST in REQ.

The result is assigned to the RESULT argument specified in CORBA-OBJECT-CREATE-REQUEST. Unlike CORBA-REQUEST-INVOKE, the invocation application receives control without waiting until server application processing terminates.

The following flag can be specified in INVOKE-FLAGS:

CORBA-INV-NO-RESPONSE

The calling application does not wait for a response. Neither of the output parameters (inout and out) are updated.

This option can be specified even when oneway is not defined in the operation.

The operation result is reported using CORBA-REQUEST-GET-RESPONSE.

When oneway is defined in the operation or CORBA-INV-NO-RESPONSE is specified in INVOKE-FLAGS, CORBA-REQUEST-GET-RESPONSE need not be called.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in REQ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.7.4 CORBA-REQUEST-DELETE

Name

CORBA-REQUEST-DELETE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY REQUEST IN CORBA REPLACING CORBA-REQUEST BY REQ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-REQUEST-DELETE" USING
REQ
ENV
ORB-STATUS.
```

Description

This function deletes the request object specified in REQ.

This function also releases all NVList objects related to the request object. Like CORBA- BOA-DISPOSE, this function discards the request object reference.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in REQ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.7.5 CORBA-REQUEST-GET-RESPONSE

Name

CORBA-REQUEST-GET-RESPONSE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY REQUEST IN CORBA REPLACING CORBA-REQUEST BY REQ.
01 COPY FLAGS IN CORBA REPLACING CORBA-FLAGS BY RESPONSE-FLAGS.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-REQUEST-GET-RESPONSE" USING
REQ
RESPONSE-FLAGS
ENV
ORB-STATUS.
```

Description

This function inquires whether the server application function (specified when the request object specified in REQ is generated), has terminated. If this function is invoked after CORBA-REQUEST-SEND, it can be determined if the server application function has terminated. If it has terminated, the output parameter and function return value associated with the request object become valid.

The following flag can be specified in RESPONSE-FLAG:

CORBA-RESP-NO-WAIT

The calling application receives control even while the server application function is being executed.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in REQ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.8 ServerRequest Interface

Windows32/64 Solaris32/64

This section describes the ServerRequest interfaces used in dynamic skeleton interfaces.

4.8.1 CORBA-SERVERREQUEST-OP-NAME

Name

CORBA-SERVERREQUEST-OP-NAME

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY REQUEST IN CORBA REPLACING CORBA-REQUEST BY REQ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 RET-NAME USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SERVERREQUEST-OP-NAME" USING
REQ
ENV
RET-NAME.
```

Description

This function retrieves a function name from the ServerRequest object specified in REQ.

Specify the object reference to be reported as the first parameter of the DSI processing function in REQ.

Return Values

For normal termination, function name RET-NAME is returned, and CORBA-NO-EXCEPT is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If you specify NULL in REQ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.8.2 CORBA-SERVERREQUEST-PARAMS

Name

CORBA-SERVERREQUEST-PARAMS

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY REQUEST IN CORBA REPLACING CORBA-REQUEST BY DSI-REQUEST.
```

```
01 COPY NVLIST IN CORBA REPLACING CORBA-NVLIST BY ARG-LIST.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SERVERREQUEST-PARAMS" USING
DSI-REQUEST
ARG-LIST
ENV.
```

Description

This function retrieves parameter information from the ServerRequest object specified in DSI-REQUEST, and sets the information in the NVList object specified in ARG-LIST. Specify the object reference to be reported as the first parameter of the DSI processing function in DSI-REQUEST. Specify the object reference generated using CORBA-ORB-CREATE in ARG-LIST, and allocate parameter information setting areas for the number of parameters using CORBA-NVLIST-ADD-ITEM.

Return Values

For normal termination, parameter information is set in the NVList object specified in ARG-LIST, and CORBA-NO-EXCEPT is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If you specify NULL in DSI-REQUEST, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.8.3 CORBA-SERVERREQUEST-RESULT

Name

CORBA-SERVERREQUEST-RESULT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY REQUEST IN CORBA REPLACING CORBA-REQUEST BY DSI-REQUEST.
01 COPY ANY IN CORBA REPLACING CORBA-ANY BY ANY-V.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 RET-NAME USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SERVERREQUEST-RESULT" USING
DSI-REQUEST
ANY-V
ENV.
```

Description

This function sets the method return value in the server application for the ServerRequest object specified in REQUEST.

Specify the any-type variable that stores the return value in ANY-VALUE.

Return Values

For normal termination, CORBA-NO-EXCEPT is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If you specify NULL in DSI-REQUEST, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.8.4 CORBA-SERVERREQUEST-CTX

Name

CORBA-SERVERREQUEST-CTX

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY REQUEST IN CORBA REPLACING CORBA-REQUEST BY REQUEST.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 RET-NAME USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SERVERREQUEST-CTX" USING
REQUEST
ENV
RET-NAME.
```

Description

This function retrieves the context object for the ServerRequest object specified in REQUEST.

Return Values

For normal termination, the context object reference is returned, and CORBA-NO-EXCEPT is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If NULL is specified in REQUEST, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.8.5 CORBA-SERVERREQUEST-EXCEPTION

Name

CORBA-SERVERREQUEST-EXCEPTION

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY SERVERREQUEST IN CORBA REPLACING CORBA-SERVERREQUEST BY DSI-REQUEST.
01 COPY EXCEPTION-TYPE IN CORBA REPLACING CORBA-EXCEPTION-TYPE BY FLAGS.
01 USER-ID USAGE POINTER.
01 PARAM USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SERVERREQUEST-EXCEPTION" USING
DSI-REQUEST
FLAGS
USER-ID
PARAM
ENV.
```

Description

This function sets a user exception caused by a server application method and posts it to the client.

The server application operation can end with an error by invoking this function before it returns. For BOA, specify the object reference that has been returned by the CORBA-ORB-BOA-INIT function. For ENV, specify the ENV that has been passed to the operation function.

One of the following values can be specified for FLAGS:

CORBA-SYSTEM-EXCEPTION: Standard exception

CORBA-USER-EXCEPTION: User exception

For USER-ID, specify an identifier to identify the exception.

Set USER-ID as follows:

DATA DIVISION.

```
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY USER-ID-LENGTH.
01 EXCEP-ID-V PIC X(20) VALUE "IDL:INTF_A/EXC_A:1.0".
```

PROCEDURE DIVISION.

```
MOVE FUNCTION LENG(EXCEP-ID-V) TO USER-ID-LENGTH.
CALL "CORBA-STRING-SET" USING
USER-ID
USER-ID-LENGTH
EXCEP-ID-V.
```

When it is specified that the exception has a parameter, set the address of the additional exception information in PARM(INFO).

Return Values

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure and detailed information is set in ID and MINOR.

For details on ID and MINOR, refer to the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

If NULL is specified for SV-REQ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.9 TypeCode Interface

This section describes TypeCode interfaces that control TypeCode.

4.9.1 CORBA-TYPECODE-EQUAL

Name

CORBA-TYPECODE-EQUAL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY T-OBJECT.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY TC.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY BOOLEAN IN CORBA REPLACING CORBA-BOOLEAN BY RESULT.
```

PROCEDURE DIVISION.

```
CALL "CORBA-TYPECODE-EQUAL" USING
T-OBJECT
TC
ENV
RESULT.
```

Description

This function compares the TypeCode object specified by T-OBJECT with that specified by TC.

Return Values

If the above TypeCode objects match, CORBA-TRUE is returned. If the TypeCode objects do not match, CORBA-FALSE is returned.

If NULL is specified in T-OBJECT or TC, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.9.2 CORBA-TYPECODE-KIND

Name

CORBA-TYPECODE-KIND

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY T-OBJECT.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY KIND.
```

PROCEDURE DIVISION.

```
CALL "CORBA-TYPECODE-KIND" USING
T-OBJECT
ENV
KIND.
```

Description

This function returns the attribute information KIND for the TypeCode object specified by T-OBJECT. The attribute information types are as follows:

```
CORBA-TCKIND PIC S9(9)  COMP-5.
88  CORBA-TK-NULL       VALUE 0.
88  CORBA-TK-VOID       VALUE 1.
88  CORBA-TK-SHORT      VALUE 2.
88  CORBA-TK-LONG       VALUE 3.
88  CORBA-TK-USHORT     VALUE 4.
88  CORBA-TK-ULONG      VALUE 5.
88  CORBA-TK-FLOAT      VALUE 6.
88  CORBA-TK-DOUBLE     VALUE 7.
88  CORBA-TK-BOOLEAN    VALUE 8.
88  CORBA-TK-CHAR       VALUE 9.
88  CORBA-TK-OCTET      VALUE 10.
88  CORBA-TK-ANY        VALUE 11.
88  CORBA-TK-TYPECODE   VALUE 12.
88  CORBA-TK-PRINCIPAL  VALUE 13.
88  CORBA-TK-OBJREF     VALUE 14.
88  CORBA-TK-STRUCT     VALUE 15.
88  CORBA-TK-UNION      VALUE 16.
88  CORBA-TK-ENUM       VALUE 17.
88  CORBA-TK-STRING     VALUE 18.
88  CORBA-TK-SEQUENCE   VALUE 19.
88  CORBA-TK-ARRAY      VALUE 20.
88  CORBA-TK-ALIAS      VALUE 21.
88  CORBA-TK-EXCEPT     VALUE 22.
88  CORBA-TK-LONGLONG   VALUE 23.
88  CORBA-TK-WCHAR      VALUE 26.
88  CORBA-TK-WSTRING    VALUE 27.
```

Return Values

For normal termination, attribute information is returned.

For abnormal termination, CORBA-TK-NULL is returned.

If NULL is specified in T-OBJECT, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.9.3 CORBA-TYPECODE-ID

Name

CORBA-TYPECODE-ID

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY T-OBJECT.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 REP-ID USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-TYPECODE-ID" USING
T-OBJECT
ENV
REP-ID.
```

Description

This function returns the repository ID of the TypeCode object specified by T-OBJECT. This function is valid for TypeCode with the following attribute information:

```
CORBA-TCKIND PIC S9(9)  COMP-5.
88  CORBA-TK-OBJREF     VALUE 14.
88  CORBA-TK-STRUCT     VALUE 15.
88  CORBA-TK-UNION      VALUE 16.
88  CORBA-TK-ENUM       VALUE 17.
88  CORBA-TK-ALIAS      VALUE 21.
88  CORBA-TK-EXCEPT     VALUE 22.
```

Since this function acquires area to store report IDs, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the repository ID is returned.

For abnormal termination, CORBA-USER-EXCEPTION is set in MAJOR of the ENV structure, and detailed information is set in ID.

ID has the following meaning and value:

EX-CORBA-TYPECODE-BADKIND

The TypeCode attribute information is invalid.

If NULL is specified in T-OBJECT, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.9.4 CORBA-TYPECODE-NAME

Name

CORBA-TYPECODE-NAME

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY T-OBJECT.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 MEM-NAME USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-TYPECODE-NAME" USING
T-OBJECT
ENV
MEM-NAME.
```

Description

This function returns the name of the TypeCode object specified by T-OBJECT. This function is valid for TypeCode with the following attribute information:

```
CORBA-TCKIND PIC S9(9)  COMP-5.
88  CORBA-TK-OBJREF     VALUE 14.
88  CORBA-TK-STRUCT     VALUE 15.
88  CORBA-TK-UNION      VALUE 16.
88  CORBA-TK-ENUM       VALUE 17.
88  CORBA-TK-ALIAS      VALUE 21.
88  CORBA-TK-EXCEPT     VALUE 22.
```

Since this function acquires area to store TypeCode object names, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the member name is returned.

For abnormal termination, CORBA-USER-EXCEPTION is set in MAJOR of the ENV structure, and detailed information is set in the ID.

The ID has the following meaning and value:

EX-CORBA-TYPECODE-BADKIND

The TypeCode attribute information is invalid.

If NULL is specified in T-OBJECT, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.9.5 CORBA-TYPECODE-MEMBER-COUNT

Name

CORBA-TYPECODE-MEMBER-COUNT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY T-OBJECT.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY M-COUNT.
```

PROCEDURE DIVISION.

```
CALL "CORBA-TYPECODE-MEMBER-COUNT" USING
T-OBJECT
ENV
M-COUNT.
```

Description

This function returns the number of members of the TypeCode object specified by T- OBJECT.

This function is valid for TypeCode with the following attribute information:

```
CORBA-TCKIND PIC S9(9)  COMP-5.
88  CORBA-TK-STRUCT     VALUE 15.
88  CORBA-TK-UNION      VALUE 16.
88  CORBA-TK-ENUM       VALUE 17.
88  CORBA-TK-EXCEPT     VALUE 22.
```

Return Values

For normal termination, the number of members is returned.

For abnormal termination, CORBA-USER-EXCEPTION is set in MAJOR of the ENV structure, and detailed information is set in the ID.

The ID has the following meaning and value:

EX-CORBA-TYPECODE-BADKIND

The TypeCode attribute information is invalid.

If NULL is specified in T-OBJECT, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.9.6 CORBA-TYPECODE-MEMBER-NAME

Name

CORBA-TYPECODE-MEMBER-NAME

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY T-OBJECT.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY T-INDEX.
```

```
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 MEM-NAME USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-TYPECODE-MEMBER-NAME" USING
T-OBJECT
T-INDEX
ENV
MEM-NAME.
```

Description

This function returns the name of the member specified by T-INDEX of the TypeCode object specified by T-OBJECT.

This function is valid for TypeCode with the following attribute information:

```
CORBA-TCKIND PIC S9(9)  COMP-5.
88  CORBA-TK-STRUCT     VALUE 15.
88  CORBA-TK-UNION      VALUE 16.
88  CORBA-TK-ENUM       VALUE 17.
88  CORBA-TK-EXCEPT     VALUE 22.
```

Since this function acquires area to store member names, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the character string of the member name is returned.

For abnormal termination, CORBA-USER-EXCEPTION is set in MAJOR of the ENV structure, and detailed information is set in the ID.

The ID has the following meanings and values:

EX-CORBA-TYPECODE-BADKIND

The TypeCode attribute information is invalid.

EX-CORBA-TYPECODE-BOUNDS

The T-INDEX value is invalid.

If NULL is specified in T-OBJECT or T-INDEX, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.9.7 CORBA-TYPECODE-MEMBER-TYPE

Name

CORBA-TYPECODE-MEMBER-TYPE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY T-OBJECT.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY T-INDEX.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY RET-TYPE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-TYPECODE-MEMBER-TYPE" USING
T-OBJECT
T-INDEX
ENV
RET-TYPE.
```

Description

This function returns the TypeCode of the member specified by T-INDEX of the TypeCode object specified by T-OBJECT.

This function is valid for TypeCode with the following attribute information:

```
CORBA-TCKIND PIC S9(9)  COMP-5.
88  CORBA-TK-STRUCT     VALUE 15.
88  CORBA-TK-UNION      VALUE 16.
88  CORBA-TK-EXCEPT     VALUE 22.
```

Since this function acquires area to store TypeCodes, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the member's TypeCode is returned.

For abnormal termination, CORBA-USER-EXCEPTION is set in MAJOR of the ENV structure, and detailed information is set in the ID.

The ID has the following meanings and values:

EX-CORBA-TYPECODE-BADKIND

The TypeCode attribute information is invalid.

EX-CORBA-TYPECODE-BOUNDS

The T-INDEX value is invalid.

If NULL is specified in T-OBJECT, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.9.8 CORBA-TYPECODE-MEMBER-LABEL

Name

CORBA-TYPECODE-MEMBER-LABEL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY T-OBJECT.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY T-INDEX.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 RET-LABEL USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-TYPECODE-MEMBER-LABEL" USING
T-OBJECT
```

```
T-INDEX
ENV
RET-LABEL.
```

Description

This function returns the label of the member specified by T-INDEX of the TypeCode object specified by T-OBJECT.

This function is valid for TypeCode with the following attribute information:

```
CORBA-TCKIND PIC S9(9)  COMP-5.
88  CORBA-TK-UNION      VALUE 16.
```

Since this function acquires area to store member labels, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the member's label is returned.

For abnormal termination, CORBA-USER-EXCEPTION is set in MAJOR of the ENV structure, and detailed information is set in the ID.

The ID has the following meanings and values:

EX-CORBA-TYPECODE-BADKIND

The TypeCode attribute information is invalid.

EX-CORBA-TYPECODE-BOUNDS

The T-INDEX value is invalid.

If NULL is specified in T-OBJECT, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.9.9 CORBA-TYPECODE-DISCRIMINATOR-TYPE

Name

CORBA-TYPECODE-DISCRIMINATOR-TYPE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY T-OBJECT.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY RET-TYPE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-TYPECODE-DISCRIMINATOR-TYPE" USING
T-OBJECT
ENV
RET-TYPE.
```

Description

This function returns the TypeCode (RET-TYPE in the above example) of the discriminator information definition for the TypeCode object specified by T-OBJECT. This function is valid for TypeCode with the following attribute information:

```
CORBA-TCKIND PIC S9(9)  COMP-5.
88  CORBA-TK-UNION      VALUE 16.
```

Since this function acquires area to store discriminator information definition TypeCodes, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the TypeCode of the discriminator information definition is returned.

For abnormal termination, CORBA-USER-EXCEPTION is set in MAJOR of the ENV structure, and detailed information is set in the ID.

The ID has the following meaning and value:

EX-CORBA-TYPECODE-BADKIND

The attribute information of TypeCode is invalid.

If NULL is specified in T-OBJECT, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.9.10 CORBA-TYPECODE-DEFAULT-INDEX

Name

CORBA-TYPECODE-DEFAULT-INDEX

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY T-OBJECT.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ULONG IN CORBA REPLACING CORBA-LONG BY T-INDEX.
```

PROCEDURE DIVISION.

```
CALL "CORBA-TYPECODE-DEFAULT-INDEX" USING
T-OBJECT
ENV
T-INDEX.
```

Description

This function returns the index (T-INDEX in the above example) of the default member of the TypeCode object specified by T-OBJECT.

This function is valid for TypeCode with the following attribute information:

```
CORBA-TCKIND PIC S9(9)  COMP-5.
88  CORBA-TK-UNION      VALUE 16.
```

Return Values

For normal termination, the index of the default member is returned. If there is no default member, -1 is returned.

For abnormal termination, CORBA-USER-EXCEPTION is set in MAJOR of the ENV structure, and detailed information is set in the ID.

The ID has the following meaning and value:

EX-CORBA-TYPECODE-BADKIND

The attribute information of TypeCode is invalid.

If NULL is specified in T-OBJECT, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.9.11 CORBA-TYPECODE-LENGTH

Name

CORBA-TYPECODE-LENGTH

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY T-OBJECT.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY T-LENGTH.
```

PROCEDURE DIVISION.

```
CALL "CORBA-TYPECODE-LENGTH" USING
T-OBJECT
ENV
T-LENGTH.
```

Description

This function returns the length of the TypeCode object specified by T-OBJECT (character string length for CORBA-TK-STRING, or the number of elements for CORBA-TK- SEQUENCE or CORBA-TK-ARRAY).

This function is valid for TypeCode with the following attribute information:

```
CORBA-TCKIND PIC S9(9)  COMP-5.
88  CORBA-TK-STRING     VALUE 18.
88  CORBA-TK-SEQUENCE   VALUE 19.
88  CORBA-TK-ARRAY      VALUE 20.
88  CORBA-TK-WSTRING    VALUE 27.
```

Return Values

For normal termination, the TypeCode object length (T-LENGTH in the above example) is returned.

For abnormal termination, CORBA-USER-EXCEPTION is set in MAJOR of the ENV structure, and detailed information is set in the ID.

The ID has the following meaning and value:

EX-CORBA-TYPECODE-BADKIND

The attribute information of TypeCode is invalid.

If NULL is specified in T-OBJECT, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.9.12 CORBA-TYPECODE-CONTENT-TYPE

Name

CORBA-TYPECODE-CONTENT-TYPE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY T-OBJECT.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY RET-TYPE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-TYPECODE-CONTENT-TYPE" USING
T-OBJECT
ENV
RET-TYPE.
```

Description

This function returns the TypeCode object of the member of the TypeCode object specified by T-OBJECT (RET-TYPE in the above example).

This function is valid for TypeCode with the following attribute information:

```
CORBA-TCKIND PIC S9(9)  COMP-5.
88  CORBA-TK-SEQUENCE   VALUE 19.
88  CORBA-TK-ARRAY      VALUE 20.
88  CORBA-TK-ALIAS      VALUE 21.
```

Since this function acquires area to store member TypeCode objects, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the TypeCode object is returned.

For abnormal termination, CORBA-USER-EXCEPTION is set in MAJOR of the ENV structure, and detailed information is set in the ID.

The ID has the following meaning and value:

EX-CORBA-TYPECODE-BADKIND

The attribute information of TypeCode is invalid.

If NULL is specified in T-OBJECT, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.10 Naming Service Interface

This section describes the functions provided by the Naming Service.

4.10.1 Naming Context Interface

This section describes the naming context interface.

4.10.1.1 COSNAMING-NAMINGCONTEXT-BIND

Name

COSNAMING-NAMINGCONTEXT-BIND

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING- NAMINGCONTEXT
BY NC.
01 COPY COSNAMING-NAME IN CORBA REPLACING COSNAMING-NAME BY N.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXT-BIND" USING
NC
N
OBJ
ENV.
```

Description

This function creates a binding with the name specified by N and the object reference specified by OBJ, and registers it in the naming context specified by NC. NC is the naming context object reference.

If N is the compound name, the binding is registered in the naming context specified last in the compound name.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detailed information is set in the ID of the ENV structure.

The ID has the following values and meanings:

IDL:CosNaming/NamingContext/NotFound:1.0

The naming context specified by N was not found.

IDL:CosNaming/NamingContext/CannotProceed:1.0

The naming context specified by NC does not exist.

IDL:CosNaming/NamingContext/InvalidName:1.0

The name is specified incorrectly.

IDL:CosNaming/NamingContext/AlreadyBound:1.0

A binding with the specified name already exists.

4.10.1.2 COSNAMING-NAMINGCONTEXT-REBIND

Name

COSNAMING-NAMINGCONTEXT-REBIND

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC.
01 COPY COSNAMING-NAME IN CORBA REPLACING COSNAMING-NAME BY N.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXT-REBIND" USING
NC
N
OBJ
ENV.
```

Description

This function creates a binding with the name specified by N and the object reference specified by OBJ, and registers it in the naming context specified by NC. No error occurs if a binding with the specified name already exists.

If N is the compound name, the binding is registered in the naming context specified last in the compound name.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detailed information is set in the ID of the ENV structure.

The ID has the following values and meanings:

IDL:CosNaming/NamingContext/NotFound:1.0

The naming context specified by N was not found.

IDL:CosNaming/NamingContext/CannotProceed:1.0

The naming context specified by NC does not exist.

IDL:CosNaming/NamingContext/InvalidName:1.0

The name is specified incorrectly.

4.10.1.3 COSNAMING-NAMINGCONTEXT-BIND-CONTEXT

Name

COSNAMING-NAMINGCONTEXT-BIND-CONTEXT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC1.
01 COPY COSNAMING-NAME IN CORBA REPLACING COSNAMING-NAME BY N.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC2.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXT-BIND-CONTEXT" USING
NC1
N
NC2
ENV.
```

Description

This function creates a binding with the name specified by N and the naming context object reference specified by NC2, and registers it in the naming context specified by NC1. NC1 is the naming context object reference.

If N is the compound name, the binding is registered in the naming context specified last in the compound name.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detailed information is set in the ID of the ENV structure.

The ID has the following values and meanings:

IDL:CosNaming/NamingContext/NotFound:1.0

The naming context specified by N was not found.

IDL:CosNaming/NamingContext/CannotProceed:1.0

The naming context specified by NC does not exist.

IDL:CosNaming/NamingContext/InvalidName:1.0

The name is specified incorrectly.

IDL:CosNaming/NamingContext/AlreadyBound:1.0

A binding with the specified name already exists.

4.10.1.4 COSNAMING-NAMINGCONTEXT-REBIND-CONTEXT

Name

COSNAMING-NAMINGCONTEXT-REBIND-CONTEXT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC1.
01 COPY COSNAMING-NAME IN CORBA REPLACING COSNAMING-NAME BY N.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC2.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXT-REBIND-CONTEXT" USING
NC1
N
NC2
ENV.
```

Description

This function creates a binding with the name specified by N and the object reference of the naming context specified by NC2, and registers it in the naming context specified by NC1. No error occurs if a binding with the specified name already exists.

If N is the compound name, the binding is registered in the naming context specified last in the compound name.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detailed information is set in the ID of the ENV structure.

The ID has the following values and meanings:

IDL:CosNaming/NamingContext/NotFound:1.0

The naming context specified by N was not found.

IDL:CosNaming/NamingContext/CannotProceed:1.0

The naming context specified by NC does not exist.

IDL:CosNaming/NamingContext/InvalidName:1.0

The name is specified incorrectly.

4.10.1.5 COSNAMING-NAMINGCONTEXT-RESOLVE

Name

COSNAMING-NAMINGCONTEXT-RESOLVE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC.
01 COPY COSNAMING-NAME IN CORBA REPLACING COSNAMING-NAME BY N.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXT-RESOLVE" USING
NC
N
ENV
OBJ.
```

Description

This function returns an object reference concatenated to the name specified by N in the naming context specified by NC.

Since this function acquires area to store object references, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the object reference is returned, and CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detailed information is set in the ID of the ENV structure.

The ID has the following values and meanings:

IDL:CosNaming/NamingContext/NotFound:1.0

The naming context specified by N was not found.

IDL:CosNaming/NamingContext/CannotProceed:1.0

The naming context specified by NC does not exist.

IDL:CosNaming/NamingContext/InvalidName:1.0

The name is specified incorrectly.

4.10.1.6 COSNAMING-NAMINGCONTEXT-UNBIND

Name

COSNAMING-NAMINGCONTEXT-UNBIND

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC.
```

```
01 COPY COSNAMING-NAME IN CORBA REPLACING COSNAMING-NAME BY N.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXT-UNBIND" USING
NC
N
ENV.
```

Description

This function deletes the binding with the name specified by N from the naming context specified by NC.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detailed information is set in the ID of the ENV structure.

The ID has the following values and meanings:

IDL:CosNaming/NamingContext/NotFound:1.0

The naming context specified by N was not found.

IDL:CosNaming/NamingContext/CannotProceed:1.0

The naming context specified by NC does not exist.

IDL:CosNaming/NamingContext/InvalidName:1.0

The name is specified incorrectly.

4.10.1.7 COSNAMING-NAMINGCONTEXT-NEW-CONTEXT

Name

COSNAMING-NAMINGCONTEXT-NEW-CONTEXT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC-RET.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXT-NEW-CONTEXT" USING
NC
ENV
NC-RET.
```

Description

This function creates a new naming context in the naming server that manages the naming context specified by NC, and returns the object reference of the created naming context.

Since this function acquires area to store object references, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the object reference of the naming context is returned, and CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set.

Notes

- Fujitsu recommends using the COSNAMING-NAMINGCONTEXT-BIND-NEW-CONTEXT to create and register a naming context object.
- When you create a naming context object using this method, create the binding, then register it in the naming context using the COSNAMING-NAMINGCONTEXT-BIND-CONTEXT function.

4.10.1.8 COSNAMING-NAMINGCONTEXT-BIND-NEW-CONTEXT

Name

COSNAMING-NAMINGCONTEXT-BIND-NEW-CONTEXT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC.
01 COPY COSNAMING-NAME IN CORBA REPLACING COSNAMING-NAME BY N.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC-RET.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXT-BIND-NEW-CONTEXT" USING
NC
N
ENV
NC-RET.
```

Description

This function creates a new naming context, then creates a binding with the object reference of the created naming context and the name specified by N. The binding is then registered in the existing naming context.

This function returns the object reference of the new naming context to the caller. If N is the compound name, the binding is registered in the naming context specified last in the compound name.

Since this function acquires area to store object references, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the naming context object reference is returned, and CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detailed information is set in the ID of the ENV structure.

The ID has the following values and meanings:

IDL:CosNaming/NamingContext/NotFound:1.0

The naming context specified by N was not found.

IDL:CosNaming/NamingContext/CannotProceed:1.0

The naming context specified by NC does not exist.

IDL:CosNaming/NamingContext/InvalidName:1.0

The name is specified incorrectly.

IDL:CosNaming/NamingContext/AlreadyBound:1.0

A binding with the specified name already exists.

4.10.1.9 COSNAMING-NAMINGCONTEXT-DESTROY

Name

COSNAMING-NAMINGCONTEXT-DESTROY

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXT-DESTROY" USING
NC
ENV.
```

Description

This function deletes the naming context specified by NC.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detailed information is set in the ID of the ENV structure.

The ID has the following value and meaning:

IDL:CosNaming/NamingContext/NotEmpty:1.0

The naming context specified by NC contains a binding.

4.10.1.10 COSNAMING-NAMINGCONTEXT-LIST

Name

COSNAMING-NAMINGCONTEXT-LIST

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXT IN CORBA REPLACING COSNAMING-NAMINGCONTEXT
BY NC.
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY HOW-MANY.
01 BL USAGE POINTER.
01 BI USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXT-LIST" USING
NC
HOW-MANY
BL
BI
ENV.
```

Description

This function returns binding lists in the naming context up to the number specified by HOW-MANY. If the value specified in HOW-MANY is greater than the maximum number of bindings specified in the bl_how_many parameter in the nsconfig file, the Naming Service returns the maximum number of bindings in the bl_how_many parameter (refer to "nsconfig" in the "CORBA Service Environment Definition" appendix of the Tuning Guide). If 0 is specified in HOW_MANY, the client returns BI for access to the bindings and a sequence BL of length 0.

The lists are set in the COSNAMING-BINDINGLIST structure specified by BL.

COSNAMING-BINDINGLIST is declared in the library as follows:

01 COSNAMING-BINDING.

02 BINDING-NAME.

03 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-MAXIMUM.

03 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-LENGTH.

03 SEQ-BUFFER USAGE IS POINTER.

02 COPY ENUM IN CORBA REPLACING CORBA-ENUM BY BINDING-TYPE.

88 COSNAMING-NOBJECT VALUE 0.

88 COSNAMING-NCONTEXT VALUE 1.

01 COPY ENUM IN CORBA REPLACING CORBA-ENUM BY COSNAMING- BINDINGTYPE.

88 COSNAMING-NOBJECT VALUE 0.

88 COSNAMING-NCONTEXT VALUE 1.

01 COSNAMING-BINDINGLIST.

02 SEQ-MAXIMUM PIC 9(9) COMP-5.

02 SEQ-LENGTH PIC 9(9) COMP-5.

02 SEQ-BUFFER USAGE IS POINTER.

If the number of bindings in the naming context exceeds the number specified by HOW- MANY, an object (indicating current locations in the naming context) is generated, and its object reference is returned to BI. This object is called the binding iterator.

The object reference returned to BI is used when COSNAMING-BINDINGITERATOR- NEXT-ONE or COSNAMING-BINDINGITERATOR-NEXT-N is called.

Specify the Naming Service object reference (indicating the initial naming context of the Naming Service) acquired in NC.

To do this, invoke the CORBA-ORB-RESOLVE-INITIAL-REFERENCE function, or the object reference of the naming context created using COSNAMING-NAMINGCONTEXT-NEW-CONTEXT.

Since this function acquires area to store the binding iterator and the binding list, use COSNAMING-NAMINGCONTEXT-DESTROY and CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure. For abnormal termination, CORBA-USER-EXCEPTION is set.

4.10.2 Binding Iterator Interface

This section describes the binding iterator interface.

4.10.2.1 COSNAMING-BINDINGITERATOR-NEXT-ONE

Name

COSNAMING-BINDINGITERATOR-NEXT-ONE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 B USAGE POINTER.
01 BI USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY RET.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-BINDINGITERATOR-NEXT-ONE" USING
BI
B
ENV
RET.
```

Description

This function retrieves the next bindings from the current location in the naming context indicated by the binding iterator specified by BI, and stores them in B.

Since this function acquires area to store the binding iterator and the binding list, use COSNAMING-NAMINGCONTEXT-DESTROY and CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination with more than one valid binding returned, CORBA-TRUE-VALUE is returned. If non listed bindings exist, CORBA-FALSE-VALUE is returned, and CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.10.2.2 COSNAMING-BINDINGITERATOR-NEXT-N

Name

COSNAMING-BINDINGITERATOR-NEXT-N

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-BINDINGITERATOR IN CORBA REPLACING
COSNAMING- BINDINGITERATOR BY BI.
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY HOW-MANY.
01 BL USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY RET.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-BINDINGITERATOR-NEXT-N" USING
BI
HOW-MANY
BL
ENV
RET.
```

Description

This function retrieves the next bindings up to the number specified by HOW-MANY from the current location in the naming context (indicated by the binding iterator specified by BI), and stores them in BL. If the value specified in HOW-MANY is greater than the maximum number of bindings specified in the bl_how_many parameter in the nsconfig file, the Naming Service returns the maximum number of bindings in the bl_how_many parameter (refer to "nsconfig" in the "CORBA Service Environment Definition" appendix of the Tuning Guide). If 0 is specified in HOW_MANY, a BAD_PARAM exception is issued.

Since this function acquires area to store the binding iterator and the binding list, use COSNAMING-NAMINGCONTEXT-DESTROY and CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination with more than one valid binding returned, CORBA-TRUE-VALUE is returned. If non listed bindings exist, CORBA-FALSE-VALUE is returned, and CORBA- NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.10.2.3 COSNAMING-BINDINGITERATOR-DESTROY

Name

COSNAMING-BINDINGITERATOR-DESTROY

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-BINDINGITERATOR IN CORBA REPLACING
COSNAMING-BINDINGITERATOR BY BI.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-BINDINGITERATOR-DESTROY" USING
BI
ENV.
```

Description

This function discards the binding iterator specified by BI.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set.

4.10.3 Naming Context Extended Interface

This section describes the naming context extended interface.

4.10.3.1 COSNAMING-NAMINGCONTEXTEXT-TO-STRING

Name

COSNAMING- NAMINGCONTEXTEXT-TO-STRING

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXTEXT IN CORBA REPLACING COSNAMING-
NAMINGCONTEXTEXT BY NCE.
01 COPY COSNAMING-NAME IN CORBA REPLACING COSNAMING-NAME BY N.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY COSNAMING-NAMINGCONTEXTEXT-STR IN CORBA REPLACING COSNAMING-
NAMINGCONTEXTEXT-STR BY SN.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXTEXT-TO-STRING" USING
NCE
N
ENV
SN.
```

Description

Converts the structure type binding name specified in N to a character string type binding name. NCE is the naming context object reference.

This function secures areas for holding the return values. When these areas are no longer required, use the CORBA-FREE function to release them.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detailed information is set in ID in the ENV structure. The value and meaning of ID are as follows:

ID:CosNaming/NamingContext/InvalidName:1.0

There is an error in the name specification.

4.10.3.2 COSNAMING-NAMINGCONTEXTEXT-TO-NAME

Name

COSNAMING- NAMINGCONTEXTEXT-TO-NAME

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXTEXT IN CORBA REPLACING COSNAMING-
NAMINGCONTEXTEXT BY NCE.
01 COPY COSNAMING-NAMINGCONTEXTEXT-STR IN CORBA REPLACING COSNAMING-
NAMINGCONTEXTEXT-STR BY SN.
```

```
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 N USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXTEXT-TO-NAME" USING
NCE
SN
ENV
N.
```

Description

Converts the character string type binding name specified in SN to a structure type binding name. NCE is the naming context object reference.

This function secures areas for holding the return values. When these areas are no longer required, use the CORBA-FREE function to release them.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detailed information is set in ID in the ENV structure. The value and meaning of ID are as follows:

ID:CosNaming/NamingContext/InvalidName:1.0

There is an error in the name specification.

4.10.3.3 COSNAMING-NAMINGCONTEXTEXT-TO-URL

Name

COSNAMING- NAMINGCONTEXTEXT-TO-URL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXTEXT IN CORBA REPLACING
COSNAMING-NAMINGCONTEXTEXT BY NCE.
01 COPY COSNAMING-NAMINGCONTEXTEXT-ADD IN CORBA
REPLACING COSNAMING-NAMINGCONTEXTEXT-ADD BY ADDRKEY.
01 COPY COSNAMING-NAMINGCONTEXTEXT-STR IN CORBA
REPLACING COSNAMING-NAMINGCONTEXTEXT-STR BY SN.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY COSNAMING-NAMINGCONTEXTEXT-URL IN CORBA
REPLACING COSNAMING-NAMINGCONTEXTEXT-URL BY URL.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXTEXT-TO-NAME" USING
NCE
ADDRKEY
SN
```

```
ENV
URL.
```

Description

Creates a URL schema from the address specified in ADDRKEY and the character string binding name specified in SN. NCE is the naming context object reference.

This function secures areas for holding the return values. When these areas are no longer required, use the CORBA-FREE function to release them.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detailed information is set in ID in the ENV structure. The value and meaning of ID are as follows:

ID:CosNaming/NamingContextExt/InvalidAddress:1.0

There is an error in the address specification.

ID:CosNaming/NamingContext/InvalidName:1.0

There is an error in the name specification.

4.10.3.4 COSNAMING-NAMINGCONTEXTEXT-RESOLVE-STR

Name

COSNAMING- NAMINGCONTEXTEXT-RESOLVE-STR

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY COSNAMING-NAMINGCONTEXTEXT IN CORBA REPLACING COSNAMING-
NAMINGCONTEXTEXT BY NCE.
01 COPY COSNAMING-NAMINGCONTEXTEXT-STR IN CORBA REPLACING COSNAMING-
NAMINGCONTEXTEXT-STR BY SN.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
```

PROCEDURE DIVISION.

```
CALL "COSNAMING-NAMINGCONTEXTEXT-RESOLVE-STR" USING
NCE
SN
ENV
OBJ.
```

Description

Returns an object reference to which is bound the character string binding name specified in SN in the naming context specified by NCE.

This function secures areas for holding the return values. When these areas are no longer required, use the CORBA-OBJECT-RELEASE function to release them.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detailed information is set in ID in the ENV structure. The value and meaning of ID are as follows:

ID:CosNaming/NamingContext/NotFound:1.0

The name specified in N was not found.

ID:CosNaming/NamingContext/CannotProceed:1.0

The naming context specified in NC was not found.

ID:CosNaming/NamingContext/InvalidName:1.0

There is an error in the specified name.

4.11 Interface Repository Interface

This section describes the COBOL interface provided by the Interface Repository.

4.11.1 Type Definition

Synopsis

This section describes the libraries required to create application programs using the Interface Repository in OD. Refer to REPLACE.cbl for details of the REPLACE function (provided by the Interface Repository) that can be used through inheritance.

```
CORBA-REPOSITORYID USAGE IS POINTER.    * Repository ID
CORBA-IDENTIFIER USAGE IS POINTER.      * Identifier
CORBA-VERSIONSPEC USAGE IS POINTER.     * Version information
CORBA-SCOPEDNAME USAGE IS POINTER.      * Scope name
* Object Type
CORBA-DEFINITIONKIND PIC  9(9) COMP-5.
88 CORBA-DK-NONE                  VALUE 0.
88 CORBA-DK-ALL                   VALUE 1.
88 CORBA-DK-ATTRIBUTE             VALUE 2.
88 CORBA-DK-CONSTANT              VALUE 3.
88 CORBA-DK-EXCEPTION             VALUE 4.
88 CORBA-DK-INTERFACE             VALUE 5.
88 CORBA-DK-MODULE                VALUE 6.
88 CORBA-DK-OPERATION             VALUE 7.
88 CORBA-DK-TYPEDEF               VALUE 8.
88 CORBA-DK-ALIAS                 VALUE 9.
88 CORBA-DK-STRUCT                VALUE 10.
88 CORBA-DK-UNION                 VALUE 11.
88 CORBA-DK-ENUM                  VALUE 12.
88 CORBA-DK-PRIMITIVE             VALUE 13.
88 CORBA-DK-STRING                VALUE 14.
88 CORBA-DK-SEQUENCE              VALUE 15.
88 CORBA-DK-ARRAY                 VALUE 16.
88 CORBA-DK-REPOSITORY            VALUE 17.
88 CORBA-DK-WSTRING               VALUE 18.
88 CORBA-DK-FIXED                 VALUE 19.
*IROBJECT Object Reference
CORBA-IROBJECT USAGE IS POINTER.
*CONTAINED Object Reference
CORBA-CONTAINED USAGE IS POINTER.
*CONTAINER Object Reference
CORBA-CONTAINER USAGE IS POINTER.
*MODULEDEF Object Reference
CORBA-MODULEDEF USAGE IS POINTER.
```

```
*CONSTANTDEF Object Reference
CORBA-CONSTANTDEF USAGE IS POINTER.
*IDLTYPE Object Reference
CORBA-IDLTYPE USAGE IS POINTER.
*EXCEPTIONDEF Object Reference
CORBA-EXCEPTIONDEF USAGE IS POINTER.
*STRUCTDEF Object Reference
CORBA-STRUCTDEF USAGE IS POINTER.
*UNIONDEF Object Reference
CORBA-UNIONDEF USAGE IS POINTER.
*ENUMDEF Object Reference
CORBA-ENUMDEF USAGE IS POINTER.
*ALIASDEF Object Reference
CORBA-ALIASDEF USAGE IS POINTER.
*INTERFACEDEF Object Reference
CORBA-INTERFACEDEF USAGE IS POINTER.
*ATTRIBUTEDEF Object Reference
CORBA-ATTRIBUTEDEF USAGE IS POINTER.
*OPERATIONDEF Object Reference
CORBA-OPERATIONDEF USAGE IS POINTER.
*REPOSITORY Object Reference
CORBA-REPOSITORY USAGE IS POINTER.
*PRIMITIVEDEF Object Reference
CORBA-PRIMITIVEDEF USAGE IS POINTER.
*STRINGDEF Object Reference
CORBA-STRINGDEF USAGE IS POINTER.
*SEQUENCEDEF Object Reference
CORBA-SEQUENCEDEF USAGE IS POINTER.
*ARRAYDEF Object Reference
CORBA-ARRAYDEF USAGE IS POINTER.
*TYPEDEFDEF Object Reference
CORBA-TYPEDEFDEF USAGE IS POINTER.
*INTERFACEREP Object Reference
CORBA-INTERFACEREP USAGE IS POINTER.
*FIXEDDEF Object Reference
CORBA-FIXEDDEF USAGE IS POINTER.
* CONTAINED object information
CORBA-CONTAINED-DESCRIPTION.
02 KIND PIC 9(9) COMP-5.                          *  Object type
88 CORBA-DK-NONE               VALUE 0.
88 CORBA-DK-ALL                VALUE 1.
88 CORBA-DK-ATTRIBUTE          VALUE 2.
88 CORBA-DK-CONSTANT           VALUE 3.
88 CORBA-DK-EXCEPTION          VALUE 4.
88 CORBA-DK-INTERFACE          VALUE 5.
88 CORBA-DK-MODULE             VALUE 6.
88 CORBA-DK-OPERATION          VALUE 7.
88 CORBA-DK-TYPEDEF            VALUE 8.
88 CORBA-DK-ALIAS              VALUE 9.
88 CORBA-DK-STRUCT             VALUE 10.
88 CORBA-DK-UNION              VALUE 11.
88 CORBA-DK-ENUM               VALUE 12.
88 CORBA-DK-PRIMITIVE          VALUE 13.
88 CORBA-DK-STRING             VALUE 14.
88 CORBA-DK-SEQUENCE           VALUE 15.
88 CORBA-DK-ARRAY              VALUE 16.
88 CORBA-DK-REPOSITORY         VALUE 17.
88 CORBA-DK-WSTRING            VALUE 18.
02 IDL-VALUE.                                    * Object-specific information
03 ANY-TYPE USAGE POINTER.
03 ANY-VALUE USAGE POINTER.

 * Sequence of Contained Object Reference
```

```
CORBA-CONTAINEDSEQ.
02 SEQ-MAXIMUM PIC 9(9) COMP-5.
02 SEQ-LENGTH PIC 9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.

 * CONTAINED object information
CORBA-CONTAINER-DESCRIPTION
 *  Object Reference
02 CONTAINED-OBJECT USAGE IS POINTER.
02 KIND PIC 9(9) COMP-5.                          *  Object type
88 CORBA-DK-NONE             VALUE 0.
88 CORBA-DK-ALL              VALUE 1.
88 CORBA-DK-ATTRIBUTE        VALUE 2.
88 CORBA-DK-CONSTANT         VALUE 3.
88 CORBA-DK-EXCEPTION        VALUE 4.
88 CORBA-DK-INTERFACE        VALUE 5.
88 CORBA-DK-MODULE           VALUE 6.
88 CORBA-DK-OPERATION        VALUE 7.
88 CORBA-DK-TYPEDEF          VALUE 8.
88 CORBA-DK-ALIAS            VALUE 9.
88 CORBA-DK-STRUCT           VALUE 10.
88 CORBA-DK-UNION            VALUE 11.
88 CORBA-DK-ENUM             VALUE 12.
88 CORBA-DK-PRIMITIVE        VALUE 13.
88 CORBA-DK-STRING           VALUE 14.
88 CORBA-DK-SEQUENCE         VALUE 15.
88 CORBA-DK-ARRAY            VALUE 16.
88 CORBA-DK-REPOSITORY       VALUE 17.
88 CORBA-DK-WSTRING          VALUE 18.
88 CORBA-DK-FIXED            VALUE 19.
02 IDL-VALUE.                                    * Object-specific information
03 ANY-TYPE USAGE POINTER.
03 ANY-VALUE USAGE POINTER.

 * Sequence of CONTAINER-DESCRIPTION
CORBA-CONTAINER-DESCRIPTIONSEQ.
02 SEQ-MAXIMUM PIC 9(9) COMP-5.
02 SEQ-LENGTH PIC 9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.

 * MODULEDEF information
CORBA-MODULEDESCRIPTION.
02 NAME USAGE IS POINTER.                         * Identifier
02 IDL-ID USAGE IS POINTER.                       * Repository ID
02 DEFINED-IN USAGE IS POINTER.                   * Repository ID of main object
02 VERSION USAGE IS POINTER.                      * Version information

 * CONSTANTDEF information
CORBA-CONSTANTDESCRIPTION.
02 NAME USAGE IS POINTER.                         * Identifier
02 IDL-ID USAGE IS POINTER.                       * Repository ID
02 DEFINED-IN USAGE IS POINTER.                   * Repository ID of main object
02 VERSION USAGE IS POINTER.                      * Version information
02 IDL-TYPE  USAGE IS POINTER.                    * Type code
02 IDL-VALUE.                                     * Constant value
03 ANY-TYPE  USAGE POINTER.
03 ANY-VALUE USAGE POINTER.

* TYPEDEF information
CORBA-TYPEDESCRIPTION.
02 NAME USAGE IS POINTER.
02 IDL-ID USAGE IS POINTER.
02 DEFINED-IN USAGE IS POINTER.
```

```
02 VERSION USAGE IS POINTER.
02 IDL-TYPE USAGE IS POINTER.

* STRUCTMEMBER information
CORBA-STRUCTMEMBER.
02 NAME USAGE IS POINTER.                        * Identifier
02 IDL-TYPE USAGE IS POINTER.                    * Type code
02 TYPE-DEF USAGE IS POINTER.                    * Object Reference of members

* Sequence of structure members
CORBA-STRUCTMEMBERSEQ.
02 SEQ-MAXIMUM PIC 9(9) COMP-5.
02 SEQ-LENGTH PIC 9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.

* UNIONMEMBER information
CORBA-UNIONMEMBER.
02 NAME USAGE IS POINTER.                        * Identifier
02 IDL-LABEL.                                    * Valve value
03 ANY-TYPE   USAGE POINTER.
03 ANY-VALUE  USAGE POINTER.
02 IDL-TYPE      USAGE POINTER.                  * Type code
02 TYPE-DEF      USAGE POINTER.                  * Object Reference of members

* Sequence of union members
CORBA-UNIONMEMBERSEQ.
02 SEQ-MAXIMUM PIC 9(9) COMP-5.
02 SEQ-LENGTH PIC 9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.

* Sequence of Enum members
CORBA-ENUMMEMBERSEQ.
02 SEQ-MAXIMUM PIC 9(9) COMP-5.
02 SEQ-LENGTH PIC 9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.

* PRIMITIVE types
CORBA-PRIMITIVEKIND  PIC 9(9) COMP-5.
88 CORBA-PK-NULL             VALUE 0.
88 CORBA-PK-VOID             VALUE 1.
88 CORBA-PK-SHORT            VALUE 2.
88 CORBA-PK-LONG             VALUE 3.
88 CORBA-PK-USHORT           VALUE 4.
88 CORBA-PK-ULONG            VALUE 5.
88 CORBA-PK-FLOAT            VALUE 6.
88 CORBA-PK-DOUBLE           VALUE 7.
88 CORBA-PK-BOOLEAN          VALUE 8.
88 CORBA-PK-CHAR             VALUE 9.
88 CORBA-PK-OCTET            VALUE 10.
88 CORBA-PK-ANY              VALUE 11.
88 CORBA-PK-TYPECODE         VALUE 12.
88 CORBA-PK-PRINCIPAL        VALUE 13.
88 CORBA-PK-STRING           VALUE 14.
88 CORBA-PK-OBJREF           VALUE 15.
88 CORBA-PK-LONGLONG         VALUE 16.
88 CORBA-PK-WCHAR            VALUE 19.
88 CORBA-PK-WSTRING          VALUE 20.
88 CORBA-PK-ULONGLONG        VALUE 17.
88 CORBA-PK-LONGDOUBLE       VALUE 18.

* EXCEPTIONDEF information
CORBA-EXCEPTIONDESCRIPTION.
02 NAME USAGE IS POINTER.                        * Identifier
```

```
02 IDL-ID USAGE IS POINTER.                    * Repository ID
02 DEFINED-IN USAGE IS POINTER.                * Repository of main object
02 VERSION USAGE IS POINTER.                   * Version information
02 IDL-TYPE USAGE IS POINTER.                  * Type code

* Attribute type of ATTRIBUTE
CORBA-ATTRIBUTEMODE  PIC 9(9) COMP-5.
88 CORBA-ATTR-NORMAL          VALUE 0.
88 CORBA-ATTR-READONLY        VALUE 1.

* ATTRIBUTEDEF information
CORBA-ATTRIBUTEDESCRIPTION.
02 NAME USAGE IS POINTER.                      * Identifier
02 IDL-ID USAGE IS POINTER.                    * Repository ID
02 DEFINED-IN USAGE IS POINTER.                * Repository of main object
02 VERSION USAGE IS POINTER.                   * Version information
02 IDL-TYPE USAGE IS POINTER.                  * Type code
02 IDL-MODE PIC 9(9) COMP-5.                   * Attribute type
88 CORBA-ATTR-NORMAL          VALUE 0.
88 CORBA-ATTR-READONLY        VALUE 1.

* Attribute type of parameter
CORBA-PARAMETERMODE  PIC 9(9) COMP-5.
88 CORBA-PARAM-IN             VALUE 0.
88 CORBA-PARAM-OUT            VALUE 1.
88 CORBA-PARAM-INOUT          VALUE 2.

* Parameter information
CORBA-PARAMETERDESCRIPTION.
02 NAME USAGE IS POINTER.                      * Identifier
02 IDL-TYPE USAGE IS POINTER.                  * Type code
02 TYPE-DEF USAGE IS POINTER.                  * Object Reference
02 IDL-MODE.PIC 9(9) COMP-5.                   * Attribute
88 CORBA-PARAM-IN             VALUE 0.
88 CORBA-PARAM-OUT            VALUE 1.
88 CORBA-PARAM-INOUT          VALUE 2.

* Sequence of parameter information
CORBA-PARDESCRIPTIONSEQ.
02 SEQ-MAXIMUM PIC 9(9) COMP-5.
02 SEQ-LENGTH PIC 9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.

* Context
CORBA-CONTEXTIDENTIFIER USAGE IS POINTER.

* Sequence of context
CORBA-CONTEXTIDSEQ.
02 SEQ-MAXIMUM PIC 9(9) COMP-5.
02 SEQ-LENGTH PIC 9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.

* Sequence of EXCEPTIONDEF object reference
CORBA-EXCEPTIONDEFSEQ.
02 SEQ-MAXIMUM PIC 9(9) COMP-5.
02 SEQ-LENGTH PIC 9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.

* Sequence of EXCEPTIONDEF information
CORBA-EXCDESCRIPTIONSEQ.
02 SEQ-MAXIMUM PIC S9(9) COMP-5.
02 SEQ-LENGTH PIC S9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.
```

```
* OPERATION attribute type
CORBA-OPERATIONMODE  PIC 9(9) COMP-5.
88 CORBA-OP-NORMAL            VALUE 0.
88 CORBA-OP-ONEWAY            VALUE 1.

CORBA-OPERATIONDESCRIPTION.
02 NAME USAGE IS POINTER.                      * Identifier
02 IDL-ID USAGE IS POINTER.                    * Repository ID
02 DEFINED-IN USAGE IS POINTER.                * Repository ID of main object
02 VERSION USAGE IS POINTER.                   * Version information
02 RESULT USAGE IS POINTER.                    * Type code of return value
02 IDL-MODE   PIC 9(9) COMP-5.                 * Attribute
88 CORBA-OP-NORMAL            VALUE 0.
88 CORBA-OP-ONEWAY            VALUE 1.
02 CONTEXTS.                                   * Context
03 SEQ-MAXIMUM PIC (9) COMP-5.
03 SEQ-LENGTH PIC (9) COMP-5.
03 SEQ-BUFFER USAGE IS POINTER.

02 PARAMETERS.                                 * Parameter information
03 SEQ-MAXIMUM PIC 9(9) COMP-5.
03 SEQ-LENGTH PIC 9(9) COMP-5.
03 SEQ-BUFFER USAGE IS POINTER.
02 EXCEPTIONS.                                 * Exception information
03 SEQ-MAXIMUM PIC 9(9) COMP-5.
03 SEQ-LENGTH PIC 9(9) COMP-5.
03 SEQ-BUFFER USAGE IS POINTER.
CORBA-INTERFACEDEFSEQ.                      * Sequence of INTERFACEDEF object
02 SEQ-MAXIMUM PIC 9(9) COMP-5.
02 SEQ-LENGTH PIC 9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.

CORBA-REPOSITORYIDSEQ.                         * Sequence of REPOSITORYID
02 SEQ-MAXIMUM PIC 9(9) COMP-5.
02 SEQ-LENGTH PIC 9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.

CORBA-OPDESCRIPTIONSEQ.   * Sequence of OPERATIONDESCRIPTION information class
02 SEQ-MAXIMUM PIC 9(9) COMP-5.
02 SEQ-LENGTH PIC 9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.

CORBA-ATTRDESCRIPTIONSEQ.         * Sequence of ATTRIBUTEDEF information class
02 SEQ-MAXIMUM PIC 9(9) COMP-5.
02 SEQ-LENGTH PIC 9(9) COMP-5.
02 SEQ-BUFFER USAGE IS POINTER.

CORBA-INTERFACEDEF-FULLINTERFA.                * Interface information class
02 NAME USAGE IS POINTER.                      * Identifier
02 IDL-ID USAGE IS POINTER.                    * Repository ID
02 DEFINED-IN USAGE IS POINTER.                * Repository ID of main object
02 VERSION USAGE IS POINTER.                   * Version information
02 OPERATIONS.                       * Sequence of Operation information class
03 SEQ-MAXIMUM PIC 9(9) COMP-5.
03 SEQ-LENGTH PIC 9(9) COMP-5.
03 SEQ-BUFFER USAGE IS POINTER.
02 ATTRIBUTES.                       * Sequence of ATTRIBUTE information class
03 SEQ-MAXIMUM PIC 9(9) COMP-5.
03 SEQ-LENGTH PIC 9(9) COMP-5.
03 SEQ-BUFFER USAGE IS POINTER.
02 BASE-INTERFACES.             * Sequence of inherited Interface RepositoryID
03 SEQ-MAXIMUM PIC 9(9) COMP-5.
```

```
03 SEQ-LENGTH PIC 9(9) COMP-5.
03 SEQ-BUFFER USAGE IS POINTER.
02 IDL-TYPE USAGE IS POINTER.                  * Type code

CORBA-INTERFACEDESCRIPTION.
02 NAME USAGE IS POINTER.                      * Identifier
02 IDL-ID USAGE IS POINTER.                    * Repository ID
02 DEFINED-IN USAGE IS POINTER.                * Repository ID of main object
02 VERSION USAGE IS POINTER.                   * Version information
02 BASE-INTERFACES.          * Sequence of inheritance Interface RepositoryID
03 SEQ-MAXIMUM PIC 9(9) COMP-5.
03 SEQ-LENGTH PIC 9(9) COMP-5.
03 SEQ-BUFFER USAGE IS POINTER.
```

4.11.2 IRObject Common Interface

This section describes the function inherited by the Interface Repository interface objects used as an interface object function in the inherited section.

4.11.2.1 CORBA-IROBJECT--GET-DEF-KIND

Name

CORBA-IROBJECT--GET-DEF-KIND

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY DEFINITIONKIND IN CORBA REPLACING CORBA-DEFINITIONKIND
BY DEFINITIONKIND.
```

PROCEDURE DIVISION.

```
CALL "CORBA-IROBJECT--GET-DEF-KIND" USING
OBJ
ENV
DEFINITIONKIND.
```

Description

This function returns the interface type of the interface object specified in OBJ.

Return Values

For normal termination, the interface object's interface type is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.3 Contained Common Interface

This section describes the functions inherited by the interface objects that can be contained in another interface object, and used as interface object functions in the inherited section.

4.11.3.1 CORBA-CONTAINED--GET-ID

Name

CORBA-CONTAINED--GET-ID

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY REPOSITORYID IN CORBA REPLACING CORBA-REPOSITORYID BY REPOSITORYID.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTAINED--GET-ID" USING
OBJ
ENV
REPOSITORYID.
```

Description

This function returns the repository ID of the interface object specified in OBJ.

Since this function acquires area to store the repository ID, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the repository ID is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.3.2 CORBA-CONTAINED--GET-NAME

Name

CORBA-CONTAINED--GET-NAME

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY IDENTIFIER IN CORBA REPLACING CORBA-IDENTIFIER BY IDENTIFIER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTAINED--GET-NAME" USING
OBJ
ENV
IDENTIFIER.
```

Description

This function returns the interface object name specified in OBJ.

Since this function acquires area to store the object name, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the CORBA-Identifier name is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.3.3 CORBA-CONTAINED--GET-DEFINED-IN

Name

CORBA-CONTAINED--GET-DEFINED-IN

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY CONTAINED IN CORBA REPLACING CORBA-CONTAINED BY C-CONTAINED.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTAINED--GET-DEFINED-IN " USING
OBJ
ENV
C-CONTAINED.
```

Description

This function returns the object reference of the Container object, including the interface object specified in OBJ.

Since this function acquires area to store the object reference, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the object reference is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.3.4 CORBA-CONTAINED-DESCRIBE

Name

CORBA-CONTAINED-DESCRIBE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 CONTAINED-DESCRIPTION-ADDR USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTAINED-DESCRIBE" USING
OBJ
ENV
CONTAINED-Description-ADDR.
```

Description

This function returns the CORBA-CONTAINED-Description. This is the structure-identified interface object information specified in OBJ. The information returned differs, depending on the interface object.

The specific information about each interface object is stored in the VALUE member of the CORBA-CONTAINED-Description structure. VALUE is a CORBA-ANY type structure, and the following structure addresses are set according to the interface object type for VALUE member.

Refer to "4.11.1 Type Definition" for details of each structure.

CORBA-MODULEDEF object

MODULEDESCRIPTION structure

CORBA-CONSTANTDEF object

ConstantDESCRIPTION structure

CORBA-STRUCTDEF object

TYPEDESCRIPTION structure

CORBA-UNIONDEF object

TYPEDESCRIPTION structure

CORBA-ENUMDEF object

TYPEDESCRIPTION structure

CORBA-ALIASDEF object

TYPEDESCRIPTION structure

CORBA-EXCEPTIONDEF object

EXCEPTIONDESCRIPTION structure

CORBA-ATTRIBUTEDEF object

ATTRIBUTEDESCRIPTION structure

CORBA-OPERATIONDEF object

OPERATIONDESCRIPTION structure

CORBA-INTERFACEDEF object

INTERFACEDESCRIPTION structure

Since this function acquires area to store the CORBA-CONTAINED-Description structure, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the CORBA-Description structure address is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.3.5 Functions Usable when Inherited

(1) CORBA-CONTAINED--GET-DEF-KIND

Refer to "4.11.2 IRObject Common Interface" for details about (1).

4.11.4 Container Common Interface

This section describes the function inherited by the interface objects that can be contained in another interface object, and can be used as interface object functions in the inherited section.

4.11.4.1 CORBA-CONTAINER-LOOKUP

Name

CORBA-CONTAINER-LOOKUP

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 SEARCH-NAME USAGE IS POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY CONTAINED IN CORBA REPLACING CORBA-CONTAINED BY C- CONTAINED.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTAINER-LOOKUP" USING
OBJ
SEARCH-NAME
ENV
C-CONTAINED.
```

Description

This function searches for an object with the specified name, and returns that object's reference. The search is carried out among the interface objects that include the interface object specified by OBJ.

The search name (SCOPEDName) must be specified in SEARCH-NAME. If the object to be returned cannot be found, object reference of NIL (null) is returned and operation ends normally.

Since this function acquires area to store the object reference, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the object reference of the specified object name is returned.

For abnormal termination, a NIL (empty) object reference is returned. CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure , and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.4.2 CORBA-CONTAINER-CONTENTS

Name

CORBA-CONTAINER-CONTENTS

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY DEFINITIONKIND IN CORBA REPLACING CORBA-DEFINITIONKIND BY LIMIT-TYPE.
01 COPY BOOLEAN IN CORBA REPLACING CORBA-BOOLEAN BY EXCLUDE-INHERITED.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 CONTAINEDSEQ-ADDR USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTAINER-CONTENTS" USING
OBJ
LIMIT-TYPE
EXCLUDE-INHERITED
ENV
CONTAINEDSEQ-ADDR.
```

Description

This function returns (in list format) the object reference of the interface object included by inheritance or directly in the interface object specified by OBJ.

Since this function acquires area to store the list of the object reference, use CORBA-FREE to release the area as soon as it is no longer needed.

- The included object of the interface type specified in limit-type is returned.
- When TRUE is specified in exclude-inherited, inherited objects are not returned.
- When CORBA-DK-ALL is specified in limit-type and FALSE is specified in excluded-inherited, a list of all object references (included or inherited) is returned.
- If the object to be returned cannot be found, 0 is set to SEQ-LENGTH of the return list and object reference becomes undefined.

Return Values

For normal termination, the object reference list of the searched object is returned.

For abnormal termination, a NIL (empty) object reference list is returned. CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure , and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.4.3 CORBA-CONTAINER-LOOKUP-NAME

Name

CORBA-CONTAINER-LOOKUP-NAME

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 SEARCH-NAME USAGE IS POINTER.
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY LEVELS-TO-SEARCH.
01 COPY DEFINITIONKIND IN CORBA REPLACING CORBA-DEFINITIONKIND BY LIMIT-TYPE.
01 COPY BOOLEAN IN CORBA REPLACING CORBA-BOOLEAN BY EXCLUDE-INHERITED.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY CONTAINEDSEQ IN CORBA REPLACING CORBA-CONTAINEDSEQ USAGE
IS CONTAINEDSEQ.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTAINER-LOOKUP-NAME" USING
OBJ
SEARCH-NAME
LEVELS-TO-SEARCH
LIMIT-TYPE
EXCLUDE-INHERITED
ENV
CONTAINEDSEQ.
```

Description

This function searches for an object with the specified name. A list of object references (the object and other objects that include or inherit the object) are returned. The search is carried out among objects included either directly or by inheritance into the interface object specified by OBJ.

Since this function acquires area to store the list of the object reference, use CORBA-FREE to release the area as soon as it is no longer needed.

- Specify the search key name (IDENTIFIER) in search-name.
- Specify the hierarchical depth of the search object in levels-to-search.

  If -1 is specified, the entire hierarchy is searched.

  If 1 is specified, only the objects immediately under the specified object are searched.
- The interface type (including the object specified in limit-type) is returned.
- If TRUE is specified in exclude-inherited, inherited objects are not returned.
- If CORBA-DK-ALL is specified in limit-type, and FALSE is specified in excluded-inherited, the specified object and a list of all object references (included or inherited) is returned.
- If the object to be returned cannot be found, 0 is set to SEQ-LENGTH of the return list and object reference becomes undefined.

Return Values

For normal termination, the object reference list of the searched objects is returned.

For abnormal termination, a NIL (empty) object reference list is returned. CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure , and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.4.4 CORBA-CONTAINER-DESCRIBE-CONTENTS

Name

CORBA-CONTAINER-DESCRIBE-CONTENTS

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY DEFINITIONKIND IN CORBA REPLACING CORBA-DEFINITIONKIND BY LIMIT-TYPE.
01 COPY BOOLEAN IN CORBA REPLACING CORBA-BOOLEAN BY EXCLUDE-INHERITED.
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY MAX-RETURNED-OBJS.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 CONTAINER-DescriptionSEQ-ADDR USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONTAINER-DESCRIBE-CONTENTS" USING
OBJ
LIMIT-TYPE
EXCLUDE-INHERITED
MAX-RETURNED-OBJS
ENV
CONTAINER-DescriptionSEQ-ADDR.
```

Description

This function returns (in Description structure list format) the object definition information included directly or by inheritance in the interface object specified in OBJ.

Refer to "4.11.1 Type Definition" for more details.

Since this function acquires area to store the list of Description structure, use CORBA-FREE to release the area as soon as it is no longer needed.

The interface type (including the object specified in limit-type) is returned.

If TRUE is specified in exclude-inherited, inherited objects are not returned.

The number of definition information items specified by max-returned-objs is returned.

If CORBA-DK-ALL is specified in limit-type, and FALSE is specified in excluded-inherited, a list of all object definition information items (included or inherited) is returned. Note that only the number of items specified in max-returned-objs is returned.

To acquire the registered number of items, specify -1 in max-returned-objs.

If the object to be returned cannot be found, 0 is set to SEQ-LENGTH of the return list and object reference becomes undefined.

Return Values

For normal termination, the address of the object definition information list is returned.

For abnormal termination, a NIL (empty) definition information list is returned. CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure , and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.4.5 Functions Usable when Inherited

(1) CORBA-CONTAINER--GET-DEF-KIND

Refer to "4.11.2 IRObject Common Interface" for details about (1).

4.11.5 IDLType Common Interface

This section details the functions associated with the IDLType Common Interface.

4.11.5.1 CORBA-IDLTYPE--GET-TYPE

Name

CORBA-IDLTTYPE--GET-TYPE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY TYPECODE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-IDLTYPE--GET-TYPE" USING
OBJ
ENV
TYPECODE.
```

Description

This function returns the IDLType object type code.

Since this function acquires area to store the type code, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the IDLType object type code is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.5.2 Functions Usable when Inherited

(1) CORBA-IDLTYPE--GET-DEF-KIND

Refer to "4.11.2 IRObject Common Interface" for details about (1).

4.11.6 Repository Interface

This section describes the Interface Repository interface.

4.11.6.1 CORBA-REPOSITORY-LOOKUP-ID

Name

CORBA-REPOSITORY-LOOKUP-ID

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY REPOSITORYID IN CORBA REPLACING CORBA-REPOSITORYID BY SEARCH-ID.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY CONTAINED IN CORBA REPLACING CORBA-CONTAINED BY C-CONTAINED.
```

PROCEDURE DIVISION.

```
CALL "CORBA-REPOSITORY-LOOKUP-ID" USING
OBJ
SEARCH-ID
ENV
C-CONTAINED.
```

Description

This function retrieves the object with the repository ID specified in SEARCH-ID, and returns its object reference. The Interface Repository object reference must be specified in OBJ. If the object to be returned cannot be found, object reference of NIL (null) is returned and operation ends normally.

Since this function acquires area to store the object reference, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the object reference of the searched object is returned.

For abnormal termination, a NIL (empty) object reference is returned.

If NULL is specified in OBJ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.6.2 Functions Usable when Inherited

(1) CORBA-REPOSITORY--GET-DEF-KIND

(2) CORBA-REPOSITORY-LOOKUP

(3) CORBA-REPOSITORY-CONTENTS

(4) CORBA-REPOSITORY-LOOKUP-NAME

(5) CORBA-REPOSITORY-DESCRIBE-CONTENTS

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.4 Container Common Interface" for details about (2) to (5).

4.11.7 ModuleDef Interface

This section describes the ModuleDef interface.

4.11.7.1 Functions Usable when Inherited

(1) CORBA-MODULEDEF--GET-DEF-KIND

(2) CORBA-MODULEDEF--GET-ID

(3) CORBA-MODULEDEF--GET-NAME

(4) CORBA-MODULEDEF--GET-DEFINED-IN

(5) CORBA-MODULEDEF-DESCRIBE

(6) CORBA-MODULEDEF-LOOKUP

(7) CORBA-MODULEDEF-CONTENTS

(8) CORBA-MODULEDEF-LOOKUP-NAME

(9) CORBA-MODULEDEF-DESCRIBE-CONTENTS

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.3 Contained Common Interface" for details about (2) to (5).

Refer to "4.11.4 Container Common Interface" for details about (6) to (9).

4.11.8 ConstantDef Interface

This section describes the ConstantDef interface.

4.11.8.1 CORBA-CONSTANTDEF--GET-TYPE

Name

CORBA-CONSTANTDEF--GET-TYPE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY IS TYPECODE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONSTANTDEF--GET-TYPE" USING
OBJ
ENV
TYPECODE.
```

Description

This function returns the TypeCode of the constant definition object specified in OBJ.

Since this function acquires area to store the type code, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the TypeCode is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.8.2 CORBA-CONSTANTDEF--GET-VALUE

Name

CORBA-CONSTANTDEF--GET-VALUE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 ANY-ADDR USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-CONSTANTDEF--GET-VALUE" USING
OBJ
ENV
ANY-ADDR.
```

Description

This function returns the constant value of the ConstantDef object specified in OBJ, in CORBA-ANY format.

Since this function acquires area to store the constant value (CORBA-ANY), use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the constant value (CORBA-ANY) address is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.8.3 Functions Usable when Inherited

(1) CORBA-CONSTANTDEF--GET-DEF-KIND

(2) CORBA-CONSTANTDEF--GET-ID

(3) CORBA-CONSTANTDEF--GET-NAME

(4) CORBA-CONSTANTDEF--GET-DEFINED-IN

(5) CORBA-CONSTANTDEF-DESCRIBE

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.3 Contained Common Interface" for details about (2) to (5).

4.11.9 StructDef Interface

This section describes the StructDef interface.

4.11.9.1 CORBA-STRUCTDEF--GET-MEMBERS

Name

CORBA-STRUCTDEF--GET-MEMBERS

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 STRUCTMEMBERSEQ-ADDR USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-STRUCTDEF--GET-MEMBERS" USING
OBJ
ENV
STRUCTMEMBERSEQ-ADDR.
```

Description

This function returns the StructDef object member information specified in OBJ, in STRUCTMEMBER structure list format.

Refer to "4.11.1 Type Definition" for more information.

Since this function acquires area to store the list of STRUCTMEMBER structure, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the address of StructDef object member information (STRUCTMEMBER structure) is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.9.2 Functions Usable when Inherited

(1) CORBA-STRUCTDEF--GET-DEF-KIND

(2) CORBA-STRUCTDEF--GET-ID

(3) CORBA-STRUCTDEF--GET-NAME

(4) CORBA-STRUCTDEF--GET-DEFINED-IN

(5) CORBA-STRUCTDEF-DESCRIBE

(6) CORBA-STRUCTDEF--GET-TYPE

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.3 Contained Common Interface" for details about (2) to (5).

Refer to "4.11.5 IDLType Common Interface" for details about (6).

4.11.10 UnionDef Interface

This section describes the UnionDef interface.

4.11.10.1 CORBA-UNIONDEF--GET-DISCRIMINATOR-TYPE

Name

CORBA-UNIONDEF--GET-DISCRIMINATOR-TYPE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY TYPECODE TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY TYPECODE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-UNIONDEF--GET-DISCRIMINATOR-TYPE" USING
OBJ
ENV
TYPECODE.
```

Description

This function returns the TypeCode of the UnionDef object value information definition specified in OBJ.

Since this function acquires area to store the type code, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the TypeCode of the value information definition is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.10.2 CORBA-UNIONDEF--GET-MEMBERS

Name

CORBA-UNIONDEF--GET-MEMBERS

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 UNIONMEMBERSEQ-ADDR USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-UNIONDEF--GET-MEMBERS" USING
OBJ
ENV
UNIONMEMBERSEQ.
```

Description

This function returns the UnionDef object member information specified in OBJ, in UNIONMEMBER structure list format. Refer to "4.11.1 Type Definition" for more information.

Since this function acquires area to store the list of UNIONMEMBER structure, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the address of UnionDef object member information (UNIONMEMBER structure) is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.10.3 Functions Usable when Inherited

(1) CORBA-UNIONDEF--GET-DEF-KIND

(2) CORBA-UNIONDEF--GET-ID

(3) CORBA-UNIONDEF--GET-NAME

(4) CORBA-UNIONDEF--GET-DEFINED-IN

(5) CORBA-UNIONDEF-DESCRIBE

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.3 Contained Common Interface" for details about (2) to (5).

4.11.11 EnumDef Interface

This section describes the EnumDef interface.

4.11.11.1 CORBA-ENUMDEF--GET-MEMBERS

Name

CORBA-ENUMDEF--GET-MEMBERS

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 ENUMMEMBERSEQ-ADDR USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ENUMDEF--GET-MEMBERS" USING
OBJ
ENV
ENUMMEMBERSEQ-ADDR.
```

Description

This function returns the EnumDef object member information specified in OBJ, in list format. Refer to "4.11.1 Type Definition" for more information.

Since this function acquires area to store the list of the EnumDef object, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the address of the EnumDef object member information is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.11.2 Functions Usable when Inherited

(1) CORBA-ENUMDEF--GET-DEF-KIND

(2) CORBA-ENUMDEF--GET-ID

(3) CORBA-ENUMDEF--GET-NAME

(4) CORBA-ENUMDEF--GET-DEFINED-IN

(5) CORBA-ENUMDEF-DESCRIBE

Refer to "4.11.2 IRObject Common Interface" for details about (1). Refer to "4.11.3 Contained Common Interface" for details about (2) to (5).

4.11.12 AliasDef Interface

This section describes the AliasDef interface.

4.11.12.1 CORBA-ALIASDEF--GET-ORIGINAL-TYPE-DEF

Name

CORBA-ALIASDEF--GET-ORIGINAL-TYPE-DEF

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
```

```
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY IDLTYPE IN CORBA REPLACING CORBA-IDLTYPE BY IDLTYPE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ALIASDEF--GET-ORIGINAL-TYPE-DEF" USING
OBJ
ENV
IDLTYPE.
```

Description

This function returns the object reference of the original ALIASDEF object data type.

Since this function acquires area to store the object reference, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the object reference of the original ALIASDEF object data type is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.12.2 Functions Usable when Inherited

(1) CORBA-ALIASDEF--GET-DEF-KIND

(2) CORBA-ALIASDEF--GET-ID

(3) CORBA-ALIASDEF--GET-NAME

(4) CORBA-ALIASDEF--GET-DEFINED-IN

(5) CORBA-ALIASDEF-DESCRIBE

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.3 Contained Common Interface" for details about (2) to (5).

4.11.13 StringDef Interface

This section describes the StringDef interface.

4.11.13.1 CORBA-STRINGDEF--GET-BOUND

Name

CORBA-STRINGDEF--GET-BOUND

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY BOUND.
```

PROCEDURE DIVISION.

```
CALL "CORBA-STRINGDEF--GET-BOUND" USING
OBJ
```

```
ENV
BOUND.
```

Description

This function returns the maximum number of characters of the StringDef object specified in OBJ.

Return Values

For normal termination, the maximum number of characters is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.13.2 Functions Usable when Inherited

(1) CORBA-STRINGDEF--GET-DEF-KIND

(2) CORBA-STRINGDEF--GET-TYPE

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.5 IDLType Common Interface" for details about (2).

4.11.14 SequenceDef Interface

This section describes the SequenceDef interface.

4.11.14.1 CORBA-SEQUENCEDEF--GET-BOUND

Name

CORBA-SEQUENCEDEF--GET-BOUND

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY BOUND.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SEQUENCEDEF--GET-BOUND" USING
OBJ
ENV
BOUND.
```

Description

This function returns the maximum value of the sequence element defined in the SequenceDef object specified in OBJ.

Return Values

For normal termination, the maximum value of the sequence element is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.14.2 CORBA-SEQUENCEDEF--GET-ELEMENT-TYPE

Name

CORBA-SEQUENCEDEF--GET-ELEMENT-TYPE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY TYPECODE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SEQUENCEDEF--GET-ELEMENT-TYPE" USING
OBJ
ENV
TYPECODE.
```

Description

This function returns the TypeCode identifying the sequence element type defined in the SequenceDef object specified in OBJ.

Since this function acquires area to store the type code, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the TypeCode identifying the sequence element type is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Mijor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.14.3 Functions Usable when Inherited

(1) CORBA-SEQUENCEDEF--GET-DEF-KIND

(2) CORBA-SEQUENCEDEF--GET-TYPE

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.5 IDLType Common Interface" for details about (2).

4.11.15 ArrayDef Interface

This section describes ArrayDef interface.

4.11.15.1 CORBA-ARRAYDEF--GET-LENGTH

Name

CORBA-ARRAYDEF--GET-LENGTH

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY G-LENGTH.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ARRAYDEF--GET-LENGTH" USING
OBJ
ENV
G-LENGTH.
```

Description

This function returns the number of elements in the ArrayDef object array specified in OBJ.

Return Values

For normal termination, the number of array elements is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Mijor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.15.2 CORBA-ARRAYDEF--GET-ELEMENT-TYPE

Name

CORBA-ARRAYDEF--GET-ELEMENT-TYPE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY TYPECODE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ARRAYDEF--GET-ELEMENT-TYPE" USING
OBJ
ENV
TYPECODE.
```

Description

This function returns the TypeCode identifying the element type of the ArrayDef object array specified in OBJ.

Since this function acquires area to store the type code, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the TypeCode identifying the array element type is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Mijor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.15.3 Functions Usable when Inherited

(1) CORBA-ARRAYDEF--GET-DEF-KIND

(2) CORBA-ARRAYDEF--GET-TYPE

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.5 IDLType Common Interface" for details about (2).

4.11.16 WstringDef Interface

This section describes the WstringDef interface.

4.11.16.1 CORBA-WSTRINGDEF-GET-BOUND

Name

CORBA-WSTRINGDEF-GET-BOUND

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY BOUND.
```

PROCEDURE DIVISION.

```
CALL "CORBA-WSTRINGDEF-GET-BOUND" USING
OBJ
ENV
BOUND.
```

Description

This function returns the maximum number of characters of the WstringDef object specified in OBJ.

Return Values

For normal termination, the number of characters is returned. In the event of abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR of the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Mijor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.16.2 Functions Usable when Inherited

(1) CORBA-WSTRINGDEF--GET-DEF-KIND

(2) CORBA-WSTRINGDEF--GET-TYPE

For details of (1), refer to "4.11.2 IRObject Common Interface".

For details of (2), refer to "4.11.5 IDLType Common Interface".

4.11.17 FixedDef Interface

This section describes the FixedDef interface.

4.11.17.1 CORBA-FIXEDDEF-GET-DIGITS

Name

CORBA-FIXEDDEF-GET-DIGITS

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY USHORT IN CORBA REPLACING CORBA-UNSIGNED-SHORT BY DIGITS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-FIXEDDEF--GET-DIGITS" USING
OBJ
ENV
DIGITS.
```

Description

This function returns the digits of the FixedDef.

Return Values

For normal termination, the digits is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual,.

4.11.17.2 CORBA-FIXEDDEF -- GET-SCALE

Name

CORBA-FIXEDDEF -- GET-SCALE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY SHORT IN CORBA REPLACING CORBA-SHORT BY SCALE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-FIXEDDEF--GET-SCALE" USING
OBJ
ENV
SCALE.
```

Description

This function returns the scale of the FixedDef.

Return Values

For normal termination, the scale is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.17.3 Functions Usable when Inherited

(1) CORBA-FIXEDDEF--GET-DEF-KIND

(2) CORBA-FIXEDDEF--GET-TYPE

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.5 IDLType Common Interface" for details about (2).

4.11.18 InterfaceDef Interface

This section describes the InterfaceDef interface.

4.11.18.1 CORBA-INTERFACEDEF-DESCRIBE-INTERFACE

Name

CORBA-INTERFACEDEF-DESCRIBE-INTERFACE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 FULLINTERFA-ADDR USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-INTERFACEDEF-DESCRIBE-INTERFACE" USING
OBJ
ENV
FULLINTERFA-ADDR.
```

Description

This function returns (in CORBA-INTERFACEDEF-FULLINTERFA structure format) the InterfaceDef object definition information (including inheritance information) specified in OBJ. Refer to "4.11.1 Type Definition" for more information.

Since this function acquires area to store the CORBA-INTERFACEDEF-FULLINTERFA structure, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, a pointer to InterfaceDef object definition information (INTERFACE-FULLINTERFA structure) is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.18.2 Functions Usable when Inherited

(1) CORBA-INTERFACEDEF--GET-DEF-KIND

(2) CORBA-INTERFACEDEF--GET-ID

(3) CORBA-INTERFACEDEF--GET-NAME

(4) CORBA-INTERFACEDEF--GET-DEFINED-IN

(5) CORBA-INTERFACEDEF-DESCRIBE

(6) CORBA-INTERFACEDEF-LOOKUP

(7) CORBA-INTERFACEDEF-CONTENTS

(8) CORBA-INTERFACEDEF-LOOKUP-NAME

(9) CORBA-INTERFACEDEF-DESCRIBE-CONTENTS

(10) CORBA-INTERFACEDEF-GET-TYPE

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.3 Contained Common Interface" for details about (2) to (5).

Refer to "4.11.4 Container Common Interface" for details about (6) to (9).

Refer to "4.11.5 IDLType Common Interface" for details about (10).

4.11.19 OperationDef Interface

This section describes the OperationDef interface.

4.11.19.1 CORBA-OPERATIONDEF--GET-RESULT

Name

CORBA-OPERATIONDEF--GET-RESULT

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY TYPECODE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-OPERATIONDEF--GET-RESULT" USING
OBJ
ENV
TYPECODE.
```

Description

This function returns the TypeCode identifying the operation return value defined in the OperationDef object specified in OBJ.

Since this function acquires area to store the type code, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the TypeCode identifying the operation return value is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.19.2 CORBA-OPERATIONDEF--GET-PARAMS

Name

CORBA-OPERATIONDEF--GET-PARAMS

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 PARDescriptionSEQ-ADDR USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-OPERATIONDEF--GET-PARAMS" USING
OBJ
ENV
PARDESCRIPTIONSEQ-ADDR.
```

Description

This function returns (in CORBA-PARDescriptionSEQ structure list format) the operation parameter information defined in the OperationDef object specified in OBJ. Refer to "4.11.1 Type Definition" for more information.

Since this function acquires area to store the list of CORBA-PARDescriptionSEQ structure, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, a pointer to the operation parameter information (CORBA-PARDescriptionSEQ) is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.19.3 CORBA-OPERATIONDEF--GET-CONTEXTS

Name

CORBA-OPERATIONDEF--GET-CONTEXTS

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 CONTEXTIDSEQ-ADDR USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-OPERATIONDEF--GET-CONTEXTS" USING
OBJ
ENV
CONTEXTIDSEQ-ADDR.
```

Description

This function returns (in list format) the context identifier for the operation defined in the OperationDef object specified in OBJ. Refer to "4.11.1 Type Definition" for more information.

Since this function acquires area to store the list of the context identifier, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the context identifier list is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.19.4 CORBA-OPERATIONDEF--GET-EXCEPTIONS

Name

CORBA-OPERATIONDEF--GET-EXCEPTIONS

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 EXCEPTIONDEFSEQ-ADDR USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-OPERATIONDEF--GET-EXCEPTIONS" USING
OBJ
ENV
EXCEPTIONDEFSEQ-ADDR.
```

Description

This function returns (in list format) the definition information of the operation exceptions defined in the OperationDef object specified in OBJ. Refer to "4.11.1 Type Definition" for more information.

Since this function acquires area to store the list of the definition information of the exceptions, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the operation exceptions definition information (EXCEPTIONDEFSEQ) is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.19.5 Functions Usable when Inherited

(1) CORBA-OPERATIONDEF--GET-DEF-KIND

(2) CORBA-OPERATIONDEF--GET-ID

(3) CORBA-OPERATIONDEF--GET-NAME

(4) CORBA-OPERATIONDEF--GET-DEFINED-IN

(5) CORBA-OPERATIONDEF-DESCRIBE

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.3 Contained Common Interface" for details about (2) to (5).

4.11.20 AttributeDef Interface

This section describes the AttributeDef interface.

4.11.20.1 CORBA-ATTRIBUTEDEF--GET-TYPE

Name

CORBA-ATTRIBUTEDEF--GET-TYPE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY TYPECODE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ATTRIBUTEDEF--GET-TYPE" USING
OBJ
ENV
TYPECODE.
```

Description

This function returns the TypeCode of the AttributeDef object specified in OBJ.

Since this function acquires area to store the type code, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the AttributeDef object TypeCode is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual,

4.11.20.2 Functions Usable when Inherited

(1) CORBA-ATTRIBUTEDEF--GET-DEF-KIND

(2) CORBA-ATTRIBUTEDEF--GET-ID

(3) CORBA-ATTRIBUTEDEF--GET-NAME

(4) CORBA-ATTRIBUTEDEF--GET-DEFINED-IN

(5) CORBA-ATTRIBUTEDEF-DESCRIBE

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.3 Contained Common Interface" for details about (2) to (5).

4.11.21 ExceptionDef Interface

This section describes the ExceptionDef interface.

4.11.21.1 CORBA-EXCEPTIONDEF--GET-TYPE

Name

CORBA-EXCEPTIONDEF--GET-TYPE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY TYPECODE IN CORBA REPLACING CORBA-TYPECODE BY TYPECODE.
```

PROCEDURE DIVISION.

```
CALL "CORBA-EXCEPTIONDEF--GET-TYPE" USING
OBJ
ENV
TYPECODE.
```

Description

This function returns the TypeCode of the ExceptionDef object specified in OBJ.

Since this function acquires area to store the type code, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, the TypeCode of ExceptionDef object is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.21.2 CORBA-EXCEPTIONDEF--GET-MEMBERS

Name

CORBA-EXCEPTIONDEF--GET-MEMBERS

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 STRUCTMEMBERSEQ-ADDR USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-EXCEPTIONDEF--GET-MEMBERS" USING
OBJ
ENV
STRUCTMEMBERSEQ.
```

Description

This function returns definition information of the exception event (CORBA- STRUCTMEMBERSEQ) defined by the ExceptionDef object specified in obj, in list format.

Since this function acquires area to store the list of the definition information of the exception event, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, definition information of the exception event (CORBA- STRUCTMEMBERSEQ) is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is returned in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.11.21.3 Functions Usable when Inherited

(1) CORBA-EXCEPTIONDEF--GET-DEF-KIND

(2) CORBA-EXCEPTIONDEF--GET-ID

(3) CORBA-EXCEPTIONDEF--GET-NAME

(4) CORBA-EXCEPTIONDEF--GET-DEFINED-IN

(5) CORBA-EXCEPTIONDEF-DESCRIBE

Refer to "4.11.2 IRObject Common Interface" for details about (1).

Refer to "4.11.3 Contained Common Interface" for details about (2) to (5).

4.12 Other Functions

This section describes the CORBA defined functions not included in the previous interfaces, and the Fujitsu Extended Interface.

4.12.1 CORBA-SEND-MULTIPLE-REQUESTS

Name

CORBA-SEND-MULTIPLE-REQUESTS

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 REQ USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY C-COUNT.
01 COPY FLAGS IN CORBA REPLACING CORBA-FLAGS BY INVOKE-FLAGS.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SEND-MULTIPLE-REQUESTS" USING
REQ
ENV
C-COUNT
INVOKE-FLAGS
ORB-STATUS.
```

Description

This function allows one or more requests to be sent to the server in a single function call. Multiple requests are sent in parallel. Specify the request objects in REQ. The number of request objects is specified using C-COUNT. Like CORBA-REQUEST-SEND, the system returns to the calling application without waiting for the server application function to terminate.

The following value can be specified in INVOKE-FLAG:

CORBA-INV-NO-RESPONSE

The calling application does not wait for a response. None of the parameters output (inout or out) are updated.

This option can be specified even when the operation is not defined as oneway.

CORBA-INV-TERM-ON-ERR

If an error occurs with any of the requests, the remaining requests cannot be sent.

To determine whether the server application function has terminated, invoke CORBA- REQUEST-GET-RESPONSE, and CORBA-REQUEST-GET-NEXT-RESPONSE.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

If NULL is specified in REQ, CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure.

4.12.2 CORBA-GET-NEXT-RESPONSE

Name

CORBA-GET-NEXT-RESPONSE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY FLAGS IN CORBA REPLACING CORBA-FLAGS BY RESPONSE-FLAGS.
01 COPY REQUEST IN CORBA REPLACING CORBA-REQUEST BY REQ.
01 COPY STATUS IN CORBA REPLACING CORBA-STATUS BY ORB-STATUS.
```

PROCEDURE DIVISION.

```
CALL "CORBA-GET-NEXT-RESPONSE" USING
ENV
RESPONSE-FLAGS
REQ
ORB-STATUS.
```

Description

If CORBA-REQUEST-SEND-MULTIPLE-REQUESTS is invoked to send a request, this function returns the object references of the request object (sent by CORBA-REQUEST-SEND-MULTIPLE-REQUESTS) related to the server application function to be terminated next. The order of terminated requests is not guaranteed.

The following flag can be specified in RESPONSE-FLAG:

CORBA-RESP-NO-WAIT

Returns even if the server application function is in progress.

Return Values

For normal termination, CORBA-OK is returned.

For abnormal termination, CORBA-FAILED is returned.

4.12.3 CORBA-XX-ALLOC

Name

CORBA-XX-ALLOC

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 RET-POINTER USAGE POINTER.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY STR-LEN.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SHORT-ALLOC" USING
RET-POINTER.
CALL "CORBA-LONG-ALLOC" USING
RET-POINTER.
CALL "CORBA-UNSIGNED-SHORT-ALLOC" USING
RET-POINTER.
CALL "CORBA-UNSIGNED-LONG-ALLOC" USING
RET-POINTER.
CALL "CORBA-LONG-LONG-ALLOC" USING
RET-POINTER.
CALL "CORBA-FLOAT-ALLOC" USING
RET-POINTER.
CALL "CORBA-DOUBLE-ALLOC" USING
RET-POINTER.
CALL "CORBA-CHAR-ALLOC" USING
RET-POINTER.
CALL "CORBA-WCHAR-ALLOC"  USING
RET-POINTER.
CALL "CORBA-BOOLEAN-ALLOC"  USING
RET-POINTER.
CALL  "CORBA-OCTECT-ALLOC" USING
RET-POINTER.
CALL "CORBA-ENUM-ALLOC" USING
RET-POINTER.
CALL "CORBA-ANY-ALLOC" USING
RET-POINTER.
CALL "CORBA-STRING-PTR-ALLOC" USING
RET-POINTER.
```

```
CALL "CORBA-WSTRING-PTR-ALLOC" USING
RET-POINTER.
CALL "CORBA-STRING-ALLOC" USING
STR-LEN
RET-POINTER.
CALL "CORBA-WSTRING-ALLOC" USING
STR-LEN
RET-POINTER.
CALL "CORBA-OBJECT-ALLOC" USING
RET-POINTER.
CALL "CORBA-TYPECODE-ALLOC" USING
RET-POINTER.
```

Description

This function (CORBA-STRING-ALLOC and CORBA-WSTRING-ALLOC) retrieves the area of the number of characters specified at argument, and returns its address.

The CORBA-STRING-PTR-ALLOC and CORBA-WSTRING-PTR-ALLOC allocate the storage area with the pointer of CORBA-STRING or CORBA-WSTRING, and returns its address.

Other CORBA-XX-ALLOC functions retrieve the address and size of the data type (denoted by XX).

Use the CORBA-FREE function to release the retrieved area here.

Return Value

If the program ends normally, the address of the acquired area is returned. If not, NULL is returned.

4.12.4 CORBA-SEQUENCE-XX-ALLOC

Name

CORBA-SEQUENCE-XX-ALLOC

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 RET-POINTER USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SEQUENCE-OCTET-ALLOC" USING
RET-POINTER.
CALL "CORBA-SEQUENCE-CHAR-ALLOC" USING
RET-POINTER.
CALL "CORBA-SEQUENCE-STRING-ALLOC" USING
RET-POINTER.
```

Description

The CORBA-SEQUENCE-XX-ALLOC function allocates the sequence data management area defined with the name XX and returns the address of the area.

Return Value

If it is defined in advance in each IDL file, the basic type sequence is automatically generated during IDL compilation.

4.12.5 CORBA-SEQUENCE-XX-ALLOCBUF

Name

CORBA-SEQUENCE-XX-ALLOCBUF

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 RET-POINTER USAGE POINTER.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY T-LENGTH.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SEQUENCE-OCTET-ALLOCBUF" USING
T-LENGTH
RET-POINTER.
CALL "CORBA-SEQUENCE-CHAR-ALLOCBUF" USING
T-LENGTH
RET-POINTER.
CALL "CORBA-SEQUENCE-STRING-ALLOCBUF" USING
T-LENGTH
RET-POINTER.
```

Description

This function retrieves data in the format requested (denoted by XX). The size of the data (for the array specified in SEQ-BUFFER which is in sequence type data format) is specified in T-LENGTH, and its address is returned. For fixed length sequence data, use the CORBA-SEQUENCE-XX-no or elements-ALLOCBUF functions.

Use the CORBA-FREE function to release the retrieved area.

Note

Standard type sequence is auto generated during IDL compilation, by defining it in each IDL file.

4.12.6 CORBA-FREE

Name

CORBA-FREE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 ADDR USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-FREE" USING
ADDR.
```

Description

This function releases an active COBOL data area of the specified size.

Return Values

None.

4.12.7 CORBA-ANY-GET-RELEASE

Name

CORBA-ANY-GET-RELEASE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 SEQ-POINTER USAGE POINTER.
01 COPY BOOLAEN IN CORBA REPLACING CORBA-BOOLEAN BY REL-FLAG.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ANY-GET-RELEASE" USING
SEQ-POINTER
REL-FLAG.
```

Description

This function references the release flag in the area specified by SEQ-POINTER, and returns either of the following as the response:

CORBA-TRUE-VALUE Released

CORBA-FALSE-VALUE Not released

Refer to information on "*any* Type and Sequence Type Release Flags" in the Distributed Application Development Guide (CORBA Service Edition) for details of the release flag.

4.12.8 CORBA-ANY-SET-RELEASE

Name

CORBA-ANY-SET-RELEASE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 SEQ-POINTER USAGE POINTER.
01 COPY BOOLAEN IN CORBA REPLACING CORBA-BOOLEAN BY REL-FLAG.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ANY-SET-RELEASE" USING
SEQ-POINTER
REL-FLAG.
```

Description

This function sets the release flag to specify whether the area specified by SEQ-POINTER has been released.

The following values can be set in REL-FLAG:

CORBA-TRUE-VALUE Released

CORBA-FALSE-VALUE Not released

Refer to information on "*any* Type and Sequence Type Release Flags" in the Distributed Application Development Guide (CORBA Service Edition) for details of the release flag.

4.12.9 CORBA-SEQUENCE-GET-RELEASE

Name

CORBA-SEQUENCE-GET-RELEASE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 SEQ-POINTER USAGE POINTER.
01 COPY BOOLAEN IN CORBA REPLACING CORBA-BOOLEAN BY REL-FLAG.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SEQUENCE-GET-RELEASE" USING
SEQ-POINTER
REL-FLAG.
```

Description

This function references the release flag in the area specified by SEQ-POINTER, and returns either of the following as the response:

CORBA-TRUE-VALUE Released

CORBA-FALSE-VALUE Not released

Refer to information on "*any* Type and Sequence Type Release Flags" in the Distributed Application Development Guide (CORBA Service Edition) for details of the release flag.

4.12.10 CORBA-SEQUENCE-SET-RELEASE

Name

CORBA-SEQUENCE-SET-RELEASE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 SEQ-POINTER USAGE POINTER.
01 COPY BOOLAEN IN CORBA REPLACING CORBA-BOOLEAN BY REL-FLAG.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SEQUENCE-SET-RELEASE" USING
SEQ-POINTER
REL-FLAG.
```

Description

This function sets the release flag to specify whether the area specified by SEQ-POINTER has been released.

The following values can be set in REL-FLAG:

CORBA-TRUE-VALUE Released

CORBA-FALSE-VALUE Not released

Refer to information on "*any* Type and Sequence Type Release Flags" in the Distributed Application Development Guide (CORBA Service Edition) for details of the release flag.

4.12.11 CORBA-ORB-SET-CLIENT-TIMER

Name

CORBA-ORB-SET-CLIENT-TIMER

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
  COPY CONST IN CORBA.
  01 COPY ORB         IN CORBA REPLACING CORBA-ORB BY ORB.
  01 COPY OBJECT      IN CORBA REPLACING CORBA-OBJECT BY OBJ.
  01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
  01 COPY LONG        IN CORBA REPLACING CORBA-LONG BY TIME.
```

PROCEDURE DIVISION.

```
    CALL "CORBA-ORB-SET-CLIENT-TIMER" USING
          ORB
          OBJ
          TIME
          ENV.
```

Description

In a client application, specify in TIME the period to wait until a server method corresponding to a server object specified in OBJ is returned. This waiting period is valid for all server objects running on a host specified in OBJ.

OBJ: Specifies the object reference of the server object for which the waiting period is set.

TIME: Specifies the period to wait in seconds until the server method is returned. If 0 is specified, the time until the server method is returned is not monitored.

Return Values

For normal termination, no value is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to "CORBA Service Exception Information" and "CORBA Service Minor Codes" in the Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.12.12 CORBA-ORB-SET-CLIENT-REQUEST-TIMER

Name

CORBA-ORB-SET-CLIENT-REQUEST-TIMER

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
  COPY CONST IN CORBA.
  01 COPY ORB         IN CORBA REPLACING CORBA-ORB BY ORB.
  01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
  01 COPY LONG        IN CORBA REPLACING CORBA-LONG BY TIME.
```

PROCEDURE DIVISION.

```
    CALL "CORBA-ORB-SET-CLIENT-REQUEST-TIMER" USING
          ORB
          TIME
          ENV.
```

Description

This function sets the response time, during which the client application will wait until the server method returns. This time is actually specified by the parameter time. The response time set here is effective on all the requests that are issued by the thread that invokes this function. Once this function is invoked, CORBA-ORB-CLEAR-CLIENT-REQUEST-TIMER() must also be invoked before the thread terminates.

Parameters

ORB

The ORB object reference acquired by CORBA-ORB-INIT().

TIME

Response time until the server method returns (seconds). The response time can be specified by a value from 0 to 100000000. If 0 is specified, the response time until the server method returns will not be monitored.

ENV

A structure that may contain exception information.

Return Values

When this method terminates normally, no value is returned.

When the method does not terminate normally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

Notes

- Use this method if you need to change the response time, during which the client process waits.
- When invoked before requests are sent to the server, this method sets the response time for the thread.
- If this method is invoked, CORBA-ORB-CLEAR-CLIENT-REQUEST-TIMER() must also be invoked before the thread terminates; otherwise, a memory leak will occur.
- To change the timeout value once this method is invoked, re-invoke CORBA-ORB-SET-CLIENT-REQUEST-TIMER(). When doing so, you do not need to invoke CORBA-ORB-CLEAR-CLIENT-REQUEST-TIMER() beforehand.
- The priorities of response time settings are indicated below, in descending order.
  1. Setting made by CORBA-ORB-SET-CLIENT-REQUEST-TIMER().
  2. Setting made by CORBA-ORB-SET-CLIENT-TIMER().
  3. Setting made by the period_receive_timeout parameter in the CORBA service environment setup file (config).

4.12.13 CORBA-ORB-GET-CLIENT-REQUEST-TIMER

Name

CORBA-ORB-GET-CLIENT-REQUEST-TIMER

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ORB         IN CORBA REPLACING CORBA-ORB BY ORB.
01 COPY OBJECT      IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
  CALL "CORBA-ORB-GET-CLIENT-REQUEST-TIMER" USING
       ORB
       OBJ
       ENV.
```

Description

This method acquires the server method response time (in seconds).

The priorities of response time settings are indicated below, in descending order. If NULL is assigned to OBJ, this method will acquire the setting 1 or 3.

1. Setting made by CORBA-ORB-SET-CLIENT-REQUEST-TIMER()
2. Setting made by CORBA-ORB-SET-CLIENT-TIMER()
3. Setting made by the period_receive_timeout parameter in the CORBA service environment setup file (config)

Use this function when you want to see the response time of a server method in a client application.

Parameters

ORB

The ORB object reference acquired by CORBA-ORB-INIT().

OBJ

The object reference to the server object whose response time is to be acquired.

ENV

A structure that may contain exception information.

Return Values

When this method terminates normally, no value is returned.

When the method does not terminate normally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.12.14 CORBA-ORB-CLEAR-CLIENT-REQUEST-TIMER

Name

CORBA-ORB-CLEAR-CLIENT-REQUEST-TIMER

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ORB         IN CORBA REPLACING CORBA-ORB BY ORB.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
  CALL "CORBA-ORB-CLEAR-CLIENT-REQUEST-TIMER" USING
       ORB
       ENV.
```

Description

This function resets the server method response time set by CORBA-ORB-SET-CLIENT-REQUEST-TIMER(). After execution of this function, the server method response time (during which requests issued by the thread that invoked this function will wait) returns to the system default value.

If the server method response time was set by CORBA-ORB-SET-CLIENT-REQUEST-TIMER(), you need to invoke this function before terminating the thread. This function does nothing if it is invoked by a thread that has not set any server method response time by CORBA-ORB-SET-CLIENT-REQUEST-TIMER().

Parameters

ORB

The ORB object reference acquired by CORBA-ORB-INIT().

ENV

A structure that may contain exception information.

Return Values

The server method response time returns.

When this method does not terminate normally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID and MINOR. For details on ID and MINOR, refer to the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

4.12.15 FJ-IMPLEMENTATIONREP-LOOKUP-ID

Name

FJ-IMPLEMENTATIONREP-LOOKUP-ID

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY FJ-IMPLEMENTATIONDEF IN CORBA REPLACING FJ- IMPLEMENTATIONDEF
BY IMPL-REP.
01 IMPL-REPID USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY IMPLEMENTATIONDEF IN CORBA REPLACING CORBA-IMPLEMENTATIONDEF
BY IMPL.
```

PROCEDURE DIVISION.

```
CALL " FJ-IMPLEMENTATIONREP-LOOKUP-ID" USING
IMPL-REP
IMPL-REPID
ENV
IMPL.
```

Description

This function returns the CORBA-IMPLEMENTATIONDEF object corresponding to the Implementation Repository ID specified in IMPL-REPID.

The CORBA-IMPLEMENTATIONDEF object is the Implementation Repository ID of the server application registered using the *OD_impl_inst* command.

Return Values

For normal termination, the CORBA-IMPLEMENTATIONDEF object corresponding to IMPL-REPID is returned. For abnormal termination, CORBA-NO-IMPLEMENT is set in MAJOR in the ENV structure.

Note

Specify the pointer clause of the Implementation Repository ID in the second parameter of the FJ-IMPLEMENTATIONREP-LOOKUP-ID function.

4.13 COBOL Extended Interface

This section describes the COBOL extended interface.

4.13.1 CORBA-EXCEPTION-ID

Name

CORBA-EXCEPTION-ID

Synopsis

ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 EXCEP USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-EXCEPTION-ID" USING
ENV
EXCEP.
```

Description

This function returns identifiers indicating the exceptions specified in the ENV structure.

Return Values

If an exception is set, an identifier indicating the exception in the ENV structure is returned. If an exception is not set (CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure), a NULL pointer is returned.

4.13.2 CORBA-EXCEPTION-VALUE

Name

CORBA-EXCEPTION-VALUE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 EXCEP USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-EXCEPTION-VALUE" USING
ENV
EXCEP.
```

Description

This function returns identifiers indicating the exceptions specified in the ENV structure.

Return Values

If an exception has been set, a parameter indicating the exception in the ENV structure is returned. If an exception has not been set (CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure), a NULL pointer is returned.

4.13.3 CORBA-ALLOC

Name

CORBA-ALLOC

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY A-SIZE.
01 ADDR USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-ALLOC" USING
A-SIZE
ADDR.
```

Description

This function acquires an active COBOL data area of the specified size.

Return Values

For normal termination, a pointer to the acquired data is returned.

For abnormal termination, a NULL pointer is returned.

4.13.4 CORBA-EXCEPTION-FREE

Name

CORBA-EXCEPTION-FREE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "CORBA-EXCEPTION-FREE" USING
ENV.
```

Description

This function releases the parameter area set in the ENV structure.

Return Values

None.

4.13.5 CORBA-SEQUENCE-ELEMENT-GET

Name

CORBA-SEQUENCE-ELEMENT-GET

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 SEQ-P USAGE POINTER.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY S-INDEX.
01 ELEMENT-TYPE USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SEQUENCE-ELEMENT-GET" USING
SEQ-P
S-INDEX
ELEMENT-TYPE.
```

Description

This function retrieves the contents specified in S-INDEX from the Sequence specified in SEQ-P.

Return Values

For normal termination, a pointer to the elements set in Sequence is returned. If the elements specified in S-INDEX do not exist, a NULL pointer is returned.

4.13.6 CORBA-SEQUENCE-ELEMENT-SET

Name

CORBA-SEQUENCE-ELEMENT-SET

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 SEQ-P USAGE POINTER.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY S-INDEX.
01 ELEMENT-TYPE USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "CORBA-SEQUENCE-ELEMENT-SET" USING
SEQ-P
S-INDEX
ELEMENT-TYPE.
```

Description

This function sets the S-INDEX elements of the Sequence specified in SEQ-P.

Return Values

For normal termination, a pointer to the elements set in Sequence is returned. If the elements specified in the S-INDEX do not exist, a NULL pointer is returned.

4.13.7 CORBA-STRING-GET

Name

CORBA-STRING-GET

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 STRING-IN USAGE POINTER.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY S-LENGTH.
01 STRING-OUT PIC X(n)
```

PROCEDURE DIVISION.

```
CALL "CORBA-STRING-GET" USING
STRING-IN
S-LENGTH
STRING-OUT.
```

Description

This function copies the string from STRING-IN to STRING-OUT, and either truncates or adds padding depending on the size of STRING-IN and STRING-OUT. STRING-OUT is PIC X (n). The size is indicated by S-LENGTH.

Return Values

For normal termination, a character string is retrieved in STRING-OUT. If the length of STRING-IN is less than S-LENGTH, STRING-IN is copied with the remaining length space padded.

The length of S-LENGTH must not exceed that of STRING-OUT.

4.13.8 CORBA-STRING-SET

Name

CORBA-STRING-SET

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 STRING-OUT USAGE POINTER.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY S-LENGTH.
01 STRING-IN PIC X(n)
```

PROCEDURE DIVISION.

```
CALL "CORBA-STRING-SET" USING
STRING-OUT
S-LENGTH
STRING-IN.
```

Description

This function creates a new string in STRING-OUT from the string in STRING-IN. A character string, whose length is specified in S-LENGTH, is stored in STRING-OUT. When storing the character string, this function adds a null pointer to STRING-OUT.

The size of S-LENGTH must include the string null pointer terminator. This means that STRING-OUT size must be such that the string length plus the string null pointer terminator can be stored. STRING-OUT is PIC X (n). The size is indicated by S-LENGTH.

Return Values

For normal termination, a character string set in STRING-OUT is stored.

4.13.9 CORBA-WSTRING-GET

Name

CORBA-WSTRING-GET

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 STRING-IN USAGE POINTER.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY S-LENGTH.
01 STRING-OUT PIC N(n)
```

PROCEDURE DIVISION.

```
CALL "CORBA-WSTRING-GET" USING
STRING-IN
S-LENGTH
STRING-OUT.
```

Description

This function copies the string characters from STRING-IN to STRING-OUT, and truncates or adds padding depending on the size of STRING-IN and STRING-OUT.

The size (PIC N(n)) of STRING-OUT is specified in S-LENGTH.

Return Values

For normal termination, a character string is retrieved in STRING-OUT. If the length of STRING-IN is less than S-LENGTH, STRING-IN is copied up to a null pointer, with the remaining length space padded.

The length of S-LENGTH must not exceed that of STRING-OUT.

4.13.10 CORBA-WSTRING-SET

Name

CORBA-WSTRING-SET

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 STRING-OUT USAGE POINTER.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY S-LENGTH.
01 STRING-IN PIC N(n)
```

PROCEDURE DIVISION.

```
CALL "CORBA-WSTRING-SET" USING
STRING-OUT
S-LENGTH
STRING-IN.
```

Description

This function takes a string of size S-LENGTH from the string specified in STRING-IN, stores it in STRING-OUT, and appends a NULL pointer. The size of STRING-OUT must be S-LENGTH + 1 (for the NULL pointer).

The size (PIC N(n)) of STRING-OUT is specified in S-LENGTH.

Return Values

For normal termination, the string set in STRING-OUT is stored.

4.14 Server Application Interface Windows32 Solaris32 Linux32

This is not valid for Linux (64 bit).

This section describes the memory management application interfaces (API) required by server applications to transfer data to and from skeletons. The following APIs are available:

- string-type memory acquisition API (TDSTRINGALLOC)
- string area \0 addition API (TDSTRINGSET)
- string area \0 deletion API (TDSTRINGGET)
- Memory release API (TDFREE)
- SMO name acquisition API (TDGETSMONAME)
- Client identifier acquisition API (TDGETCLIENTID)
- User identifier acquisition API (TDGETUSERINFOMATION)
- Memory release API for sequence type data areas (TDFREESEQUENCEBUF)
- Memory get API for sequence type data areas (TDSEQUENCEELEMENTGET)
- Memory set API for sequence type data areas (TDSEQUENCEELEMENTSET)
- Session ID notice API(TDGETSESSIONID)
- Session continuation declaration API(TDSETCONTCVT)
- Session ID reference API(TDREFSESSIONID)

Each API is described below.

4.14.1 TDSTRINGALLOC Windows32 Solaris32 Linux32

Name

TDSTRINGALLOC

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
COPY TDDEF IN TD.
01 COPY ULONG IN CORBA-REPLACING CORBA-UNSIGNED-LONG BY LENGTH.
01 ADDR USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "TDSTRINGALLOC" USING
LENGTH
ADDR.
```

Description

This API is equivalent to CORBA-STRING-ALLOC. It acquires a dynamic string area of the size specified in SIZE.

Return Values

For normal termination, a pointer to the acquired data is returned.

For abnormal termination, a NULL pointer is returned.

4.14.2 TDSTRINGSET Windows32 Solaris32 Linux32

Name

TDSTRINGSET

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
COPY TDDEF IN TD.
01 OUT-STRING USAGE POINTER.
01 COPY ULONG IN CORBA-REPLACING CORBA-UNSIGNED-LONG BY LENGTH.
01 IN-STRING PIC X(10).
```

PROCEDURE DIVISION.

```
MOVE "ABCDEF" TO IN-STRING.
MOVE FUNCTION LENG ( IN-STRING ) TO LENGTH.
CALL "TDSTRINGSET" USING
OUT-STRING
LENGTH
IN-STRING.
```

Description

This API is equivalent to CORBA-STRING-SET. It extracts a character string of a length specified in LENGTH, from the character string specified in IN-STRING, and assigns it to OUT-STRING. \0 is added before it is assigned. The area for OUT-STRING is dynamically acquired within this API. LENGTH must be shorter than or equal to IN-STRING.

The area acquired by this API must be released when no longer necessary.

Return Values

For normal termination, \0 is appended to the end of the character string assigned to IN- STRING, and the character string is assigned to OUT-STRING.

4.14.3 TDSTRINGGET Windows32 Solaris32 Linux32

Name

TDSTRINGGET

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
COPY TDDEF IN TD.
01 IN-STRING USAGE POINTER.
```

```
01 COPY ULONG IN CORBA-REPLACING CORBA-UNSIGNED-LONG BY LENGTH.
01 OUT-STRING PIC X(10).
```

PROCEDURE DIVISION.

```
MOVE FUNCTION LENG ( IN-STRING ) TO LENGTH.
CALL "TDSTRINGSET" USING
IN-STRING
LENGTH
OUT-STRING.
```

Description

This API is equivalent to CORBA-STRING-GET. It extracts a character string with a length specified in LENGTH, from the character string (ending with \0) specified in IN-STRING, then assigns the character string to OUT-STRING. \0 is deleted from the character string in IN-STRING before assigning it.

The area for OUT-STRING must be longer than the character string in IN-STRING. If the area for OUT-STRING is longer, a character string with the length specified in LENGTH is assigned to the area, with its remaining part left blank.

LENGTH must be shorter than or equal to OUT-STRING.

Return Values

For normal termination, the character string in IN-STRING from which \0 is removed is assigned to OUT-STRING.

4.14.4 TDFREE Windows32 Solaris32 Linux32

Name

TDFREE

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
COPY TDDEF IN TD.
01 ADDR USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "TDFREE" USING
ADDR.
```

Description

This API is equivalent to CORBA-FREE. It releases the area for the specified pointer.

Return Values

None.

4.14.5 TDGETSMONAME Windows32 Solaris32 Linux32

Name

TDGETSMONAME

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
COPY TDDEF IN TD.
01 SMONAME PIC X(256).
01 SMONAME-P USAGE POINTER.
```

PROCEDURE DIVISION.

```
MOVE FUNCTION ADDR (SMONAME) TO SMONAME-P.
CALL "TDGETCLIENTID" USING SMONAME-P.
```

Description

This API sets the SMO name to the specified area.

Return Values

For normal termination, 0 is returned.

For abnormal termination, one of the following is returned:

1: Parameter Error

2: Protocol Error

3: Sequence Error

99: System Error

4.14.6 TDGETCLIENTID 

Name

TDGETCLIENTID

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
01 ISTD-CLIENTID.
02 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-MAXIMUM.
02 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-LENGTH.
02 SEQ-BUFFER USAGE POINTER.
02 FILLER OCCURS 48
03 COPY OCTET IN CORBA REPLACING CORBA-OCTET BY CLIENTID-VALUE.
02 SEQ-LENGTH  PIC S9(9) COMP-5.
01 RETCODE PIC S9(9)COMP-5.
```

PROCEDURE DIVISION.

```
MOVE FUNCTION ADDR(CLIENT-VALUE(1)) TO SEQ-BUFFER
MOVE 48 TO SEQ-MAXIMUM OF CLIENTID.
CALL "TDGETCLIENTID" USING
SMONAME-ISTD-CLIENTID
RETCODE.
```

Description

This API sets a client ID (CLIENT-VALUE) in ISTD-CLIENTID. The length of the client ID is returned in SEQ-LENGTH.

Return Values

For normal termination, 0 is returned.

For abnormal termination, one of the following is returned:

1: Parameter Error

2: Sequence Error

3: Protocol Error

99: System Error

4.14.7 TDGETUSERINFORMATION Windows32 Solaris32 Linux32

Name

TDGETUSERINFORMATION

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
01 USERINFO.
02 DATALENGTH PIC S9(9) COMP-5
02 USERID PIC X(24) VALUE SPACE.
02 RETCODE PIC S9(9) COMP-5
```

PROCEDURE DIVISION.

```
CALL "TDGETUSERINFORMATION" USING
USERINFO
RETCODE.
```

Description

This API sets the length of the user identification information of the area specified in USERINFO to DATALENGTH, and the user identifier to USERID.

Return Values

For normal termination, 0 is returned.

For abnormal termination, one of the following is returned:

1: Parameter Error

2: Protocol Error

3: Sequence Error

99: System Error

4.14.8 TDFREESEQUENCEBUF Windows32 Solaris32 Linux32

Name

TDFREESEQUENCEBUF

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
01 ADDR USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "TDFREESEQUENCEBUF" USING
ADDR.
```

Description

Releases COBOL data type variable length sequence type data areas.

Return Values

None

4.14.9 TDSEQUENCEELEMENTGET Windows32 Solaris32 Linux32

Name

TDSEQUENCEELEMENTGET

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
01 SEQ-P USAGE POINTER.
01 DATA-P USAGE POINTER.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY ELM-NUM.
```

PROCEDURE DIVISION.

```
CALL "TDSEQUENCEELEMENTGET" USING
SEQ-P
ELM-NUM
DATA-P.
```

Description

Equivalent to CORBA-SEQUENCE-ELEMENT-GET. Fetches the contents of the element number specified in ELM-NUM from the sequence specified by SEQ-P.

Return Values

At normal termination, returns a pointer to the sequence element requested in DATA-P.

Returns a NULL pointer in DATA-P if the element specified by ELM-NUM is not found.

4.14.10 TDSEQUENCEELEMENTSET Windows32 Solaris32 Linux32

Name

TDSEQUENCEELEMENTSET

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
01 SEQ-P USAGE POINTER.
01 DATA-P USAGE POINTER.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY ELM-NUM.
```

PROCEDURE DIVISION.

```
CALL "TDSEQUENCEELEMENTSET" USING
SEQ-P
ELM-NUM
DATA-P.
```

Description

Equivalent to CORBA-SEQUENCE-ELEMENT-SET. Sets the data specified by DATA-P in the ELM-NUM element in the sequence specified by SEQ-P.

Return Values

At normal termination, returns the same value as before the DATA-P request.

Returns a NULL pointer in DATA-P if data could not be set in the element specified by ELM-NUM.

4.14.11 TDGETSESSIONID Windows32 Solaris32 Linux32

Name

TDGETSESSIONID

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
01 ISTD-SESSIONID USAGE POINTER.
01 SESSIONID.
  02 FILLER OCCURS 48
    03 COPY OCTET IN CORBA REPLACING CORBA-OCTET BY FILLER.
01 RETCODE PIC S9(9)COMP-5.
```

PROCEDURE DIVISION.

```
MOVE FUNCTION ADDR(SESSIONID) TO ISTD-SESSIONID.
CALL "TDGETSeSSIONID" USING
ISTD-SESSIONID
RETCODE.
```

Description

This API allocates and notifies the keys (session IDs) required to operate the following functions:

- Process pointer
- Session information management
- Windows32 Solaris32
  AIM linkage session inheritance

This API sets a pointer to the session ID's memory location in ISTD-SESSIONID. It also sets the return code's memory location in RETCODE.

Return Values

Stores a session ID in the location specified by ISTD-SESSIONID.

Stores one of the following return values in RETCODE.

0: Normal termination

1: Parameter error

2: Protocol error (duplicate issues of TDGETSESSIONID)

99: System error

4.14.12 TDSETCONTCVT Windows32 Solaris32 Linux32

Name

TDSETCONTCVT

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
01 RETCODE PIC S9(9)COMP-5.
```

PROCEDURE DIVISION.

```
CALL "TDSETCONTCVT" USING
SESSIONID-P RETCODE.
```

Description

Issuing this API informs the system to continue the session. The return code memory location is set in RETCODE.

Return Values

Stores one of the following return values in RETCODE.

0: Normal termination

1: Parameter error

2: Protocol error (issue from a non resident application, process binding not defined, or session ID not notified)

99: System error

4.14.13 TDREFSESSIONID Windows32 Solaris32 Linux32

Name

TDREFSESSIONID

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
01 SESSIONID-P USAGE POINTER.
01 SESSIONID.
  02 FILLER OCCURS 48
    03 COPY OCTET IN CORBA REPLACING CORBA-OCTET BY FILLER.
01 RETCODE PIC S9(9)COMP-5.
```

PROCEDURE DIVISION.

```
MOVE FUNCTION ADDR(SESSIONID) TO SESSIONID-P.
CALL "TDREFSESSIONID" USING
SESSIONID-P
RETCODE.
```

Description

Allows a user application to refer to a pre-allocated session ID.

You must obtain the session information memory location before issuing the API.

The memory address for session continuing information is set in SESSIONID-P.

The return code memory location is set in RETCODE.

Return Values

Normal termination: Returns 0 and stores a session ID in the location specified by SESSIONID-P.

Stores one of the following return values in RETCODE.

0: Normal termination

1: Parameter error

99: System error

4.15 Current Interface Windows32 Solaris32

Type definition

```
* CURRENT Object Reference
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSTRANSACTIONS- CURRENT.
* Transaction Status
01 COPY ENUM IN CORBA REPLACING CORBA-ENUM BY COSTRANSACTIONS- STATUS.
88 COSTRANSACTIONS-STATUSACTIVE   VALUE 0.
88 COSTRANSACTIONS-STATUSMARKEDRO VALUE 1.
88 COSTRANSACTIONS-STATUSPREPARED VALUE 2.
88 COSTRANSACTIONS-STATUSCOMMITTE VALUE 3.
88 COSTRANSACTIONS-STATUSROLLEDBA VALUE 4.
88 COSTRANSACTIONS-STATUSUNKNOWN  VALUE 5.
88 COSTRANSACTIONS-STATUSNOTRANSA VALUE 6.
88 COSTRANSACTIONS-STATUSPREPARIN VALUE 7.
88 COSTRANSACTIONS-STATUSCOMMITTI VALUE 8.
88 COSTRANSACTIONS-STATUSROLLINGB VALUE 9.
```

However, the Database Linkage Service COBOL library is stored as follows:

Windows32

TD installation directory\ots\include\COBOL

Solaris32

$OTS_HOME/include/COBOL

4.15.1 COSTRANSACTIONS-CURRENT-BEGIN Windows32 Solaris32

Name

COSTRANSACTIONS-CURRENT-BEGIN

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSTRANSACTIONS.
COPY CONST IN CORBA.
01 COPY CURRENT IN CORBA REPLACING COSTRANSACTIONS-CURRENT BY CURRENTOBJ
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSTRANSACTIONS-CURRENT-BEGIN" USING
CURRENTOBJ
ENV.
```

Description

This function creates a new transaction.

Specify the object reference of the CURRENT interface obtained using CORBA-ORB- RESOLVE-INITIAL-REFERENCES (CORBA-ORB-OBJECTID-TRANCURRENT) in CURRENT.

If no timeout observation time is set using COSTRANSACTIONS-CURRENT-SET-TIMEOUT, the timeout time specified for TRAN_TIME_OUT in the operating environment file is assumed.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

The values of ID are described as follows:

EX-COSTRANSACTIONS-SUBTRANSACT:

An attempt was made to create a nested transaction.

EX-CORBA-STEXCEP-NO-IMPLEMENT:

The Database Linkage Service system is not started.

EX-CORBA-STEXCEP-COMM-FAILURE:

The following causes are possible:

- A communication error occurred.
- Host information on which Database Linkage Service system exists, is not defined in OD environment file; initial_hosts (inithost on PC).

EX-CORBA-STEXCEP-NO-RESOURCES:

A resource became insufficient. The maximum number of transactions may have been exceeded.

EX-CORBA-STEXCEP-NO-MEMORY:

Failed to secure dynamic memory.

4.15.2 COSTRANSACTIONS-CURRENT-COMMIT Windows32 Solaris32

Name

COSTRANSACTIONS-CURRENT-COMMIT

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSTRANSACTIONS.
COPY CONST IN CORBA.
01 COPY CURRENT IN CORBA REPLACING COSTRANSACTIONS-CURRENT BY CURRENTOBJ
01 COPY BOOLEAN IN CORBA REPLACING CORBA-BOOLEAN BY REPORT-HEURISTICS.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSTRANSACTIONS-CURRENT-COMMIT" USING
CURRENTOBJ
```

```
REPORT-HEURISTICS
ENV.
```

Description

This function commits a transaction.

Specify the object reference of the CURRENT interface obtained using CORBA-ORB- RESOLVE-INITIAL-REFERENCES (CORBA-ORB-OBJECTID-TRANCURRENT) in CURRENT. If CORBA-TRUE is specified for REPORT-HEURISTICS, a heuristic error is reported.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

The values of ID are described as follows:

EX-COSTRANSACTIONS-NOTRANSACTI:

No transaction has been created.

EX-CORBA-STEXCEP-NO-PERMISSION:

Permission to commit transactions is denied.

EX-COSTRANSACTIONS-HEURISTICMI:

Some transactions are in the commit completion state, and others are in the rollback completion state.

EX-COSTRANSACTIONS-HEURISTICHA:

The transaction completion status is unknown.

EX-CORBA-STEXCEP-TRANSACTION-R:

The transaction has been rolled back.

EX-CORBA-STEXCEP-NO-IMPLEMENT:

The Database Linkage Service system or Resource Manager is not started.

EX-CORBA-STEXCEP-COMM-FAILURE:

A communication error has occurred.

EX-CORBA-STEXCEP-NO-RESOURCES:

A resource became insufficient. The maximum number of transactions may have been exceeded.

EX-CORBA-STEXCEP-NO-MEMORY:

Failed to secure dynamic memory.

4.15.3 COSTRANSACTIONS-CURRENT-ROLLBACK Windows32 Solaris32

Name

COSTRANSACTIONS-CURRENT-ROLLBACK

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSTRANSACTIONS.
COPY CONST IN CORBA.
01 COPY CURRENT IN CORBA REPLACING COSTRANSACTIONS-CURRENT BY CURRENTOBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSTRANSACTIONS-CURRENT-ROLLBACK" USING
CURRENTOBJ
ENV.
```

Description

This function rolls back a transaction.

Specify the object reference of the CURRENT interface obtained using CORBA-ORB- RESOLVE-INITIAL-REFERENCES (CORBA-ORB-OBJECTID-TRANCURRENT) in CURRENT.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

The values of ID are described as follows:

EX-COSTRANSACTIONS-NOTRANSACTI:

No transaction has been created.

EX-CORBA-STEXCEP-NO-PERMISSION:

Permission to roll back transactions is denied.

EX-CORBA-STEXCEP-TRANSACTION-R:

The transaction has been rolled back.

EX-CORBA-STEXCEP-NO-IMPLEMENT:

The Database Linkage Service system or Resource Manager is not started.

EX-CORBA-STEXCEP-COMM-FAILURE:

A communication error has occurred.

EX-CORBA-STEXCEP-NO-RESOURCES:

A resource became insufficient. The maximum number of transactions may have been exceeded.

EX-CORBA-STEXCEP-NO-MEMORY:

Failed to secure dynamic memory.

4.15.4 COSTRANSACTIONS-CURRENT-ROLLBACK-ONLY 

Name

COSTRANSACTIONS-CURRENT-ROLLBACK-ONLY

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSTRANSACTIONS.
COPY CONST IN CORBA.
01 COPY CURRENT IN CORBA REPLACING COSTRANSACTIONS-CURRENT BY CURRENTOBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSTRANSACTIONS-CURRENT-ROLLBACK-ONLY" USING
CURRENTOBJ
ENV.
```

Description

This function enables the transaction rollback operation only.

Specify the object reference of the CURRENT interface obtained using the CORBA-ORB-RESOLVE-INITIAL-REFERENCES method (CORBA-ORB-OBJECTID-TRANCURRENT) in CURRENT.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

The values of ID are described as follows:

EX-COSTRANSACTIONS-NOTRANSACTI:

No transaction has been created.

EX-CORBA-STEXCEP-TRANSACTION-R:

The transaction has been rolled back.

EX-CORBA-STEXCEP-NO-IMPLEMENT:

The Database Linkage Service system is not started.

EX-CORBA-STEXCEP-COMM-FAILURE:

A communication error has occurred.

EX-CORBA-STEXCEP-NO-RESOURCES:

A resource became insufficient. The maximum number of transactions may have been exceeded.

EX-CORBA-STEXCEP-NO-MEMORY:

Failed to secure dynamic memory.

4.15.5 COSTRANSACTIONS-CURRENT-GET-STATUS Windows32 Solaris32

Name

COSTRANSACTIONS-CURRENT-GET-STATUS

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSTRANSACTIONS.
COPY CONST IN CORBA.
01 COPY CURRENT IN CORBA REPLACING COSTRANSACTIONS-CURRENT BY CURRENTOBJ.
01 COPY STATUS IN  CORBA REPLACING COSTRANSACTIONS-STATUS  BY TRANSTATUS.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSTRANSACTIONS-CURRENT-GET-STATUS" USING
CURRENTOBJ
ENV
TRANSTATUS.
```

Description

This function returns the status of a transaction.

Specify the object reference of the CURRENT interface obtained using the CORBA-ORB-RESOLVE-INITIAL-REFERENCES method (CORBA-ORB-OBJECTID-TRANCURRENT) in CURRENT.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure and one of the following values is returned:

COSTRANSACTIONS-STATUSACTIVE:

The transaction is being executed.

COSTRANSACTIONS-STATUSMARKEDRO:

The transaction is marked rollback.

COSTRANSACTIONS-STATUSPREPARED:

The transaction is prepared.

COSTRANSACTIONS-STATUSCOMMITTE:

The transaction has been committed.

COSTRANSACTIONS-STATUSROLLEDBA:

The transaction has been rolled back.

COSTRANSACTIONS-STATUSUNKNOWN:

The transaction status is unknown.

COSTRANSACTIONS-STATUSNOTRANSA:

No transaction is being executed.

COSTRANSACTIONS-STATUSPREPARIN:

Prepare processing is being executed.

COSTRANSACTIONS-STATUSCOMMITTI:

Commit processing is being executed.

COSTRANSACTIONS-STATUSROLLINGB:

Rollback processing is being executed.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

The values of ID are described as follows:

EX-CORBA-STEXCEP-TRANSACTION-R:

The transaction has been rolled back.

EX-CORBA-STEXCEP-NO-IMPLEMENT:

The Database Linkage Service system is not started.

EX-CORBA-STEXCEP-COMM-FAILURE:

A communication error has occurred.

EX-CORBA-STEXCEP-NO-RESOURCES:

A resource became insufficient. The maximum number of transactions may have been exceeded.

EX-CORBA-STEXCEP-NO-MEMORY:

Failed to secure dynamic memory.

4.15.6 COSTRANSACTIONS-CURRENT-GET-TRANSACTION-NAME Windows32 Solaris32

Name

COSTRANSACTIONS-CURRENT-GET-TRANSACTION-NAME

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSTRANSACTIONS.
COPY CONST IN CORBA.
01 COPY CURRENT IN CORBA REPLACING COSTRANSACTIONS-CURRENT BY CURRENTOBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY STRING  IN CORBA REPLACING CORBA-STRING BY TRANSACTIONNAME.
```

PROCEDURE DIVISION.

```
CALL "COSTRANSACTIONS-CURRENT-GET-TRANSACTION-NAME" USING
CURRENTOBJ
ENV
TRANSACTIONNAME.
```

Description

This function returns a character string identifying a transaction.

Specify the object reference of the CURRENT interface obtained using the CORBA-ORB-RESOLVE-INITIAL-REFERENCES method (CORBA-ORB-OBJECTID-TRANCURRENT) in CURRENT.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure, and the character string identifying the transaction is returned.

If no transaction has been created, a NULL character is returned.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

The values of ID are described as follows:

EX-CORBA-STEXCEP-TRANSACTION-R:

The transaction has been rolled back.

EX-CORBA-STEXCEP-NO-IMPLEMENT:

The Database Linkage Service system is not started.

EX-CORBA-STEXCEP-COMM-FAILURE:

A communication error has occurred.

EX-CORBA-STEXCEP-NO-RESOURCES:

A resource became insufficient. The maximum number of transactions may have been exceeded.

EX-CORBA-STEXCEP-NO-MEMORY:

Failed to secure dynamic memory.

4.15.7 COSTRANSACTIONS-CURRENT-SET-TIMEOUT Windows32 Solaris32

Name

COSTRANSACTIONS-CURRENT-SET-TIMEOUT

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSTRANSACTIONS.
COPY CONST IN CORBA.
01 COPY CURRENT IN CORBA REPLACING COSTRANSACTIONS-CURRENT BY CURRENTOBJ.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY SECONDS.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSTRANSACTIONS-CURRENT-SET-TIMEOUT" USING
CURRENTOBJ
SECONDS
ENV.
```

Description

This function sets the value specified for SECONDS as the transaction timeout observation time.

If processing is not completed within the time set for SECONDS, the transaction is rolled back. If 0 is specified for SECONDS, no timeout is observed.

Specify the object reference of the CURRENT interface obtained using the CORBA-ORB-RESOLVE-INITIAL-REFERENCES method (CORBA-ORB-OBJECTID-TRANCURRENT) in CURRENT.

If COSTRANSACTIONS-CURRENT-SET-TIMEOUT is called while a transaction is being executed, the specified timeout value is not valid until the next transaction starts.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

EX-CORBA-STEXCEP-NO-MEMORY:

Failed to secure dynamic memory.

EX-CORBA-STEXCEP-INTERNAL:

A CORBA Service error occurred.

4.15.8 COSTRANSACTIONS-CURRENT-GET-CONTROL Windows32 Solaris32

Name

COSTRANSACTIONS-CURRENT-GET-CONTROL

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSTRANSACTIONS.
COPY CONST IN CORBA.
01 COPY CURRENT IN CORBA REPLACING COSTRANSACTIONS-CURRENT BY CURRENTOBJ.
01 COPY CONTROL IN CORBA REPLACING COSTRANSACTIONS-CONTROL BY CONTROLOBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSTRANSACTIONS-CURRENT-GET-CONTROL" USING
CURRENTOBJ
ENV
CONTROLOBJ.
```

Description

Returns the CONTROL object that manages the transaction context.

This CONTROL object can be used to complete transactions in server applications.

Specify in CURRENT the CURRENT interface object reference fetched by the CORBA-ORB-RESOLVE-INITIAL-REFERENCES method (CORBA-ORB-OBJECTID-TRANCURRENT).

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

EX-CORBA-STEXCEP-NO-IMPLEMENT:

The Database Linkage Service system could not be started.

EX-CORBA-STEXCEP-COMM-FAILURE:

A communications failure occurred.

EX-CORBA-STEXCEP-NO-RESOURCES:

There are insufficient resources.

EX-CORBA-STEXCEP-NO-MEMORY:

Failed to secure dynamic memory.

4.15.9 COSTRANSACTIONS-CURRENT-SUSPEND Windows32 Solaris32

Name

COSTRANSACTIONS-CURRENT-SUSPEND

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSTRANSACTIONS.
COPY CONST IN CORBA.
01 COPY CURRENT IN CORBA REPLACING COSTRANSACTIONS-CURRENT BY CURRENTOBJ.
01 COPY CONTROL IN CORBA REPLACING COSTRANSACTIONS-CONTROL BY CONTROLOBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSTRANSACTIONS-CURRENT-SUSPEND" USING
CURRENTOBJ
ENV
CONTROLOBJ.
```

Description

Disconnects a transaction generated by an application from the current thread and stops. Returns the CONTROL object that manages the transaction context that manages the transaction.

When this function terminates normally, the transaction remains valid until the RESUME function is run.

Specify in CURRENT the CURRENT interface object reference fetched by the CORBA-ORB-RESOLVE-INITIAL-REFERENCES method (CORBA-ORB-OBJECTID-TRANCURRENT).

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

EX-CORBA-STEXCEP-NO-MEMORY:

Failed to secure dynamic memory.

EX-CORBA-STEXCEP-UNKNOWN:

A CORBA Service initialization error occurred.

EX-CORBA-STEXCEP-INTERNAL:

A CORBA Service error occurred.

4.15.10 COSTRANSACTIONS-CURRENT-RESUME Windows32 Solaris32

Name

COSTRANSACTIONS-CURRENT-RESUME

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSTRANSACTIONS.
COPY CONST IN CORBA.
01 COPY CONTROL IN  CORBA REPLACING COSTRANSACTIONS-CONTROL BY CONTROLOBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSTRANSACTIONS-CURRENT-RESUME" USING
CURRENTOBJ
ENV.
```

Description

Associates a transaction with a current thread.

Specify in CONTROL the CONTROL object object reference fetched by the COSTRANSACTIONS-CURRENT-GET-CONTROL method or the CONTROL object object reference fetched by the COSTRANSACTIONS-CURRENT-SUSPEND method. The transaction related to this CONTROL object is associated with the current thread. If a NULL object is specified in CONTROL, the association between the transaction and the thread is removed.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

EX-CORBA-STEXCEP-INVALIDCONTROL:

An object other than a CONTROL object was specified.

EX-CORBA-STEXCEP-INTERNAL:

A CORBA Service error occurred

4.16 Transaction Initialization Interface Windows32 Solaris32

This is not valid for Linux (64 bit).

This section describes the transaction initialization interface.

4.16.1 OTS-INIT Windows32 Solaris32

Name

OTS-INIT

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OTS.
01 COPY ORB IN CORBA REPLACING CORBA-ORB BY ORB.
01 IMPL USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "OTS-INIT" USING
OTS
ORB
IMPL
ENV.
```

Description

This function connects a server application to the Database Linkage Service. Specify the Implementation Repository ID of the server application in IMPL.

Specify the object reference of the Database Linkage Service interface obtained using the CORBA-ORB-RESOLVE-INITIAL-REFERENCES method (CORBA-ORB-OBJECTID-TRANSVR) in the Database Linkage Service.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

EX-OTS-BAD-DESCRIPTION:

A description in the resource definition file is invalid.

EX-OTS-NO-DEFFILE:

The resource definition file cannot be found, or the resource definition file name is not specified as Implementation Repository.

EX-OTS-RM-ERROR:

A temporary error occurred during an attempt to open a database.

EX-OTS-PERMISSION-DENIED:

Permission to use the resource definition file is denied.

EX-OTS-IOERROR:

An I/O error occurred during an attempt to read the resource definition file.

Note

Do not call this function more than once in one process.

4.17 Transaction Termination Interface Windows32 Solaris32

This is not valid for Linux (64 bit).

This section describes the transaction termination interface.

4.17.1 OTS-TERM Windows32 Solaris32

Name

OTS-TERM

Synopsis

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OTS.
01 COPY ORB IN CORBA REPLACING CORBA-ORB BY ORB.
01 IMPL USAGE POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "OTS-TERM" USING
OTS
ENV.
```

Description

This function disconnects a server application from the Database Linkage Service.

Specify the object reference of the Database Linkage Service interface obtained using the CORBA-ORB-RESOLVE-INITIAL-REFERENCES method (CORBA-ORB-OBJECTID-TRANSVR) in the Database Linkage Service.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

4.18 Load Balance Function Interface Windows32 Solaris32 Linux32

This is not valid for Linux (64 bit).

This section explains the interfaces provided by the load balance function.

Note

The load balance function can only be used in the Interstage Application Server Enterprise Edition.

4.18.1 Load Balance Option Interface Windows32 Solaris32 Linux32

This section describes the load balance option interface.

4.18.1.1 ISOD-LBO-CREATE-LBG Windows32 Solaris32 Linux32

Name

ISOD-LBO-CREATE-LBG

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY COSNAMING-CONTEXT IN CORBA REPLACING COSNAMING-CONTEXT BY NC.
01 COPY COSNAMING-NAME IN CORBA REPLACING COSNAMING-NAME BY N.
01 COPY ENUM IN CORBA REPLACING CORBA-NUM BY LOADBALANCETYPE.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY DEFAULTOBJREF.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY RESULT.
```

PROCEDURE DIVISION.

```
CALL "ISOD-LBO-CREATE-LBG"  USING
OBJ
NC
N
LOADBALANCETYPE
DEFAULTOBJECTREF
ENV
RESULT.
```

Description

This function creates a load balance object group and registers it with the name specified in N under the naming context specified in NC.

If successful in generating the load balance object group, the object reference of the load balance object group is returned.

Specify ISOD-LBG-ROUNDROBIN in LOADBALANCETYPE.

If the load balance function is not operating in DEFAULTOBJECTREF, specify the default object reference to be returned by the Naming Service. If it is not specified, the object reference is not returned from the Naming Service because the load balance function is not operating.

The object reference cannot be registered in, or deleted from the BIND and UNBIND operations. If it is specified, it can be changed by the REBIND-DEFAULT operation after creating the load balance object group.

Use the OMG IDL format (format starting with "IDL:") for the Interface Repository ID of the object reference to be specified in DEFAULTOBJECTREF.

Since this function acquires area to store the object reference, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set to MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detail information is set in ID in the ENV structure.

The value of ID is described as follows.

IDL:ISOD/LBO/NotFound:1.0

The naming context specified in N could not be found.

IDL:ISOD/LBO/CannotProceed:1.0

The naming context specified in NC does not exist.

IDL:ISOD/LBO/InvalidName:1.0

Invalid name specification.

IDL:ISOD/LBO/AlreadyExist:1.0

A binding for the specified name already exists.

IDL:ISOD/LBO/InvalidType:1.0

Invalid specification of load balance type.

IDL:ISOD/LBO/BadObject:1.0

The object specified in DEFAULTOBJECTREF is invalid.

IDL:ISOD/LBO/OperationBusy:1.0

A request for multi-processing has reached the maximum limit. Retry later.

4.18.1.2 ISOD-LBO-RESOLVE-LBG Windows32 Solaris32 Linux32

Name

ISOD-LBO-RESOLVE-LBG

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY COSNAMING-CONTEXT IN CORBA REPLACING COSNAMING-CONTEXT BY NC.
01 COPY COSNAMING-NAME IN CORBA REPLACING COSNAMING-NAME BY N.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY RESULT.
```

PROCEDURE DIVISION.

```
CALL "ISOD-LBO-RESOLVE-LBG"  USING
OBJ
NC
N
ENV
RESULT.
```

Description

This function searches the load balance object group, and returns its object reference.

The load balance object group specified at N is searched for under the naming context specified at NC. If found, the load balance object group's object reference is returned.

Since this function acquires area to store the object reference, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detail information is set in ID in the ENV structure.

The values of ID are as follows:

IDL:ISOD/LBO/NotFound:1.0

Name specified in N could not be found.

IDL:ISOD/LBO/CannotProceed:1.0

Naming context specified in NC does not exist.

IDL:ISOD/LBO/InvalidName:1.0

Invalid name specification.

IDL:ISOD/LBO/OperationBusy:1.0

Request for multi-processing has reached the maximum limit. Retry later.

4.18.1.3 ISOD-LBO-DELETE-LBG Windows32 Solaris32 Linux32

Name

ISOD-LBO-DELETE-LBG

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY COSNAMING-CONTEXT IN CORBA REPLACING COSNAMING-CONTEXT BY NC.
01 COPY COSNAMING-NAME IN CORBA REPLACING COSNAMING-NAME BY N.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY RESULT.
```

PROCEDURE DIVISION.

```
CALL "ISOD-LBO-DELETE-LBG"  USING
OBJ
NC
N
ENV
RESULT.
```

Description

This function deletes the load balance object group from the Naming Service.

The load balance object group specified at N under the naming context specified at NC is deleted.

The default object is returned. If an object other than the default object is registered in the load balance object group to be deleted, an exception occurs.

Since this function acquires area to store the object reference, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detail information is set in ID in the ENV structure.

The values of ID are as follows:

IDL:ISOD/LBO/NotFound:1.0

Name specified in N could not be found.

IDL:ISOD/LBO/CannotProceed:1.0

Naming context specified in NC does not exist.

IDL:ISOD/LBO/InvalidName:1.0

Invalid name specification.

IDL:ISOD/LBO/NotEmpty:1.0

More than one object is registered in the object group.

IDL:ISOD/LBO/OperationBusy:1.0

Request for multi-processing has reached the maximum limit. Retry later.

4.18.1.4 ISOD-LBO-LIST-LBG Windows32 Solaris32 Linux32

Name

ISOD-LBO-LIST-LBG

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY COSNAMING-CONTEXT IN CORBA REPLACING COSNAMING-CONTEXT BY NC.
01 OBJECTGROUPLIST USAGE IS POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "ISOD-LBO-LIST-LBG"  USING
OBJ
NC
OBJECTGROUPLIST
ENV.
```

Description

This function returns a list of the load balance object group specified at n in the naming context specified at nc.

The load balance object group name, and the structure list (BINDINGOBJECTGROUP) containing information for the load balance type is returned.

The maximum number of load balance object group items that can be retrieved by this function is 128.

Since this function acquires area to store the list of the load balance object group, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in ENV in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set.

4.18.1.5 ISOD-LBO-NOTIFY-DOWN Windows32 Solaris32 Linux32

Name

ISOD-LBO-NOTIFY-DOWN

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY COSNAMING-CONTEXT IN CORBA REPLACING COSNAMING-CONTEXT BY NC.
01 HOSTNAME USAGE IS POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "ISOD-LBO-NOTIFY-DOWN"  USING
OBJ
HOSTNAME
ENV.
```

Description

This function notifies the load balance function that the server is down.

When this function is called, it stops the return of object references from objects on the server specified at HOSTNAME.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detail information is set in ID in the ENV structure.

The values of ID are as follows:

IDL:ISOD/LBO/InvalidArgument:1.0

The hostname specification is invalid.

IDL:ISOD/LBO/CannotProceed2:1.0

An abnormality occurred in the load balance function DB process.

4.18.1.6 ISOD-LBO-NOTIFY-RECOVER Windows32 Solaris32 Linux32

Name

ISOD-LBO-NOTIFY-RECOVER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY COSNAMING-CONTEXT IN CORBA REPLACING COSNAMING-CONTEXT BY NC.
01 HOSTNAME USAGE IS POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "ISOD-LBO-NOTIFY-RECOVER"  USING
OBJ
HOSTNAME
ENV.
```

Description

This function notifies the load balance function that the server has been restored.

When this function is called, it starts the return of object references from objects on the server specified at HOSTNAME.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detail information is set in ID in the ENV structure.

The values of ID are as follows:

IDL:ISOD/LBO/InvalidArgument:1.0

The hostname specification is invalid.

IDL:ISOD/LBO/CannotProceed2:1.0

An abnormality occurred in the load balance function DB process.

4.18.2 Load Balance Object Group Interface Windows32 Solaris32 Linux32

This section describes the load balance object group interface.

4.18.2.1 ISOD-LBG-BIND Windows32 Solaris32 Linux32

Name

ISOD-LBG-BIND

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJECTREF.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "ISOD-LBG-BIND"  USING
LBG
OBJECTREF
ENV.
```

Description

This function registers the object to be load balanced (specified at OBJECTREF) to the load balance object group specified at LBG.

Use OMG IDL format (format starting with "IDL:") for the Interface Repository ID of the object reference to be specified in OBJECTREF.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detail information is set in ID in the ENV structure.

The values of ID are as follows:

IDL:ISOD/LBG/AlreadyBound:1.0

An object containing the same information is already registered.

IDL:ISOD/LBG/CannotProceed2:1.0

An abnormality occurred in the load balance function DB process.

IDL:ISOD/LBG/BadObject:1.0

The object specified in OBJECTREF is invalid.

4.18.2.2 ISOD-LBG-UNBIND Windows32 Solaris32 Linux32

Name

ISOD-LBG-UNBIND

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJECTREF.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "ISOD-LBG-UNBIND"  USING
LBG
OBJECTREF
ENV.
```

Description

This function deletes the object to be load balanced (specified at OBJECTREF) from the load balance object group specified at LBG.

Use OMG IDL format (format starting with "IDL:") for the Interface Repository ID of the object reference to be specified in OBJECTREF.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detail information is set in ID in the ENV structure.

The values of ID are as follows:

IDL:ISOD/LBG/NotFound:1.0

The object specified at OBJECTREF could not be found.

IDL:ISOD/LBG/CannotProceed2:1.0

An abnormality occurred in the load balance function DB process.

IDL:ISOD/LBG/BadObject:1.0

The object specified in OBJECTREF is invalid.

4.18.2.3 ISOD-LBG-REBIND-DEFAULT 

Name

ISOD-LBG-REBIND-DEFAULT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJECTREF.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY RESULT.
```

PROCEDURE DIVISION.

```
CALL "ISOD-LBG-REBIND-DEFAULT"  USING
LBG
OBJECTREF
ENV
RESULT.
```

Description

This function changes the default object of the load balance object group specified at LBG to the object specified at OBJECTREF.

The object reference of the default object (before the change) is returned.

Use OMG IDL format (format starting with "IDL:") for the Interface Repository ID of the object reference to be specified in OBJECTREF.

Since this function acquires area to store the object reference, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detail information is set in ID in the ENV structure.

The values of ID are as follows:

IDL:ISOD/LBG/CannotProceed:1.0

The default object is not set.

IDL:ISOD/LBG/CannotProceed2:1.0

An abnormality occurred in the load balance function DB process.

IDL:ISOD/LBG/BadObject:1.0

The object specified in OBJECTREF is invalid.

IDL:ISOD/LBG/OperationBusy:1.0

The request for multi-processing has reached the maximum limit. Retry later.

4.18.2.4 ISOD-LBG-LIST 

Name

ISOD-LBG-LIST

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.

```
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 OBJECTLIST USAGE IS POINTER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "ISOD-LBG-LIST"  USING
LBG
OBJECTLIST
ENV.
```

Description

This function returns a list of the object references of objects already registered in the load balance object group specified at LBG.

Since this function acquires area to store the list of the object, use CORBA-FREE to release the area as soon as it is no longer needed.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR in the ENV structure.

For abnormal termination, CORBA-USER-EXCEPTION is set, and detail information is set in ID in the ENV structure.

The value of ID is as follows:

IDL:ISOD/LBG/CannotProceed2:1.0

An abnormality occurred in the load balance function DB process.

4.19 Event Service Interface

Type definitions

EVENTCHANNEL object reference

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSEVENTCHANNELADMIN-EVENT0001.
```

CONSUMERADMIN object reference

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSEVENTCHANNELADMIN-CONSUMERA.
```

SUPPLIERADMIN object reference

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSEVENTCHANNELADMIN-SUPPLIERA.
```

PULLCONSUMER object reference

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSEVENTCOMM- PULLCONSUMER.
```

PULLSUPPLIER object reference

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSEVENTCOMM- PULLSUPPLIER.
```

PUSHCONSUMER object reference

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSEVENTCOMM-PUSHCONSUMER.
```

PUSHSUPPLIER object reference

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSEVENTCOMM-PUSHSUPPLIER.
```

PROXYPUSHSUPPLIER object reference

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSEVENTCHANNELADMIN-PROXYP004.
```

PROXYPULLSUPPLIER object reference

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSEVENTCHANNELADMIN-PROXYP002.
```

PROXYPUSHCONSUMER object reference

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSEVENTCHANNELADMIN-PROXYP003.
```

PROXYPULLCONSUMER object reference

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSEVENTCHANNELADMIN-PROXYP001.
```

EVENTFACTORY object reference

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY EVENTFACTORY-CREATE.
```

4.19.1 COSEVENTCOMM Interface

This section describes the COSEVENTCOMM interface.

4.19.1.1 COSEVENTCOMM-PUSHCONSUMER-PUSH

Name

COSEVENTCOMM-PUSHCONSUMER-PUSH

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY PROXYPUSHCONSUMER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP003 BY OBJ.
COPY CONST IN CORBA.
01 COPY ANY IN CORBA REPLACING CORBA-ANY BY EVDATA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCOMM-PUSHCONSUMER-PUSH" USING
    OBJ
    EVDATA
    ENV.
```

Description

Transmits the event data specified by DATA to the consumer.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PUSH-CONSUMER.

EVDATA

The event data sent to the consumer.

ENV

A structure that may contain exception information.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detail information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSEVENTCOMM-DISCONNECTED

Not connected to the Event Channel.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the *esmkchnl* command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.1.2 COSEVENTCOMM-PUSHCONSUMER-DISCONNECT-PUSH-CONSUMER

Name

COSEVENTCOMM-PUSHCONSUMER-DISCONNECT-PUSH-CONSUMER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY PROXYPUSHCONSUMER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP003 BY OBJ.
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCOMM-PUSHCONSUMER-DISCONNECT-PUSH-CONSUMER" USING
    OBJ
    ENV.
```

Description

Declares termination of event communication by supplier.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PUSH-CONSUMER.

ENV

A structure that may contain exception information.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure. For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.1.3 COSEVENTCOMM-PUSHSUPPLIER-DISCONNECT-PUSH-SUPPLIER

Name

COSEVENTCOMM-PUSHSUPPLIER-DISCONNECT-PUSH-SUPPLIER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY PROXYPUSHSUPPLIER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP004 BY OBJ.
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCOMM-PUSHSUPPLIER-DISCONNECT-PUSH-SUPPLIER" USING
    OBJ
    ENV.
```

Description

Declares termination of event communication by consumer.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-CONSUMER ADMIN-OBTAIN-PUSH-SUPPLIER.

ENV

A structure that may contain exception information.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure. For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.1.4 COSEVENTCOMM-PULLSUPPLIER-PULL

Name

COSEVENTCOMM-PULLSUPPLIER-PULL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY PROXYPULLSUPPLIER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP002 BY OBJ.
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 DATA USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCOMM-PULLSUPPLIER-PULL" USING
    OBJ
    ENV
    EVDATA.
```

Description

Requests event data from supplier. This operation is blocked until event data can be fetched, or until an exception is generated. If the event data cannot be fetched and an immediate response is required, then please use COSEVENTCOMM-PULLSUPPLIER-TRY-PULL.

Use the CORBA-FREE function to release the area secured for the any type data when it is no longer required.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-CONSUMER ADMIN-OBTAIN-PULL-SUPPLIER.

ENV

A structure that may contain exception information.

EVDATA

The event data requested to the supplier.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA- SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detail information is set in ID in the ENV structure.

In the event of a user exception, the following detail information is specified.

EX-COSEVENTCOMM-DISCONNECTED

Not connected to the Event Channel.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the *esmkchnl* command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.1.5 COSEVENTCOMM-PULLSUPPLIER-TRY-PULL

Name

COSEVENTCOMM-PULLSUPPLIER-TRY-PULL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY PROXYPULLSUPPLIER IN COSEVENTCOMM REPLACING
```

```
COSEVENTCHANNELADMIN-PROXYP002 BY OBJ.
COPY CONST IN CORBA.
01 COPY BOOLEAN IN CORBA REPLACING CORBA-BOOLEAN BY HASEVENT.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 DATA USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCOMM-PULLSUPPLIER-TRY-PULL" USING
    OBJ
    HASEVENT
    ENV
    EVDATA.
```

Description

Requests event data from supplier. Instantly returned if event data cannot be fetched from supplier. If you want to block this operation until event data can be fetched, then please use COSEVENTCOMM-PULLSUPPLIER-PULL.

Use the CORBA-FREE function to release the area secured for the any type data when it is no longer required.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-CONSUMERADMIN-OBTAIN-PULL-SUPPLIER.

HASEVENT (out parameter)

When event data is fetched, CORBA-TRUE is set.

When event data is not fetched, CORBA-FALSE is set.

ENV

A structure that may contain exception information.

EVDATA

The event data requested to the supplier.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure. When event data has been fetched, CORBA-TRUE is set in HASEVENT. If event data has not been fetched, then CORBA-FALSE is set in HASEVENT. In this case, use the CORBA-FREE function to release the area secured for the any type data when it is no longer required.

For abnormal termination, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSEVENTCOMM-DISCONNECTED

Not connected to the Event Channel.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the *esmkchnl* command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Notes

If CORBA-FALSE is set to *HAS-EVENT* for a local transaction, a cancellation need not be sent using ES-CHANNELUTIL-LOCAL-ROLLBACK.

When the local transaction has completed, the channel is notified using ES-CHANNELUTIL-LOCAL-COMMIT .

Windows32 Solaris32

If CORBA-FALSE is set to *HAS-EVENT* for a global transaction, a cancellation need not be sent using COSTRANSACTIONS-CURRENT-ROLLBACK.

When the global transaction has completed, the channel is notified using COSTRANSACTIONS-CURRENT-COMMIT.

4.19.1.6 COSEVENTCOMM-PULLSUPPLIER-DISCONNECT-PULL-SUPPLIER

Name

COSEVENTCOMM-PULLSUPPLIER-DISCONNECT-PULL-SUPPLIER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY PROXYPULLSUPPLIER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP002 BY OBJ.
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCOMM-PULLSUPPLIER-DISCONNECT-PULL-SUPPLIER" USING
    OBJ
    ENV.
```

Description

Declares termination of event communication by consumer.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN- CONSUMERADMIN-OBTAIN-PULL-SUPPLIER.

ENV

A structure that may contain exception information.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.1.7 COSEVENTCOMM-PULLCONSUMER-DISCONNECT-PULL-CONSUMER

Name

COSEVENTCOMM-PULLCONSUMER-DISCONNECT-PULL-CONSUMER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY PROXYPULLCONSUMER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP001 BY OBJ.
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCOMM-PULLCONSUMER-DISCONNECT-PULL-CONSUMER" USING
    OBJ
    ENV.
```

Description

Declares termination of event communication by consumer.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PULL-CONSUMER.

ENV

A structure that may contain exception information.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.2 COSEVENTCHANNELADMIN Interface

This section describes the COSEVENTCHANNELADMIN interface.

4.19.2.1 COSEVENTCHANNELADMIN-EVENTCHANNEL-FOR-CONSUMERS

Name

COSEVENTCHANNELADMIN-EVENTCHANNEL-FOR-CONSUMERS

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY EVENTCHANNEL IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-CONSUMERA BY OBJ.
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCHANNELADMIN-EVENTCHANNEL-FOR-CONSUMERS" USING
    OBJ
    ENV
    RESULT.
```

Description

Acquires the object reference for the Event Channel, in order to connect the consumer to the Event Channel.

Since this method acquires area to store object references, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Parameters

OBJ

The object reference of the Event Channel to be connected.

ENV

A structure that may contain exception information.

RESULT

The object reference of the Event Channel is specified, in order to connect the consumer to the Event Channel.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM- EXCEPTION is set in MAJOR in the ENV structure, and a system exception is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.2.2 COSEVENTCHANNELADMIN-EVENTCHANNEL-FOR-SUPPLIERS

Name

COSEVENTCHANNELADMIN-EVENTCHANNEL-FOR-SUPPLIERS

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY EVENTCHANNEL IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-SUPPLIERA BY OBJ.
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCHANNELADMIN-EVENTCHANNEL-FOR-SUPPLIERS" USING
    OBJ
    ENV
    RESULT.
```

Description

Acquires the object reference for the Event Channel, in order to connect the supplier to the Event Channel.

Since this method acquires area to store object references, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Parameters

OBJ

The object reference of the Event Channel to be connected.

ENV

A structure that may contain exception information.

RESULT

The object reference of the Event Channel is specified, in order to connect the supplier to the Event Channel.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.2.3 COSEVENTCHANNELADMIN-EVENTCHANNEL-DESTROY

Name

COSEVENTCHANNELADMIN-EVENTCHANNEL-DESTROY

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCHANNELADMIN-EVENTCHANNEL-DESTROY" USING
    OBJ
    ENV.
```

Description

Destroys the Event Channel specified by OBJ.

Parameters

OBJ

The object reference of the Event Channel to be destroyed.

ENV

A structure that may contain exception information.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.2.4 COSEVENTCHANNELADMIN-CONSUMERADMIN-OBTAIN-PUSH-SUPPLIER

Name

COSEVENTCHANNELADMIN-CONSUMERADMIN-OBTAIN-PUSH-SUPPLIER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY CONSUMERADMIN IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-CONSUMERA BY OBJ.
01 COPY PROXYPUSHSUPPLIER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP004 BY RESULT.
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCHANNELADMIN-CONSUMERADMIN-OBTAIN-PUSH-SUPPLIER" USING
    OBJ
    ENV
    RESULT.
```

Description

Acquires the object reference for the Event Channel, in order to connect the Push model consumer to the Event Channel.

Since this method acquires area to store object references, use CORBA-OBJECT- RELEASE to release the area as soon as it is no longer needed.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-EVENTCHANNEL-FOR-CONSUMERS.

ENV

A structure that may contain exception information.

RESULT

The object reference of the Event Channel is specified, in order to connect the Push model consumer to the Event Channel.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.2.5 COSEVENTCHANNELADMIN-CONSUMERADMIN-OBTAIN-PULL-SUPPLIER

Name

COSEVENTCHANNELADMIN-CONSUMERADMIN-OBTAIN-PULL-SUPPLIER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY CONSUMERADMIN IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-CONSUMERA BY OBJ.
01 COPY PROXYPULLSUPPLIER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP002 BY RESULT.
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCHANNELADMIN-CONSUMERADMIN-OBTAIN-PULL-SUPPLIER" USING
    OBJ
    ENV
    RESULT.
```

Description

Acquires the object reference for the Event Channel, in order to connect the Pull model consumer to the Event Channel.

Since this method acquires area to store object references, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-EVENTCHANNEL-FOR-CONSUMERS.

ENV

A structure that may contain exception information.

RESULT

The object reference of the Event Channel is specified, in order to connect the Pull model consumer to the Event Channel.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.2.6 COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PUSH-CONSUMER

Name

COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PUSH-CONSUMER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY SUPPLIERADMIN IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-SUPPLIERA BY OBJ.
01 COPY PROXYPUSHCONSUMER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP003 BY RESULT.
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PUSH-CONSUMER" USING
    OBJ
    ENV
    RESULT.
```

Description

Acquires the object reference for the Event Channel, in order to connect the Push model supplier to the Event Channel.

Since this method acquires area to store object references, use CORBA-OBJECT- RELEASE to release the area as soon as it is no longer needed.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-EVENTCHANNEL-FOR-SUPPLIERS.

ENV

A structure that may contain exception information.

RESULT

The object reference of the Event Channel is specified, in order to connect the Push model supplier to the Event Channel.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.2.7 COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PULL-CONSUMER

Name

COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PULL-CONSUMER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY SUPPLIERADMIN IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-SUPPLIERA BY OBJ.
01 COPY PROXYPULLCONSUMER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP001 BY RESULT.
COPY CONST IN CORBA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PULL-CONSUMER" USING
    OBJ
    ENV
    RESULT.
```

Description

Acquires the object reference for the Event Channel, in order to connect the Pull model supplier to the Event Channel.

Since this method acquires area to store object references, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-EVENTCHANNEL-FOR-SUPPLIERS.

ENV

A structure that may contain exception information.

RESULT

The object reference of the Event Channel is specified, in order to connect the Pull model supplier to the Event Channel.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.2.8 COSEVENTCHANNELADMIN-PROXYPUSHCONSUMER-CONNECT-PUSH-SUPPLIER

Name

COSEVENTCHANNELADMIN-PROXYPUSHCONSUMER-CONNECT-PUSH- SUPPLIER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY PROXYPUSHCONSUMER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP003 BY OBJ.
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PUSHSUPPLIER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCHANNELADMIN-PROXYPUSHCONSUMER-CONNECT-PUSH-SUPPLIER" USING
    OBJ
```

```
    PUSHSUPPLIER
    ENV.
```

Description

Connects the Push model supplier to the Event Channel.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PUSH-CONSUMER.

PUSHSUPPLIER

The object reference of the supplier itself.

If a DISCONNECT notice is not required when the Event Channel is terminated, set CORBA-OBJECT-NIL in PUSHSUPPLIER.

ENV

A structure that may contain exception information.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSEVENTCHANNELADMIN-AL0001

The Event Channel is already connected.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Note

To reconnect to the Event Channel, start again from COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PUSH-CONSUMER.

4.19.2.9 COSEVENTCHANNELADMIN-PROXYPULLSUPPLIER-CONNECT-PULL-CONSUMER

Name

COSEVENTCHANNELADMIN-PROXYPULLSUPPLIER-CONNECT-PULL-CONSUMER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY PROXYPULLSUPPLIER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP002 BY OBJ.
COPY CONST IN CORBA.
```

```
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PULLCONSUMER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCHANNELADMIN-PROXYPULLSUPPLIER-CONNECT-PULL-CONSUMER" USING
    OBJ
    PULLCONSUMER
    ENV.
```

Description

Connects the Pull model consumer to the Event Channel.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-CONSUMERADMIN-OBTAIN-PULL-SUPPLIER.

PULLCONSUMER

The object reference of the consumer itself.

If a DISCONNECT notice is not required when the Event Channel is terminated, set CORBA-OBJECT-NIL in PULLCONSUMER.

ENV

A structure that may contain exception information.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSEVENTCHANNELADMIN-AL0001

The Event Channel is already connected.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Note

To reconnect to the Event Channel, start again from COSEVENTCHANNELADMIN-CONSUMERADMIN-OBTAIN-PULL-SUPPLIER.

4.19.2.10 COSEVENTCHANNELADMIN-PROXYPULLCONSUMER-CONNECT-PULL-SUPPLIER

Name

COSEVENTCHANNELADMIN-PROXYPULLCONSUMER-CONNECT-PULL-SUPPLIER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY PROXYPULLCONSUMER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP001 BY OBJ.
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PULLSUPPLIER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCHANNELADMIN-PROXYPULLCONSUMER-CONNECT-PULL-SUPPLIER" USING
    OBJ
    PULLSUPPLIER
    ENV.
```

Description

Connects the Pull model supplier to the Event Channel.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PULL-CONSUMER.

PULLSUPPLIER

The object reference of the supplier itself.

ENV

A structure that may contain exception information.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detail information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSEVENTCHANNELADMIN-AL0001

The Event Channel is already connected.

EX-COSEVENTCHANNELADMIN-TYPEER

The specified object type is incorrect.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Note

To reconnect to the Event Channel, start again from COSEVENTCHANNELADMIN-SUPPLIERADMIN-OBTAIN-PULL-CONSUMER.

4.19.2.11 COSEVENTCHANNELADMIN-PROXYPUSHSUPPLIER-CONNECT-PUSH-CONSUMER

Name

COSEVENTCHANNELADMIN-PROXYPUSHSUPPLIER-CONNECT-PUSH- CONSUMER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY PROXYPUSHSUPPLIER IN COSEVENTCOMM REPLACING
COSEVENTCHANNELADMIN-PROXYP004 BY OBJ.
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PUSHCONSUMER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCHANNELADMIN-PROXYPUSHSUPPLIER-CONNECT-PUSH-CONSUMER" USING
    OBJ
    PUSHCONSUMER
    ENV.
```

Description

Connects Push model consumer to the Event Channel.

Parameters

OBJ

The object reference returned by COSEVENTCHANNELADMIN-CONSUMERADMIN-OBTAIN-PUSH-SUPPLIER.

PULLSUPPLIER

The object reference of the consumer itself.

ENV

A structure that may contain exception information.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure. For abnormal termination, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detail information is set in ID in the ENV structure.

In the event of a user exception, the following detail information is specified.

EX-COSEVENTCHANNELADMIN-AL0001

The Event Channel already connected.

EX-COSEVENTCHANNELADMIN-TYPEER

The specified object type is incorrect.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Note

To reconnect to the Event Channel, start again from COSEVENTCHANNELADMIN-CONSUMERADMIN-OBTAIN-PUSH-SUPPLIER.

4.19.2.12 Interfaces Available to Inheritance

The following interfaces can be used by inheritance. For details, refer to "4.19.1 COSEVENTCOMM Interface".

1. COSEVENTCHANNELADMIN-PROXYPUSHCONSUMER-PUSH
2. COSEVENTCHANNELADMIN-PROXYPUSHCONSUMER-DISCONNECT-PUSH- CONSUMER
3. COSEVENTCHANNELADMIN-PROXYPUSHSUPPLIER-DISCONNECT-PUSH-SUPPLIER
4. COSEVENTCHANNELADMIN-PROXYPULLSUPPLIER-PULL
5. COSEVENTCHANNELADMIN-PROXYPULLSUPPLIER-TRY-PULL
6. COSEVENTCHANNELADMIN-PROXYPULLSUPPLIER-DISCONNECT-PULL-SUPPLIER
7. COSEVENTCHANNELADMIN-PROXYPULLCONSUMER-DISCONNECT-PULL-CONSUMER

4.19.3 EVENTFACTORY Interface

This section describes the EVENTFACTORY interface.

4.19.3.1 EVENTFACTORY-CREATE

Name

EVENTFACTORY-CREATE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.

```
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
01 COPY EVENTFACTORY IN COSEVENTCOMM REPLACING EVENTFACTORY BY OBJECT.
01 COPY EVENTCHANNEL IN COSEVENTCOMM REPLACING COSEVENTCHANNELADMIN-EVENT0001
BY RESULT.
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY RESULT.
01 COPY STRING IN CORBA REPLACING CORBA-STRING BY CHKEY.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "EVENTFACTORY-CREATE" USING
    OBJ
    CHKEY
    EVDATA
    ENV.
    RESULT
EVENTFACTORY-OPTION.
02 MAX-QUEUING    PIC 9(9) COMP-5.
02 LIFE-TIME      PIC 9(9) COMP-5.
02 MODEL PIC 9(9) COMP-5.
88 EVENTFACTORY-MODELANY VALUE 0.
88 EVENTFACTORY-MODELPUSH VALUE 1.
88 EVENTFACTORY-MODELPULL VALUE 2.
88 EVENTFACTORY-MODELMIXED VALUE 3.
```

Description

This operation generates the Event Channel and returns the object reference of the generated Event Channel.

Since this method acquires area to store object references, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Parameters

OBJ

The object reference specifies and fetches the "EVENTFACTORY-OBJECTID-FACTORY" in the IDENTIFIER parameter in *CORBA-ORB-RESOLVE-INITIAL-REFERENCES*.

CHKEY

The common keyword for consumer and supplier (maximum 64 characters).

When the common keyword is specified, an object reference indicating the same Event Channel is returned.

EVDATA

Specifies the EVENTFACTORY-OPTION structure.

Specify the values shown in the following table for each member of the EVENTFACTORY-OPTION structure.

Table 4.1 EVENTFACTORY-OPTION Structure Members

Member	Set value
MAX-QUEUING	When you set EVENTFACTORY-ES-DEFAULT-VALUE, it uses the "maximum value of event data number which can be stored in the Event Channel" set in the Event Service configuration information setting.
LIFE-TIME	Data holding time (seconds). When you set EVENTFACTORY-ES-DEFAULT-VALUE, it uses the value of "existing time (sec.) of the stored event data" set in the Event Service configuration information setting. If 0 is set, then timeout monitoring is not implemented.
MODEL	Specifies the following connection models: EVENTFACTORY-MODELANY.....Determined when connected EVENTFACTORY-MODELPUSH.....Push model EVENTFACTPRY-MODELPULL.....Pull model EVENTFACTORY-MODELMIXED.....Mixed model

ENV

A structure that may contain exception information.

RESULT

The object reference of the generated Event Channel is specified.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.19.3.2 EVENTFACTORY-CREATE-CHANNEL

Name

EVENTFACTORY-CREATE-CHANNEL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY EVENTFACTORY IN CORBA REPLACING EVENTFACTORY BY OBJ.
01 COPY STRING IN CORBA REPLACING CORBA-STRING BY KEY.
01 COPY EVENTFACTORY-OPTION IN COSEVENTCOMM REPLACING EVENTFACTORY-OPTION BY DATA.
01 COPY EVENTFACTORY-EVENTPROPERTY IN COSEVENTCOMM REPLACING EVENTFACTORY-EVENTPROPERTY  BY PROPERTY.
01 COPY BOOLEAN IN CORBA REPLACING CORBA-BOOLEAN CREATE.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY EVENTCHNL.
```

PROCEDURE DIVISION.

```
CALL "EVENTFACTORY-CREATE-CHANNEL" USING
          OBJ
          KEY
          DATA
          PROPERTY
          CREATE
          ENV
          EVENTCHNL.

EVENTFACTORY-OPTION.
  02 MAX-QUEUING    PIC 9(9) COMP-5.
  02 LIFE-TIME         PIC 9(9) COMP-5.
  02 MODEL PIC 9(9) COMP-5.
    88 EVENTFACTORY-MODELANY VALUE 0.
    88 EVENTFACTORY-MODELPUSH VALUE 1.
    88 EVENTFACTORY-MODELPULL VALUE 2.
    88 EVENTFACTORY-MODELMIXED VALUE 3.
```

Description

Generates the Event Channel and returns the object reference for the generated Event Channel. This method is used to change the host name and port number for the generated Channel.

Since this method acquires area to store object references, use CORBA-OBJECT-RELEASE to release the area as soon as it is no longer needed.

Parameters

OBJ

The object reference specifies and fetches the "EVENTFACTORY-OBJECTID-FACTORY" in the IDENTIFIER parameter in *CORBA-ORB-RESOLVE-INITIAL-REFERENCES*.

KEY

The common keyword for consumer and supplier (maximum 64 characters).

When the common keyword is specified, an object reference indicating the same Event Channel is returned.

DATA

Specifies the EVENTFACTORY-OPTION structure.

The members shown in the table below are set for each member of the EVENTFACTORY-OPTION structure.

If the Event Channel has already been generated, the value for this parameter is ignored.

Table 4.2 EVENTFACTORY-OPTION Structure Members

Member	Set value
MAX-QUEUING	When you set EVENTFACTORY-ES-DEFAULT-VALUE, it uses the "maximum amount of event data that can be stored in the Event Channel" set in the Event Service configuration information setting.
LIFE-TIME	Data holding time (seconds). When you set EVENTFACTORY-ES-DEFAULT-VALUE, it uses the value of "existing time (sec.) of the stored event data" set in the Event Service configuration information setting. If 0 is set, then timeout monitoring is not implemented.
MODEL	Specifies the following connection models: EVENTFACTORY-MODELANY.....Determined when connected EVENTFACTORY-MODELPUSH.....Push model EVENTFACTPRY-MODELPULL.....Pull model EVENTFACTORY-MODELMIXED.....Mixed model

PROPERTY

Specifies the value shown in the table below. If an invalid name is specified, the corresponding record is invalid. If either of HOSTNAME and PORTNUMBER is set, the specified record is invalid.

If the Event Channel has already been generated, the value for this parameter is ignored.

Member (NAME)	Data type (VALUE)	Set value
HOSTNAME	CORBA-STRING	Specifies the host name or the IP address (maximum 64 characters),.
PORTNUMBER	CORBA-UNSIGNED-SHORT	Specifies the port number.

CREATE

If the Event Channel was generated, CORBA-TRUE is set.

If the Event Channel already exists, CORBA-FALSE is set.

ENV

A structure that may contain exception information.

EVENTCHNL

This is the Event Channel object reference specified in *KEY*.

Return Values

If this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Notes

This method cannot be used if the following host names are set, as the Event Channel will fail to start.

- The "Corba Host Name" Interstage operating environment definition

- "IIOP_hostname" in the config file (CORBA Service) (The host name used by the CORBA Service)

When specifying the port number, select one of the following host names specified for the CORBA Service port number:

- If SSL communication is enabled
  - The "SSL Port Number" Interstage operating environment definition
  - "UNO_IIOP_ssl_port" in the config file (CORBA Service)
- If SSL communication is disabled
  - The "Corba Port Number" Interstage operating environment definition
  - "IIOP_port" in the config file (CORBA Service)

4.19.3.3 EVENTFACTORY-GET-EVENT-CHANNEL

Name

EVENTFACTORY-GET-EVENT-CHANNEL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY EVENTFACTORY IN COSEVENTCOMM REPLACING EVENTFACTORY BY OBJ.
01 COPY STRING IN CORBA REPLACING CORBA-STRING BY KEY.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY EVENTCHNL.
```

PROCEDURE DIVISION.

```
CALL "EVENTFACTORY-GET-EVENT-CHANNEL" USING
          OBJ
          KEY
          ENV
          EVENTCHNL.
```

Description

Acquires the Event Channel object reference specified in *KEY*.

Parameters

OBJ

The object reference specifies and fetches the "EVENTFACTORY-OBJECTID-FACTORY" in the IDENTIFIER parameter in *CORBA-ORB-RESOLVE-INITIAL-REFERENCES*.

KEY

The acquired Event Channel name

ENV

A structure that may contain exception information.

EVENTCHNL

This is the Event Channel object reference specified in *KEY*.

Return Values

If this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-EVENTFACTORY-CHANNELNOTFOUN

The specified channel could not be found.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20 Notification Service Interface

Type definitions

The Admin property item that generates the Event Channel:

```
01 COSNOTIFICATION-ADMINPROPER001.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-MAXIMUM.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-LENGTH.
  02 SEQ-BUFFER USAGE IS POINTER.
```

Event Channel ID:

```
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY COSNOTIFYCHANNELADMIN-CHANNELI.
```

Event Channel ID sequence type:

```
01 COSNOTIFYCHANNELADMIN-CHANN001.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-MAXIMUM.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-LENGTH.
  02 SEQ-BUFFER USAGE IS POINTER.
```

Client type:

```
01 COPY ENUM IN CORBA REPLACING CORBA-ENUM BY COSNOTIFYCHANNELADMIN-CLIENTTY.
  88 COSNOTIFYCHANNELADMIN-ANY-EVEN VALUE 0.
  88 COSNOTIFYCHANNELADMIN-STRUCTUR VALUE 1.
```

Property structure:

```
01 COSNOTIFICATION-PROPERTY.
  02 NAME USAGE IS POINTER.
  02 COPY ANY IN CORBA REPLACING CORBA-ANY BY IDL-VALUE.
```

Proxy ID:

```
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY COSNOTIFYCHANNELADMIN-PROXYID.
```

Proxy type:

```
01 COPY ENUM IN CORBA REPLACING CORBA-ENUM BY COSNOTIFYCHANNELADMIN-PROXYTYP.
  88 COSNOTIFYCHANNELADMIN-PUSH-ANY VALUE 0.
  88 COSNOTIFYCHANNELADMIN-PULL-ANY VALUE 1.
  88 COSNOTIFYCHANNELADMIN-PUSH-STR VALUE 2.
  88 COSNOTIFYCHANNELADMIN-PULL-STR VALUE 3.
```

QosProperty structure sequence type:

```
01 COSNOTIFICATION-QOSPROPERTIES.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-MAXIMUM.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-LENGTH.
  02 SEQ-BUFFER USAGE IS POINTER.
```

STRUCTUREDEVENT structure:

```
01 COSNOTIFICATION-STRUCTUREDEVEN.
  02 HEADER.
    03 FIXED-HEADER.
      04 EVENT-TYPE.
        05 DOMAIN-NAME USAGE IS POINTER.
        05 TYPE-NAME USAGE IS POINTER.
      04 EVENT-NAME USAGE IS POINTER.
    03 VARIABLE-HEADER.
      04 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-MAXIMUM.
      04 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-LENGTH.
      04 SEQ-BUFFER USAGE IS POINTER.
  02 FILTERABLE-DATA.
    03 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-MAXIMUM.
    03 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-LENGTH.
    03 SEQ-BUFFER USAGE IS POINTER.
  02 COPY ANY IN CORBA REPLACING CORBA-ANY BY REMAINDER-OF-BODY.
```

PROPERTYERROR structure:

```
01 COSNOTIFICATION-PROPERTYERROR.
  02 COPY ENUM IN CORBA REPLACING CORBA-ENUM BY IDL-CODE.
    88 COSNOTIFICATION-UNSUPPORTED-PR VALUE 0.
    88 COSNOTIFICATION-UNAVAILABLE-PR VALUE 1.
    88 COSNOTIFICATION-UNSUPPORTED-VA VALUE 2.
    88 COSNOTIFICATION-UNAVAILABLE-VA VALUE 3.
    88 COSNOTIFICATION-BAD-PROPERTY VALUE 4.
    88 COSNOTIFICATION-BAD-TYPE VALUE 5.
    88 COSNOTIFICATION-BAD-VALUE VALUE 6.
  02 NAME USAGE IS POINTER.
  02 AVAILABLE-RANGE.
    03 COPY ANY IN CORBA REPLACING CORBA-ANY BY LOW-VAL.
    03 COPY ANY IN CORBA REPLACING CORBA-ANY BY HIGH-VAL.
```

PROPERTYERROR structure sequence type:

```
01 COSNOTIFICATION-PROPERTYERRORS.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-MAXIMUM.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-LENGTH.
  02 SEQ-BUFFER USAGE IS POINTER.
```

Object reference for the ConsumerAdmin object:

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSNOTIFYCHANNELADMIN-CONSUMER.
```

Event Channel object reference:

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSNOTIFYCHANNELADMIN-EVENTCHA.
```

EventFactory object reference:

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSNOTIFYCHANNELADMIN-EVENT001.
```

ProxyConsumer object reference:

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSNOTIFYCHANNELADMIN-PROXYCON.
```

ProxySupplier object reference:

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSNOTIFYCHANNELADMIN-PROXYPUL.
```

ProxyConsumer object reference:

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSNOTIFYCHANNELADMIN-PROXYSUP.
```

* ProxySupplier object reference:

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSNOTIFYCHANNELADMIN-PROXYSUP.
```

StructuredEvent type supplier object reference:

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSNOTIFYCHANNELADMIN-STRUC001.
```

StructuredEvent type consumer object reference:

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSNOTIFYCHANNELADMIN-STRUCTUR.
```

Object reference for SupplierAdmin object

```
COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY COSNOTIFYCHANNELADMIN-SUPPLIER.
```

4.20.1 QOSADMIN Interface

This section details the functions associated with the QOSADMIN Interface.

4.20.1.1 COSNOTIFICATION-QOSADMIN-GET-QOS

Name

COSNOTIFICATION-QOSADMIN-GET-QOS

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 QOS USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFICATION-QOSADMIN-GET-QOS" USING
          OBJ
          ENV
          QOS.
```

Description

Fetches QoSProperties.

This method secures an area for QoSProperties. Use CORBA-FREE to release the area when it is no longer required.

Parameters

OBJ

The Event Channel object reference.

ENV

A structure that may contain exception information.

QOS

QoSProperties.

Return Values

If this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.1.2 COSNOTIFICATION-QOSADMIN-SET-QOS

Name

COSNOTIFICATION-QOSADMIN-SET-QOS

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY OBJ.
01 COPY QOSPROPERTIES IN COSEVENTCOMM REPLACING COSNOTIFICATION-QOSPROPERTIES BY QOS.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFICATION-QOSADMIN-SET-QOS" USING
          OBJ
          QOS
          ENV.
```

Description

Sets QoSProperties.

Parameters

OBJ

The Event Channel object reference.

QOS

The set QoSProperties.

ENV

A structure that may contain exception information.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detail information is set in ID in the ENV structure.

In the event of a user exception, the following detail information is specified.

EX-COSNOTIFICATION-UNSUPPORTEDQOS

There is an error in the QoSProperties item or value.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.2 COSNOTIFYCOMM Interface

This section details the functions associated with the COSNOTIFYCOMM Interface.

4.20.2.1 COSNOTIFYCOMM-STRUCTUREDPUSHCONSUMER-PUSH-STRUCTURED-EVENT

Name

COSNOTIFYCOMM-STRUCTUREDPUSHCONSUMER-PUSH-STRUCTURED-EVENT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYSTRUCTUREDPROXYPUSHCON IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-STRUCTUR BY OBJ.
01 COPY STRUCTUREDEVENT IN COSEVENTCOMM REPLACING COSNOTIFICATION-STRUCTUREDEVEN BY EVDATA.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCOMM-STRUCTUREDPUSHCONSUMER-PUSH-STRUCTURED-EVENT" USING
          OBJ
          EVDATA
          ENV.
```

Description

Sends to the consumer the STRUCTUREDEVENT type event data specified in *EVDATA*.

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER*.

EVDATA

The STRUCTUREDEVENT type event data sended to consumer.

ENV

A structure that may contain exception information.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detail information is set in ID in the ENV structure.

In the event of a user exception, the following detail information is specified.

EX-COSEVENTCOMM-DISCONNECTED

The Event Channel was not connected.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the *esmkchnl* command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.2.2 COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-PULL-STRUCTURED-EVENT

Name

COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-PULL-STRUCTURED-EVENT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYSTRUCTUREDPROXYPULLSUP IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-STRUC001 BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 RECV                 USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-PULL-STRUCTURED-EVENT" USING
          OBJ
          ENV
          RECV.
```

Description

Requests STRUCTUREDEVENT type event data from the supplier. Event data can be fetched or it may be blocked until an exception occurs. Use COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-TRY-PULL-STRUCTURED-EVENT to return immediately if event data is not fetched.

Use CORBA-FREE to release the area secured for the STRUCTUREDEVENT type data when the area is no longer required.

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER*.

ENV

A structure that may contain exception information.

RECV

This is the event data from the supplier.

Return Values

If this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSEVENTCOMM-DISCONNECTED

The Event Channel was not connected.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the *esmkchnl* command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.2.3 COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-TRY-PULL-STRUCTURED-EVENT

Name

COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-TRY-PULL-STRUCTURED-EVENT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYSTRUCTUREDPROXYPULLSUP IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-STRUC001 BY OBJ.
01 COPY BOOLEAN IN CORBA REPLACING CORBA-BOOLEAN BY HAS-EVENT.
```

```
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 RECV                 USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-TRY-PULL-STRUCTURED-EVENT" USING
          OBJ
          HAS-EVENT
          ENV
          RECV.
```

Description

Requests STRUCTUREDEVENT type event data from the supplier. Immediately returns if event data is not fetched. Use COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-PULL-STRUCTURED-EVENT to block until event data is fetched.

Use CORBA-FREE to release the area secured for the STRUCTUREDEVENT type data when the area is no longer required.

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER*.

HAS-EVENT (out parameter)

When event data is fetched, CORBA-TRUE is set.

When event data is not fetched, CORBA-FALSE is set.

ENV

A structure that may contain exception information.

RECV

This is the event data from the supplier.

Return Values

If this function returns normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure. When event data is fetched, CORBA-TRUE is set in *HAS-EVENT*, and the event data from the supplier is returned. When event data is not fetched, CORBA-FALSE is set in *HAS-EVENT*. In this case, because an area is secured for the STRUCTUREDEVENT type event data, use CORBA-FREE to release the area.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSEVENTCOMM-DISCONNECTED

The Event Channel was not connected.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the *esmkchnl* command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Notes

If CORBA-FALSE is set to *HAS-EVENT* for a local transaction, a cancellation need not be sent using ES-CHANNELUTIL-LOCAL-ROLLBACK.

When the local transaction has completed, the channel is notified using ES-CHANNELUTIL-LOCAL-COMMIT.

Windows32 Solaris32

If CORBA-FALSE is set to *HAS-EVENT* for a global transaction, a cancellation need not be sent using COSTRANSACTIONS-CURRENT-ROLLBACK.

When the global transaction has completed, the channel is notified using COSTRANSACTIONS-CURRENT-COMMIT.

4.20.2.4 COSNOTIFYCOMM-STRUCTUREDPUSHCONSUMER-DISCONNECT-STRUCTURED-PUSH-CONSUMER

Name

COSNOTIFYCOMM-STRUCTUREDPUSHCONSUMER-DISCONNECT-STRUCTURED-PUSH-CONSUMER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYSTRUCTUREDPROXYPUSHCON IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-STRUCTUR BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCOMM-STRUCTUREDPUSHCONSUMER-DISCONNECT-STRUCTURED-PUSH-CONSUMER" USING
          OBJ
          ENV
```

Description

Declares the end of event transmission from the supplier.

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER*.

ENV

A structure that may contain exception information.

Return Values

If this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.2.5 COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-DISCONNECT-STRUCTURED-PULL-SUPPLIER

Name

COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-DISCONNECT-STRUCTURED-PULL-SUPPLIER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYSTRUCTUREDPROXYPULLSUP IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-STRUC001 BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-DISCONNECT-STRUCTURED-PULL-SUPPLIER" USING
          OBJ
          ENV
```

Description

Declares the end of event transmission from the consumer.

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER*.

ENV

A structure that may contain exception information.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.2.6 Inherited Interfaces

The following interfaces can be inherited and used. For details of these interfaces, refer to "4.19.1 COSEVENTCOMM Interface".

(1) COSNOTIFYCOMM-PUSHCONSUMER-PUSH

(2) COSNOTIFYCOMM-PUSHCONSUMER-DISCONNECT-PUSH-CONSUMER

(3) COSNOTIFYCOMM-PULLSUPPLIER-PULL

(4) COSNOTIFYCOMM-PULLSUPPLIER-TRY-PULL

(5) COSNOTIFYCOMM-PULLSUPPLIER-DISCONNECT-PULL-SUPPLIER

4.20.3 COSNOTIFYCHANNELADMIN Interface

This section details the functions associated with the COSNOTIFYCHANNELADMIN Interface.

4.20.3.1 COSNOTIFYCHANNELADMIN-EVENTCHANNEL--GET-DEFAULT-CONSUMER-ADMIN

Name

COSNOTIFYCHANNELADMIN-EVENTCHANNEL--GET-DEFAULT-CONSUMER-ADMIN

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYEVENTCHANNEL IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-EVENTCHA BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY NOTIFYCONSUMERADMIN IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-CONSUMER BY CONADMIN.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-EVENTCHANNEL--GET-DEFAULT-CONSUMER-ADMIN" USING
          OBJ
          ENV
          CONADMIN.
```

Description

Returns the object reference of the CONSUMERADMIN object that is standard for the Event Channel.

This method secures an area for holding the object references. Use CORBA-OBJECT-RELEASE to release the area when it is no longer required.

Parameters

OBJ

The object reference for the Event Channel connected.

ENV

A structure that may contain exception information.

CONADMIN

This is the object reference for the CONSUMERADMIN object held by the Event Channel as standard.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.3.2 COSNOTIFYCHANNELADMIN-EVENTCHANNEL--GET-DEFAULT-SUPPLIER-ADMIN

Name

COSNOTIFYCHANNELADMIN-EVENTCHANNEL--GET-DEFAULT-SUPPLIER-ADMIN

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYEVENTCHANNEL IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-EVENTCHA BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY NOTIFYSUPPLIERADMIN IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-SUPPLIER BY SUPADMIN.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-EVENTCHANNEL--GET-DEFAULT-SUPPLIER-ADMIN" USING
          OBJ
          ENV
          SUPADMIN.
```

Description

Returns the object reference of the SUPPLIERADMIN object that is standard for the Event Channel.

This method secures an area for holding the object references. Use CORBA-OBJECT-RELEASE to release the area when it is no longer required.

Parameters

OBJ

The object reference for the Event Channel connected.

ENV

A structure that may contain exception information.

SUPADMIN

This is the object reference for the SUPPLIERADMIN object held by the Event Channel as standard.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.3.3 COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER

Name

COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYCONSUMERADMIN IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-CONSUMER BY OBJ.
01 COPY CLIENTTYPE IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-CLIENTTY BY CTYPE.
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY PROXY-ID.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY NOTIFYPROXYPULLSUP IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-PROXYPUL BY PROXYPULLSUP.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER" USING
          OBJ
          CTYPE
          PROXY-ID
          ENV
          PROXYPULLSUP.
```

Description

Creates a PROXYSUPPLIER object of the client type specified in *CTYPE*.

This method secures an area for holding the object references. Use CORBA-OBJECT-RELEASE to release the area when it is no longer required.

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-EVENTCHANNEL--GET-DEFAULT-CONSUMER-ADMIN*.

CTYPE

The client type by which the PROXYSUPPLIER object is created is specified as follows:

COSNOTIFYCHANNELADMIN-ANY-EVEN: Handles ANYtype event

COSNOTIFYCHANNELADMIN-STRUCTUR: Handles STRUCTUREDEVENT type event

PROXY-ID (out parameter)

The created PROXYSUPPLIER ID is set.

ENV

A structure that may contain exception information.

PROXYPULLSUP

The object reference for the PROXYSUPPLIER object is set.

Return Values

If this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detail information is specified.

EX-COSNOTIFYCHANNELADMIN-AD001

The upper limit for creating PROXYSUPPLIER has been reached - no more can be created.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.3.4 COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER

Name

COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYSUPPLIERADMIN IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-SUPPLIER BY OBJ.
01 COPY CLIENTTYPE IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-CLIENTTY BY CTYPE.
01 COPY LONG IN CORBA REPLACING CORBA-LONG BY PROXY-ID.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY NOTIFYPROXYPUSHCON IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-PROXYPUS BY PROXYPUSHCON.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER" USING
          OBJ
          CTYPE
          PROXY-ID
          ENV
          PROXYPUSHCON
```

Description

Creates a PROXYCONSUMER object of the client type specified in *CTYPE*.

This method secures an area for holding the object references. Use CORBA-OBJECT-RELEASE to release the area when it is no longer required.

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-EVENTCHANNEL--GET-DEFAULT-SUPPLIER-ADMIN*.

CTYPE

The client type by which the PROXYCONSUMER object is created is specified as follows:

COSNOTIFYCHANNELADMIN-ANY-EVEN: Handles ANY type event

COSNOTIFYCHANNELADMIN-STRUCTUR: Handles STRUCTUREDEVENT type event

PROXY-ID (out parameter)

The created PROXYCONSUMER ID is set.

ENV

A structure that may contain exception information.

PROXYPUSHCON

This is the object reference for the PROXYCONSUMER object.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSNOTIFYCHANNELADMIN-AD001

The upper limit for creating PROXYSUPPLIER has been reached - no more can be created.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.3.5 COSNOTIFYCHANNELADMIN-CONSUMERADMIN--GET-MYCHANNEL

Name

COSNOTIFYCHANNELADMIN-CONSUMERADMIN--GET-MYCHANNEL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYCONSUMERADMIN IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-CONSUMER OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY NOTIFYEVENTCHANNEL IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-EVENTCHA BY ADMIN.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-CONSUMERADMIN--GET-MYCHANNEL" USING
          OBJ
          ENV
          ADMIN
```

Description

Returns the object reference for the Event Channel that generated CONSUMERADMIN.

This method secures an area for holding the object references. Use CORBA-OBJECT-RELEASE to release the area when it is no longer required.

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-EVENTCHANNEL--GET-DEFAULT-CONSUMER-ADMIN*.

ENV

A structure that may contain exception information.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.3.6 COSNOTIFYCHANNELADMIN-SUPPLIERADMIN--GET-MYCHANNEL

Name

COSNOTIFYCHANNELADMIN-SUPPLIERADMIN--GET-MYCHANNEL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYSUPPLIERADMIN IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-SUPPLIER OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY NOTIFYEVENTCHANNEL IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-EVENTCHA BY ADMIN.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-SUPPLIERADMIN--GET-MYCHANNEL" USING
          OBJ
          ENV
          ADMIN
```

Description

Returns the object reference for the Event Channel that generated SUPPLIERADMIN.

This method secures an area for holding the object references. Use CORBA-OBJECT-RELEASE to release the area when it is no longer required.

Parameters

OBJ

The object reference returned by

COSNOTIFYCHANNELADMIN-EVENTCHANNEL--GET-DEFAULT-SUPPLIER-ADMIN.

ENV

A structure that may contain exception information.

ADMIN

This is the object reference for the Event Channel that generated SUPPLIERADMIN.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure, and the object reference of the Event Channel is returned.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.3.7 COSNOTIFYCHANNELADMIN-PROXYPUSHCONSUMER-CONNECT-ANY-PUSH-SUPPLIER

Name

COSNOTIFYCHANNELADMIN-PROXYPUSHCONSUMER-CONNECT-ANY-PUSH-SUPPLIER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYPROXYPUSHCON IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-PROXYPUS BY OBJ.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PUSH-SUPPLIER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-PROXYPUSHCONSUMER-CONNECT-ANY-PUSH-SUPPLIER" USING
          OBJ
          PUSH-SUPPLIER
          ENV
```

Description

Connects an ANY type supplier to the Event Channel.

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER*.

PUSH-SUPPLIER

The object reference of the application that calls this method.

Specify CORBA_OBJECT_NIL in *PUSH-SUPPLIER* if notification of disconnection is not required when the Event Channel is terminated.

ENV

A structure that may contain exception information.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSEVENTCHANNELADMIN-AL0001

The Event Channel is already connected.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Note

On reconnection to the Event Channel, start again from

COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER.

4.20.3.8 COSNOTIFYCHANNELADMIN-PROXYPULLSUPPLIER-CONNECT-ANY-PULL-CONSUMER

Name

COSNOTIFYCHANNELADMIN-PROXYPULLSUPPLIER-CONNECT-ANY-PULL-CONSUMER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYPROXYPULLSUP IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-PROXYPUL OBJ.
```

```
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PULL-CONSUMER.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-PROXYPULLSUPPLIER-CONNECT-ANY-PULL-CONSUMER" USING
          OBJ
          PULL-CONSUMER
          ENV
```

Description

Connects an *any* type consumer to the Event Channel.

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER*.

PULL-CONSUMER

The object reference of the application that calls this method.

Specify CORBA-OBJECT-NIL in *PULL-CONSUMER* if notification of disconnection is not required when the Event Channel is terminated.

ENV

A structure that may contain exception information.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSEVENTCHANNELADMIN-AL0001

The Event Channel is already connected.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Note

On reconnection to the Event Channel, start again from COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER.

4.20.3.9 COSNOTIFYCHANNELADMIN-PROXYCONSUMER--GET-MYTYPE

Name

COSNOTIFYCHANNELADMIN-PROXYCONSUMER--GET-MYTYPE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
```

```
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYPROXYPUSHCON IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-PROXYPUS BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY NOTIFYPROXYTYPE IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-PROXYTYP BY PROXYTYPE.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-PROXYCONSUMER--GET-MYTYPE" USING
          OBJ
          ENV
          PROXYTYPE
```

Description

Returns the following values for the PROXY object type:

COSNOTIFYCHANNELADMIN-PUSH-ANY: ANY type Push model

COSNOTIFYCHANNELADMIN-PULL-ANY: ANY type Pull model

COSNOTIFYCHANNELADMIN-PUSH-STR: STRUCTUREDEVENT type Push model

COSNOTIFYCHANNELADMIN-PULL-STR: STRUCTUREDEVENT type Pull model

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER*.

ENV

A structure that may contain exception information.

PROXYTYPE

This is the PROXY object type.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.3.10 COSNOTIFYCHANNELADMIN-PROXYSUPPLIER--GET-MYTYPE

Name

COSNOTIFYCHANNELADMIN-PROXYSUPPLIER--GET-MYTYPE

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYPROXYPULLSUP IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-PROXYPUL BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY NOTIFYPROXYTYPE IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-PROXYTYP BY PROXYTYPE.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-PROXYSUPPLIER--GET-MYTYPE" USING
          OBJ
          ENV
          PROXYTYPE
```

Description

Returns the following values for the PROXY object type:

COSNOTIFYCHANNELADMIN-PUSH-ANY: ANY type Push model

COSNOTIFYCHANNELADMIN-PULL-ANY: ANY type Pull model

COSNOTIFYCHANNELADMIN-PUSH-STR: STRUCTUREDEVENT type Push model

COSNOTIFYCHANNELADMIN-PULL-STR: STRUCTUREDEVENT type Pull model

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER*.

ENV

A structure that may contain exception information.

PROXYTYPE

PROXY object type

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.3.11 COSNOTIFYCHANNELADMIN-STRUCTUREDPROXYPUSHCONSUMER-CONNECT-STRUCTURED-PUSH-SUPPLIER

Name

COSNOTIFYCHANNELADMIN-STRUCTUREDPROXYPUSHCONSUMER-CONNECT-STRUCTURED-PUSH-SUPPLIER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYPROXYPUSHCON IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-PROXYPUS BY OBJ.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PUSH-SUPPLIER
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-STRUCTUREDPROXYPUSHCONSUMER-CONNECT-STRUCTURED-PUSH-SUPPLIER" USING
         OBJ
         PUSH-SUPPLIER
         ENV
```

Description

Connects to the Event Channel as a STRUCTUREDEVENT type supplier.

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER*.

PUSH-SUPPLIER

The object reference of the application that calls this method.

Specify CORBA_OBJECT_NIL in *PUSH-SUPPLIER* if notification of disconnection is not required when the Event Channel is terminated.

ENV

A structure that may contain exception information.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSEVENTCHANNELADMIN-AL0001

The Event Channel is already connected.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Note

When reconnecting to the Event Channel, start again from COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER.

4.20.3.12 COSNOTIFYCHANNELADMIN-STRUCTUREDPROXYPULLSUPPLIER-CONNECT-STRUCTURED-PULL-CONSUMER

Name

COSNOTIFYCHANNELADMIN-STRUCTUREDPROXYPULLSUPPLIER-CONNECT-STRUCTURED-PULL-CONSUMER

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYPROXYPULLSUP IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-PROXYPUL BY OBJ.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PULL-CONSUMER
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-STRUCTUREDPROXYPULLSUPPLIER-CONNECT-STRUCTURED-PULL-CONSUMER" USING
         OBJ
         PULL-CONSUMER
         ENV
```

Description

Connects to the Event Channel as a STRUCTUREDEVENT type consumer.

Parameters

OBJ

The object reference returned by *COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER*.

PULL-CONSUMER

The own object reference.

Specify CORBA_OBJECT_NIL in *PULL-CONSUMER* if notification of disconnection is not required when the Event Channel is terminated.

ENV

A structure that may contain exception information.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSEVENTCHANNELADMIN-AL0001

The Event Channel is already connected.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Note

When reconnecting to the Event Channel, start again from COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER.

4.20.3.13 Interfaces Inherited

The following interfaces can be inherited and used. For details of these interfaces, refer to "4.20.2 COSNOTIFYCOMM Interface".

(1) COSNOTIFYCHANNELADMIN-STRUCTUREDPROXYPUSHCONSUMER-PUSH-STRUCTURED-EVENT

(2) COSNOTIFYCHANNELADMIN-STRUCTUREDPROXYPULLSUPPLIER-PULL-STRUCTURED-EVENT

(3) COSNOTIFYCHANNELADMIN-STRUCTUREDPROXYPULLSUPPLIER-TRY-PULL-STRUCTURED-EVENT

(4) COSNOTIFYCHANNELADMIN-STRUCTUREDPROXYPUSHCONSUMER-DISCONNECT-STRUCTURED-PUSH-CONSUMER

(5) COSNOTIFYCHANNELADMIN-STRUCTUREDPROXYPULLSUPPLIER-DISCONNECT-STRUCTURED-PULL-SUPPLIER

(6) COSNOTIFYCHANNELADMIN-PROXYPUSHCONSUMER-PUSH

(7) COSNOTIFYCHANNELADMIN-PROXYPULLSUPPLIER-PULL

(8) COSNOTIFYCHANNELADMIN-PROXYPULLSUPPLIER-TRY-PULL

(9) COSNOTIFYCHANNELADMIN-PROXYPUSHCONSUMER-DISCONNECT-PUSH-CONSUMER

(10) COSNOTIFYCHANNELADMIN-PROXYPULLSUPPLIER-DISCONNECT-PULL-SUPPLIER

4.20.4 EVENTCHANNELFACTORY Interface

COSNOTIFYCHANNELADMIN-EVENTCHANNELFACTORY is the factory interface specified by the NOTIFICATIONSERVICE.

4.20.4.1 COSNOTIFYCHANNELADMIN-EVENTCHANNELFACTORY-CREATE-CHANNEL

Name

COSNOTIFYCHANNELADMIN-EVENTCHANNELFACTORY-CREATE-CHANNEL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYEVENTFACTORY IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-EVENT001 BY OBJ.
01 COPY QOSPROPERTIES IN COSEVENTCOMM REPLACING COSNOTIFICATION-QOSPROPERTIES BY INITIAL-QOS.
01 COPY ADMINPROPERTIES IN COSEVENTCOMM REPLACING COSNOTIFICATION-ADMINPROPER001 BY INITIAL-ADMIN.
01 COPY CHANNELID IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-CHANNELI BY CHNL-ID.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY NOTIFYEVENTCHANNEL IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-EVENTCHA BY EVCHNL.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-EVENTCHANNELFACTORY-CREATE-CHANNEL" USING
          OBJ
          INITIAL-QOS
          INITIAL-ADMIN
          CHNL-ID
          ENV
          EVCHNL
```

Description

Generates the Event Channel with the QOS property items specified in *INITIAL-QOS* and *INITIAL-ADMIN* and the ADMIN property item, and specifies the Event Channel ID in *CHNL-ID*.

This method secures an area for holding the object references. Use CORBA-OBJECT-RELEASE to release the area when it is no longer required.

Parameters

OBJ

The object reference specifies and fetches the "NotificationService" in the *IDENTIFIER* parameter in *CORBA-ORB-RESOLVE-INITIAL-REFERENCES*.

INITIAL-QOS

The QOS property item which generates the Event Channel.

INITIAL-ADMIN

The ADMIN property item which generates the Event Channel.

CHNL-ID (out parameter)

The Event Channel ID is set.

ENV

A structure that may contain exception information.

EVCHNL

This is the object reference for the Event Channel that holds the QOS and ADMIN property items.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSNOTIFICATION-UNSUPPOR001

There is an error in the specified QOS property item or value.

EX-COSNOTIFICATION-UNSUPPOR002

There is an error in the specified ADMIN property item or value.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.4.2 COSNOTIFYCHANNELADMIN-EVENTCHANNELFACTORY-GET-ALL-CHANNELS

Name

COSNOTIFYCHANNELADMIN-EVENTCHANNELFACTORY-GET-ALL-CHANNELS

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYEVENTFACTORY IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-EVENT001 BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 IDSEQ USAGE IS POINTER.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-EVENTCHANNELFACTORY-GET-ALL-CHANNELS" USING
          OBJ
          ENV
          IDSEQ
```

Description

Returns as sequence types the IDs for all event channels managed by the Event Factory.

This method secures sequence type areas for holding the object references. Use CORBA-FREE to release the areas when they are no longer required.

Parameters

OBJ

The object reference specifies and fetches the "NotificationService" in the *IDENTIFIER* parameter in *CORBA-ORB-RESOLVE-INITIAL-REFERENCES*.

ENV

A structure that may contain exception information.

IDSEQ

This is the sequence type for storing event channels IDs managed by the Event Factory.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.20.4.3 COSNOTIFYCHANNELADMIN-EVENTCHANNELFACTORY-GET-EVENT-CHANNEL

Name

COSNOTIFYCHANNELADMIN-EVENTCHANNELFACTORY-GET-EVENT-CHANNEL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY NOTIFYEVENTFACTORY IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-EVENT001 BY OBJ.
01 COPY CHANNELID IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-CHANNELI BY CHNLID
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY NOTIFYEVENTCHANNEL IN COSEVENTCOMM REPLACING COSNOTIFYCHANNELADMIN-EVENTCHA BY EVCHNL.
```

PROCEDURE DIVISION.

```
CALL "COSNOTIFYCHANNELADMIN-EVENTCHANNELFACTORY-GET-EVENT-CHANNEL" USING
          OBJ
          CHNLID
          ENV
          EVCHNL
```

Description

Returns the object references for event channels with the IDs specified in *CHNLID*.

This method secures areas for holding the object references. Use CORBA-OBJECT-RELEASE to release the area when it is no longer required.

Parameters

OBJ

The object reference specifies and fetches the "NotificationService" in the *IDENTIFIER* parameter in *CORBA-ORB-RESOLVE-INITIAL-REFERENCES*.

CHNLID

The acquired Event Channel ID.

ENV

A structure that may contain exception information.

EVCHNL

This is the object reference for the Event Channel that holds the ID specified in *CHNLID*.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSNOTIFYCHANNELADMIN-CHANN

The specified channel could not be found.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.21 Connection Information Collection Function Interface

This interface is used to collect connection information for the Event Channel.

Type definitions

Consumer/Supplier connection information (IPv6) structure:

```
01 ES-CHANNELUTIL-PROXYDATA6.
  02 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PROXY.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY IDL-TIME.
  02 IPADDRESS.
    03 FILLER OCCURS 16.
      04 COPY OCTET IN CORBA REPLACING CORBA-OCTET BY IPADDRESS-V.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY IP-FORMAT.
  02 COPY ENUM IN CORBA REPLACING CORBA-ENUM BY KIND.
    88 ES-CHANNELUTIL-ALL-PROXYS VALUE 0.
    88 ES-CHANNELUTIL-PROXY-CONSUMER VALUE 1.
    88 ES-CHANNELUTIL-PROXY-SUPPLIER VALUE 2.
    88 ES-CHANNELUTIL-PROXY-PULL-CONS VALUE 3.
    88 ES-CHANNELUTIL-PROXY-PULL-SUPP VALUE 4.
    88 ES-CHANNELUTIL-PROXY-PUSH-CONS VALUE 5.
    88 ES-CHANNELUTIL-PROXY-PUSH-SUPP VALUE 6.
    88 ES-CHANNELUTIL-PROXY-PULL-C001 VALUE 7.
    88 ES-CHANNELUTIL-PROXY-PULL-S001 VALUE 8.
    88 ES-CHANNELUTIL-PROXY-PUSH-C001 VALUE 9.
    88 ES-CHANNELUTIL-PROXY-PUSH-S001 VALUE 10.
    88 ES-CHANNELUTIL-STRUCTURED-PROX VALUE 11.
    88 ES-CHANNELUTIL-STRUCTURED-P001 VALUE 12.
    88 ES-CHANNELUTIL-STRUCTURED-P002 VALUE 13.
    88 ES-CHANNELUTIL-STRUCTURED-P003 VALUE 14.
```

Consumer/Supplier connection information (IPv4) structure:

```
01 ES-CHANNELUTIL-PROXYDATA.
  02 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PROXY.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY IDL-TIME.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY IPADDRESS.
  02 COPY ENUM IN CORBA REPLACING CORBA-ENUM BY KIND.
    88 ES-CHANNELUTIL-ALL-PROXYS VALUE 0.
    88 ES-CHANNELUTIL-PROXY-CONSUMER VALUE 1.
    88 ES-CHANNELUTIL-PROXY-SUPPLIER VALUE 2.
    88 ES-CHANNELUTIL-PROXY-PULL-CONS VALUE 3.
    88 ES-CHANNELUTIL-PROXY-PULL-SUPP VALUE 4.
    88 ES-CHANNELUTIL-PROXY-PUSH-CONS VALUE 5.
    88 ES-CHANNELUTIL-PROXY-PUSH-SUPP VALUE 6.
    88 ES-CHANNELUTIL-PROXY-PULL-C001 VALUE 7.
    88 ES-CHANNELUTIL-PROXY-PULL-S001 VALUE 8.
    88 ES-CHANNELUTIL-PROXY-PUSH-C001 VALUE 9.
    88 ES-CHANNELUTIL-PROXY-PUSH-S001 VALUE 10.
    88 ES-CHANNELUTIL-STRUCTURED-PROX VALUE 11.
    88 ES-CHANNELUTIL-STRUCTURED-P001 VALUE 12.
    88 ES-CHANNELUTIL-STRUCTURED-P002 VALUE 13.
    88 ES-CHANNELUTIL-STRUCTURED-P003 VALUE 14.
```

Consumer/Supplier connection information sequence type:

```
01 COSNOTIFICATION-QOSPROPERTIES.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-MAXIMUM.
  02 COPY LONG IN CORBA REPLACING CORBA-LONG BY SEQ-LENGTH.
  02 SEQ-BUFFER USAGE IS POINTER.
```

Interface type:

```
01 COPY ENUM IN CORBA REPLACING CORBA-ENUM BY COSEVENTCHANNELADMIN-EVENTC001.
  88 COSEVENTCHANNELADMIN-EVENTCHAN VALUE 0.
```

PROXY object type:

```
01 COPY ENUM IN CORBA REPLACING CORBA-ENUM BY ES-CHANNELUTIL-PROXYKIND.
  88 ES-CHANNELUTIL-ALL-PROXYS VALUE 0.
  88 ES-CHANNELUTIL-PROXY-CONSUMER VALUE 1.
  88 ES-CHANNELUTIL-PROXY-SUPPLIER VALUE 2.
  88 ES-CHANNELUTIL-PROXY-PULL-CONS VALUE 3.
  88 ES-CHANNELUTIL-PROXY-PULL-SUPP VALUE 4.
  88 ES-CHANNELUTIL-PROXY-PUSH-CONS VALUE 5.
  88 ES-CHANNELUTIL-PROXY-PUSH-SUPP VALUE 6.
  88 ES-CHANNELUTIL-PROXY-PULL-C001 VALUE 7.
  88 ES-CHANNELUTIL-PROXY-PULL-S001 VALUE 8.
  88 ES-CHANNELUTIL-PROXY-PUSH-C001 VALUE 9.
  88 ES-CHANNELUTIL-PROXY-PUSH-S001 VALUE 10.
  88 ES-CHANNELUTIL-STRUCTURED-PROX VALUE 11.
  88 ES-CHANNELUTIL-STRUCTURED-P001 VALUE 12.
  88 ES-CHANNELUTIL-STRUCTURED-P002 VALUE 13.
  88 ES-CHANNELUTIL-STRUCTURED-P003 VALUE 14.
```

4.21.1 COSEVENTCHANNELADMIN Interface

This section details the functions associated with the COSEVENTCHANNELADMIN Interface.

4.21.1.1 COSEVENTCHANNELADMIN-EVENTCHANNEL-CREATE-UTIL

Name

COSEVENTCHANNELADMIN-EVENTCHANNEL-CREATE-UTIL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT  BY OBJ.
01 COPY UTILTYPE IN COSEVENTCOMM REPLACING COSEVENTCHANNELADMIN-EVENTC001 BY UTIL-TYPE.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY CHANNELUTIL IN COSEVENTCOMM REPLACING ES-CHANNELUTIL BY CHNLUTIL.
```

PROCEDURE DIVISION.

```
CALL "COSEVENTCHANNELADMIN-EVENTCHANNEL-CREATE-UTIL" USING
          OBJ
```

```
        UTIL-TYPE
        ENV
        CHNLUTIL
```

Description

Creates the ES-CHANNELUTIL interace object reference.

This method secures an area for holding the object references. Use CORBA-OBJECT-RELEASE to release the area when it is no longer required.

Parameters

OBJ

The COSEVENTCHANNELADMIN-EVENTCHANNEL object reference.

UTIL-TYPE

The interface type.

Specify the interface type in *UTIL-TYPE*.

Value specifiable in *UTIL-TYPE*	Return value information
COSEVENTCHANNELADMIN-EVENTCHAN	Fetches ES-CHANNELUTIL interface object reference

ENV

A structure that may contain exception information.

CHNLUTIL

This is the object reference for the ES-CHANNELUTIL interface.

Return Values

When this function terminates abnormally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.21.1.2 Interface Inherited

The following interface can be inherited and used: For details, refer to "4.21.1.1 COSEVENTCHANNELADMIN-EVENTCHANNEL-CREATE-UTIL".

(1) COSNOTIFYCHANNELADMIN-EVENTCHANNEL-CREATE-UTIL

4.21.2 ES Interface

This section details the functions associated with the ES Interface.

4.21.2.1 ES-CHANNELUTIL-GET-PROXYS

Name

ES-CHANNELUTIL-GET-PROXYS

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY CHANNELUTIL IN COSEVENTCOMM REPLACING ES-CHANNELUTIL BY OBJ.
01 COPY PROXYKIND IN COSEVENTCOMM REPLACING ES-CHANNELUTIL-PROXYKIND BY PROXY-KIND.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 PROXY-DATA-SEQ-PTR USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "ES-CHANNELUTIL-GET-PROXYS" USING
          OBJ
          PROXY-KIND
          ENV
          PROXY-DATA-SEQ-PTR.
```

Description

Fetches consumers/suppliers connected to the Event Channel, and connection information.

This method secures areas for holding the connection information and suppliers/consumers connected to the Event Channel. Use CORBA-FREE to release the areas when they are no longer required.

Parameters

OBJ

The ES-CHANNELUTIL object reference.

PROXY-KIND

The type of Proxy object fetched.

The connection information fetched depends on the value specified in *UTIL-KIND*. The table below lists the values specified in *PROXY-KIND* and the corresponding return inforrmation.

Table 4.3 PROXY-KIND Values and Corresponding Return Information

Allowed Value for *PROXY-KIND*	Return Value Information
ES-CHANNELUTIL-ALL-PROXYS	Fetches all connection information
ES-CHANNELUTIL-PROXY-CONSUMER	Fetches a list of all PROXYCONSUMER
ES-CHANNELUTIL-PROXY-SUPPLIER	Fetches a list of all PROXYSUPPLIER
ES-CHANNELUTIL-PROXY-PULL-CONS	Fetches a list of COSEVENTCHANNELADMIN PROXYPULLCONSUMER
ES-CHANNELUTIL-PROXY-PULL-SUPP	Fetches a list of COSEVENTCHANNELADMIN PROXYPULLSUPPLIER
ES-CHANNELUTIL-PROXY-PUSH-CONS	Fetches a list of COSEVENTCHANNELADMIN PROXYPUSHCONSUMER
ES-CHANNELUTIL-PROXY-PUSH-SUPP	Fetches a list of COSEVENTCHANNELADMIN PROXYPUSHSUPPLIER
ES-CHANNELUTIL-PROXY-PULL-C001	Fetches a list of COSNOTIFYCHANNELADMIN PROXYPULLCONSUMER

Allowed Value for *PROXY-KIND*	Return Value Information
ES-CHANNELUTIL-PROXY-PULL-S001	Fetches a list of COSNOTIFYCHANNELADMIN PROXYPULLSUPPLIER
ES-CHANNELUTIL-PROXY-PUSH-C001	Fetches a list of COSNOTIFYCHANNELADMIN PROXYPUSHCONSUMER
ES-CHANNELUTIL-PROXY-PUSH-S001	Fetches a list of COSNOTIFYCHANNELADMIN PROXYPUSHSUPPLIER
ES-CHANNELUTIL-STRUCTURED-PROX	Fetches a list of COSNOTIFYCHANNELADMIN STRUCTUREDPROXYPULLCONSUMER
ES-CHANNELUTIL-STRUCTURED-P001	Fetches a list of COSNOTIFYCHANNELADMIN STRUCTUREDPROXYPULLSUPPLIER
ES-CHANNELUTIL-STRUCTURED-P002	Fetches a list of COSNOTIFYCHANNELADMIN STRUCTUREDPROXYPUSHCONSUMER
ES-CHANNELUTIL-STRUCTURED-P003	Fetches a list of COSNOTIFYCHANNELADMIN STRUCTUREDPROXYPUSHSUPPLIER

The following table shows the values that can be specified for ES-CHANNELUTIL-PROXYDATA structure members:

Table 4.4 ES-CHANNELUTIL-PROXYDATA Structure Members and Settings

Member	Setting
PROXY	Object references for consumers/suppliers connected to the Event Channel.
TIME	Time connected to the Event Channel.
IPADDRESS	IP addresses of consumers/suppliers connected to the Event Channel.
KIND	Proxy types of consumers/suppliers connected to the Event Channel. ES-CHANNELUTIL-PROXY-CONSUMER ES-CHANNELUTIL-PROXY-SUPPLIER ES-CHANNELUTIL-PROXY-PULL-CONS ES-CHANNELUTIL-PROXY-PULL-SUPP ES-CHANNELUTIL-PROXY-PUSH-CONS ES-CHANNELUTIL-PROXY-PUSH-SUPP ES-CHANNELUTIL-PROXY-PULL-C001 ES-CHANNELUTIL-PROXY-PULL-S001 ES-CHANNELUTIL-PROXY-PUSH-C001 ES-CHANNELUTIL-PROXY-PUSH-S001 ES-CHANNELUTIL-STRUCTURED-PROX ES-CHANNELUTIL-STRUCTURED-P001 ES-CHANNELUTIL-STRUCTURED-P002 ES-CHANNELUTIL-STRUCTURED-P003

ENV

A structure that may contain exception information.

PROXY-DATA-SEQ-PTR

This is the connection information about consumers/suppliers connected to the Event Channel.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Notes

The 'IPADDRESS' member is set to 0 when using the IP address in an IPv6 environment. In an IPv6 environment, use ES-CHANNELUTIL-GET-PROXYS6.

4.21.2.2 ES-CHANNELUTIL-GET-PROXYS6

Name

ES-CHANNELUTIL-GET-PROXYS6

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY CHANNELUTIL IN COSEVENTCOMM REPLACING ES-CHANNELUTIL BY OBJ.
01 COPY PROXYKIND IN COSEVENTCOMM REPLACING ES-CHANNELUTIL-PROXYKIND BY PROXY-KIND.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 PROXY-DATA6-SEQ-PTR USAGE POINTER.
```

PROCEDURE DIVISION.

```
CALL "ES-CHANNELUTIL-GET-PROXYS6" USING
          OBJ
          PROXY-KIND
          ENV
          PROXY-DATA6-SEQ-PTR.
```

Description

Fetches consumers/suppliers connected to the Event Channel and connection information.

When 'IPADDRESS' is set via the IPv6 form, 'IP-FORMAT' is set to 1. When 'IPADDRESS' is set via the IPv4 form, 'IP-FORMAT' is set to 0.

This method secures areas for storing the connection information and suppliers/consumers connected to the Event Channel. Use CORBA-FREE to release these areas when they are no longer required.

Parameters

OBJ

The ES-CHANNELUTIL object reference.

PROXY-KIND

The type of PROXY object fetched.

The connection information fetched depends on the value specified in *PROXY-KIND*. For details of values that can be specified in *UTIL-KIND*, and the corresponding return inforrmation, refer to Table 4.3 PROXY-KIND Values and Corresponding Return Information under ES-CHANNELUTIL-GET-PROXYS.

ENV

A structure that can contain exception information.

PROXY-DATA6-SEQ-PTR

This is the connection information about consumers/suppliers connected to the Event Channel.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure, and the consumers/suppliers connected to the Event Channel and the connection information are returned.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

Notes

The IPv4 client member 'IPADDRESS' is set to the parallel IPv4 address when using the IP address in a IPv6 environment. For configuration details when operating in a IPv6 environment, refer to "Operating in an IPv6 Environment" and "config" in the "CORBA Service Environment Definition" appendix of the Tuning Guide.

4.21.2.3 ES-CHANNELUTIL-GET-CONSUMER-COUNT

Name

ES-CHANNELUTIL-GET-CONSUMER-COUNT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY CHANNELUTIL IN COSEVENTCOMM REPLACING ES-CHANNELUTIL BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY CON-COUNT.
```

PROCEDURE DIVISION.

```
CALL "ES-CHANNELUTIL-GET-CONSUMER-COUNT" USING
          OBJ
          ENV
          CON-COUNT.
```

Description

Fetches the number of consumers connected to the Event Channel.

Parameters

OBJ

The ES-CHANNELUTIL object reference.

ENV

A structure that may contain exception information.

CON-COUNT

This is the number of consumers connected to the Event Channel.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.21.2.4 ES-CHANNELUTIL-GET-SUPPLIER-COUNT

Name

ES-CHANNELUTIL-GET-SUPPLIER-COUNT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY CHANNELUTIL IN COSEVENTCOMM REPLACING ES-CHANNELUTIL BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY SUP-COUNT.
```

PROCEDURE DIVISION.

```
CALL "ES-CHANNELUTIL-GET-SUPPLIER-COUNT" USING
          OBJ
          ENV
          SUP-COUNT.
```

Description

Fetches the number of suppliers connected to the Event Channel.

Parameters

OBJ

The ES-CHANNELUTIL object reference.

ENV

A structure that may contain exception information.

SUP-COUNT

This is the number of suppliers connected to the Event Channel.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.21.2.5 ES-CHANNELUTIL-GET-QUEUE-LENGTH

Name

ES-CHANNELUTIL-GET-QUEUE-LENGTH

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY CHANNELUTIL IN COSEVENTCOMM REPLACING ES-CHANNELUTIL BY OBJ.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
01 COPY ULONG IN CORBA REPLACING CORBA-UNSIGNED-LONG BY QUEUE-LENGTH.
```

PROCEDURE DIVISION.

```
CALL "ES-CHANNELUTIL-GET-QUEUE-LENGTH" USING
          OBJ
          ENV
          QUEUE-LENGTH.
```

Description

Fetches the number of event data items queued in the Event Channel.

Parameters

OBJ

The ES-CHANNELUTIL object reference.

ENV

A structure that may contain exception information.

QUEUE-LENGTH

This is the number of event data items queued in the Event Channel.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.21.2.6 ES-CHANNELUTIL-LOCAL-BEGIN

Name

ES-CHANNELUTIL-LOCAL-BEGIN

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY CHANNELUTIL IN COSEVENTCOMM REPLACING ES-CHANNELUTIL BY OBJ.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PROXY.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "ES-CHANNELUTIL-LOCAL-BEGIN" USING
          OBJ
          PROXY
          ENV.
```

Description

Notifies the channel of starting the local transaction.

In the case of consumer, the following methods can be issued before ES-CHANNELUTIL-LOCAL-COMMIT or ES-CHANNELUTIL-LOCAL-ROLLBACK is issued.

- COSEVENTCOMM-PULLSUPPLIER-PULL
- COSEVENTCOMM-PULLSUPPLIER-TRY-PULL
- COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-PULL-STRUCTURED-EVENT
- COSNOTIFYCOMM-STRUCTUREDPULLSUPPLIER-TRY-PULL-STRUCTURED-EVENT

In the case of supplier, the following methods can be issued before ES-CHANNELUTIL-LOCAL-COMMIT or ES-CHANNELUTIL-LOCAL-ROLLBACK is issued.

- COSEVENTCOMM-PUSHCONSUMER-PUSH
- COSNOTIFYCOMM-STRUCTUREDPUSHCONSUMER-PUSH-STRUCTURED-EVENT

If ES-CHANNELUTIL-LOCAL-COMMIT or ES-CHANNELUTIL-LOCAL-ROLLBACK is not issued after issuance of this method, they are automatically rolled back by the notification service immediately after the local transaction timeout time passes.

This method cannot be issued twice for a PROXY.

Parameters

OBJ

The ES-CHANNELUTIL object reference.

PROXY

The object reference returned by *COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER* or *COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER*.

ENV

A structure that may contain exception information.

Return Values

For normal termination, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

For abnormal termination, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detail information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.21.2.7 ES-CHANNELUTIL-LOCAL-COMMIT

Name

ES-CHANNELUTIL-LOCAL-COMMIT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY CHANNELUTIL IN COSEVENTCOMM REPLACING ES-CHANNELUTIL BY OBJ.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PROXY.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "ES-CHANNELUTIL-LOCAL-COMMIT" USING
          OBJ
          PROXY
          ENV.
```

Description

Notifies the channel of completing the local transaction.

For a consumer, message reception from the Event Channel completes when this method completes.

For a supplier, sending of data from the Event Channel completes when this method completes.

Parameters

OBJ

The ES-CHANNELUTIL object reference.

PROXY

The object reference returned by *COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER* or *COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER*.

ENV

A structure that may contain exception information.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.21.2.8 ES-CHANNELUTIL-LOCAL-ROLLBACK

Name

ES-CHANNELUTIL-LOCAL-ROLLBACK

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY CHANNELUTIL IN COSEVENTCOMM REPLACING ES-CHANNELUTIL BY OBJ.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PROXY.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "ES-CHANNELUTIL-LOCAL-ROLLBACK" USING
          OBJ
          PROXY
          ENV.
```

Description

Notifies the channel of cancellation of the local transaction. Calling this method allows the Event Channel status to return to what it was before ES-CHANNELUTIL-LOCAL-BEGIN was issued.

Parameters

OBJ

The ES-CHANNELUTIL object reference.

PROXY

The object reference returned by *COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER* or *COSNOTIFYCHANNELADMIN-SUPPLIERADMIN-OBTAIN-NOTIFICATION-PUSH-CONSUMER*.

ENV

A structure that may contain exception information.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.21.2.9 ES-CHANNELUTIL-PULL-CANCEL

Name

ES-CHANNELUTIL-PULL-CANCEL

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY CHANNELUTIL IN COSEVENTCOMM REPLACING ES-CHANNELUTIL BY OBJ.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PROXY.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "ES-CHANNELUTIL-PULL-CANCEL" USING
          OBJ
          PROXY
          ENV.
```

Description

This method cancels the pull waiting status (waiting status created by issuing pull and ES-CHANNELUTIL-PULL-WAIT) of the target PROXY.

Parameters

OBJ

The ES-CHANNELUTIL object reference.

PROXY

The object reference returned by *COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER*.

Note: Specify the object reference specified in pull and ES-CHANNELUTIL-PULL-WAIT.

ENV

A structure that may contain exception information.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.

4.21.2.10 ES-CHANNELUTIL-PULL-WAIT

Name

ES-CHANNELUTIL-PULL-WAIT

Synopsis

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

```
SPECIAL-NAMES.
SYMBOLIC CONSTANT
COPY SYMBOL-CONST IN CORBA.
.
```

DATA DIVISION.
WORKING-STORAGE SECTION.

```
COPY CONST IN COSEVENTCOMM.
COPY CONST IN CORBA.
01 COPY CHANNELUTIL IN COSEVENTCOMM REPLACING ES-CHANNELUTIL BY OBJ.
01 COPY OBJECT IN CORBA REPLACING CORBA-OBJECT BY PROXY.
01 COPY ENVIRONMENT IN CORBA REPLACING CORBA-ENVIRONMENT BY ENV.
```

PROCEDURE DIVISION.

```
CALL "ES-CHANNELUTIL-PULL-WAIT" USING
          OBJ
          PROXY
          ENV.
```

Description

This method waits until the status of the target PROXY becomes enabled to allow the PROXY to fetch data with the PULL method or the TRY-PULL method. The method returns to the invoking side when the status becomes enabled to allow fetching data after execution of this method.

The method returns immediately if the target PROXY is already enabled to fetch data with the PULL method. If the status does not become enabled to allow fetching data within the waiting time, this method returns with timeout.

Parameters

OBJ

The ES-CHANNELUTIL object reference.

PROXY

The object reference returned by *COSNOTIFYCHANNELADMIN-CONSUMERADMIN-OBTAIN-NOTIFICATION-PULL-SUPPLIER*.

ENV

A structure that may contain exception information.

Return Values

When this function terminates normally, CORBA-NO-EXCEPTION is set in MAJOR of the ENV structure.

If it terminates abnormally, CORBA-SYSTEM-EXCEPTION or CORBA-USER-EXCEPTION is set in MAJOR in the ENV structure, and detailed information is set in ID in the ENV structure.

In the event of a user exception, the following detailed information is specified.

EX-COSEVENTCOMM-DISCONNECTED

Not connected to the Event Channel.

If automatic collection of connection information is enabled when the event channel is generated (i.e. the -autodiscon option is specified when the *esmkchnl* command is executed), the connection may close because the CORBA Service client non-communication monitoring time is exceeded. To continue communication, use the Event Channel connection.

In the event of a system exception, refer to the "Exception Information Minor Codes Reported from the Event Service" and "Exception Information Minor Codes Reported from the CORBA Service" chapters of the Messages manual, and correct the problem.