FUJITSU

FUJITSU Software
Interstage Studio

General Description

B1WD-3157-02ENZ0(00)
November 2013

Preface

Interstage Studio is a component-oriented Java integrated development environment that supports development of Web applications, EJBs, and Web services.

Interstage Studio runs on the following operating systems:

- Microsoft(R) Windows(R) XP Home Edition operating system, Microsoft(R) Windows(R) XP Professional operating system
  Hereafter abbreviated as Windows XP
- Microsoft(R) Windows Server(R) 2003, Standard Edition, Microsoft(R) Windows Server(R) 2003, Enterprise Edition, Microsoft(R) Windows Server(R) 2003, Standard x64 Edition, Microsoft(R) Windows Server(R) 2003, Enterprise x64 Edition, Microsoft(R) Windows Server(R) 2003 R2, Standard Edition, Microsoft(R) Windows Server(R) 2003 R2, Enterprise Edition, Microsoft(R) Windows Server(R) 2003 R2, Standard x64 Edition, Microsoft(R) Windows Server(R) 2003 R2, Enterprise x64 Edition
  Hereafter abbreviated as Windows Server 2003
- Windows Vista(R) Home Basic, Home Premium, Business, Enterprise and Ultimate
  Hereafter abbreviated as Windows Vista
- Microsoft(R) Windows Server(R) 2008 Foundation, Microsoft(R) Windows Server(R) 2008 Standard, Microsoft(R) Windows Server(R) 2008 Enterprise, Microsoft(R) Windows Server(R) 2008 Standard without Hyper-V(TM), Microsoft(R) Windows Server(R) 2008 Enterprise without Hyper-V(TM), Microsoft(R) Windows Server(R) 2008 R2 Foundation, Microsoft(R) Windows Server(R) 2008 R2 Standard, Microsoft(R) Windows Server(R) 2008 R2 Enterprise
  Hereafter abbreviated as Windows Server 2008
- Windows(R) 7 Home Premium, Professional, Enterprise and Ultimate
  Hereafter abbreviated as Windows 7
- Microsoft(R) Windows Server(R) 2012 Foundation, Microsoft(R) Windows Server(R) 2012 Standard, Microsoft(R) Windows Server(R) 2012 Datacenter
  Hereafter abbreviated as Windows Server 2012
- Windows(R) 8, Pro, Enterprise
  Hereafter abbreviated as Windows 8

Purpose of This Manual

This manual provides information that users must be familiar with before they start using Interstage Studio to develop applications. By reading the manual, users can get the product overview and learn the basics of the product.

Intended Readers

This manual was written with an assumption that the reader has a basic knowledge of Windows and application development, including processing of GUI or databases.

Structure of this Manual

This manual consists of the following chapters:

- Chapter 1 Overview of Interstage Studio

  Provides an overview and explains the positioning, organization, and application development policy of Interstage Studio.

- Chapter 2 Features of Interstage Studio

  Explains features of the Java integrated development environment provided by Interstage Studio.

- Chapter 3 Functions of Interstage Studio

  Provides an overview of each function of the Java integrated development environment provided by Interstage Studio.

- Chapter 4 Supported Server Linkages

  Explains server linkages supported by the applications that can be developed with Interstage Studio.

- Chapter 5 Supported Components

  Explains components that can be developed with Interstage Studio, in sections organized by application type.

- Appendix A Online Manuals

  Provides a list of the online manuals available for Interstage Studio.

Conventions

- This manual uses the following notation:
  - [...]: Indicates the name of a menu item or button.
- "Interstage Application Server" is abbreviated as "Interstage".
- "Oracle Solaris" is abbreviated as "Solaris".
- "Enterprise JavaBeans" is abbreviated as "EJB".
- "Java TM Platform, Enterprise Edition" is abbreviated as "Java EE".
- "Java TM 2 Platform, Enterprise Edition" is abbreviated as "J2EE".
- "JavaServer Pages" is abbreviated as "JSP".
- "Simple Object Access Protocol" is abbreviated as "SOAP".
- "Common Object Request Broker Architecture" is abbreviated as "CORBA".

About product names

In this manual, "Interstage Studio Standard-J Edition" is abbreviated as "Interstage Studio".

Export Controls

This document or a portion thereof may not be exported (or re-exported) without authorization from the appropriate government authorities in accordance with the pertinent laws.

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

Revision History

Date of Publication and Version	Manual Code
November 2013: 2nd Version	B1WD-3157-02ENZ0(00)/B1WD-3157-02ENZ2(00)
November 2012: 1st Version	B1WD-3157-01ENZ0(00)/B1WD-3157-01ENZ2(00)

Copyright Notice

Applications that Can be Developed

The table below lists applications that can be developed with this product. The listed applications are classified into three groups corresponding to processing logic layers in the three-tier model. For an overview of this product and details on the tree-tier model, refer to "Chapter 1 Overview of Interstage Studio".

Application lists

Processing logic layer	Application type	
Presentation	Java application	Development of Java applications, Applet and JavaBeans
	Web application	Development of Web applications that use HTML, Servlet or JSP
Business logic	Enterprise Bean	Development of Enterprise JavaBeans
	Web service application	Development of Web service applications and Web service client applications
Presentation and business logic	Enterprise application	Development of enterprise applications
	Apcoordinator application	Development of Web applications that use an application framework
		Development of Enterprise JavaBeans that use an application framework

Contents

Chapter 1 Overview of Interstage Studio

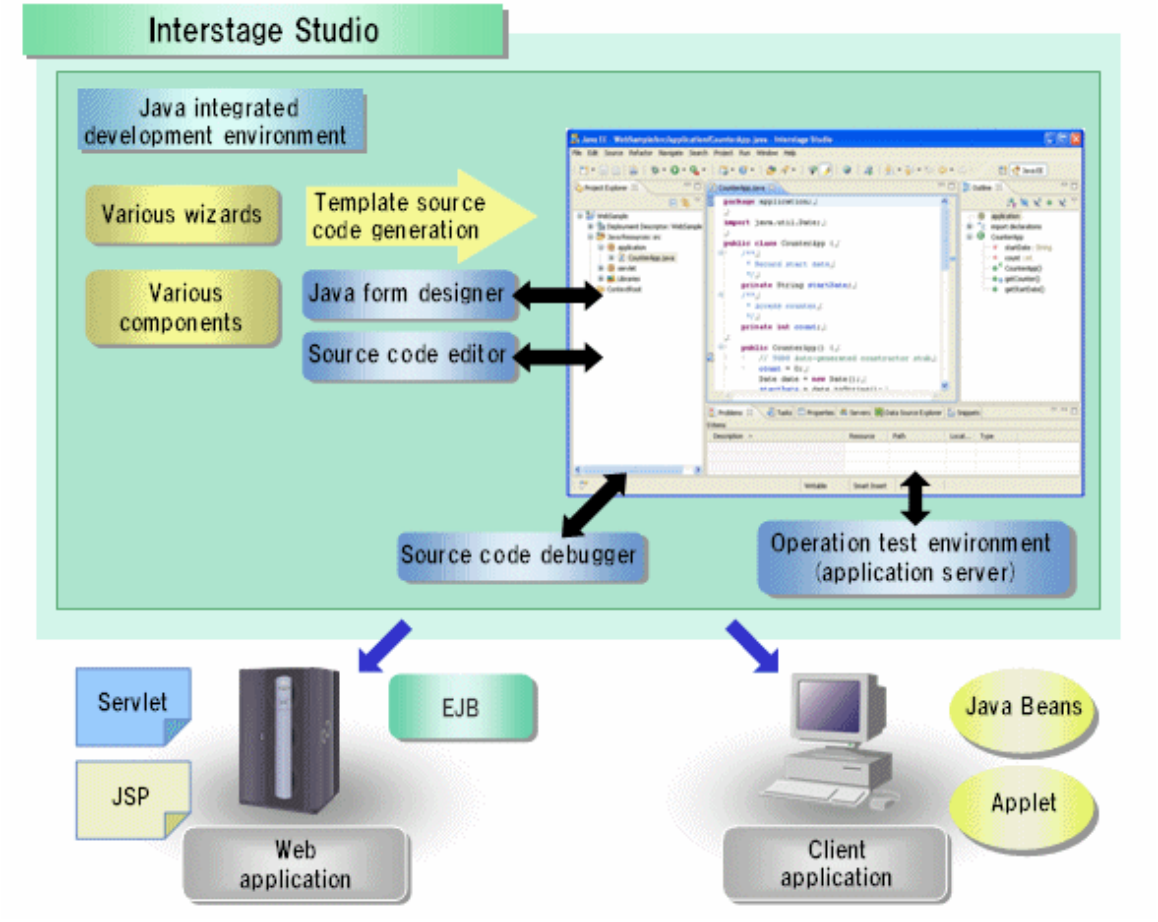
This chapter provides an overview of Interstage Studio and explains its application development policy and usage as well as supported application development.

1.1 About Interstage Studio

Interstage Studio is a component-oriented Java integrated development environment for the development of various applications based on Java/Java EE/J2EE, Web service, and similar open standard infrastructure technology. It provides a working environment that is easy to understand, even for novice users, through an interactive GUI-based development environment, while supporting a variety of functions that enable efficient development of various applications.

- Efficient application development via a Java integrated development environment that uses open source Eclipse
- Easy operation, from development through to debugging, deployment, and operation
- Framework use enabling quick construction of Web/Java systems
- Reduction of system management and operation costs due to a multi-server batch deployment function

Figure 1.1 Overview of Interstage Studio

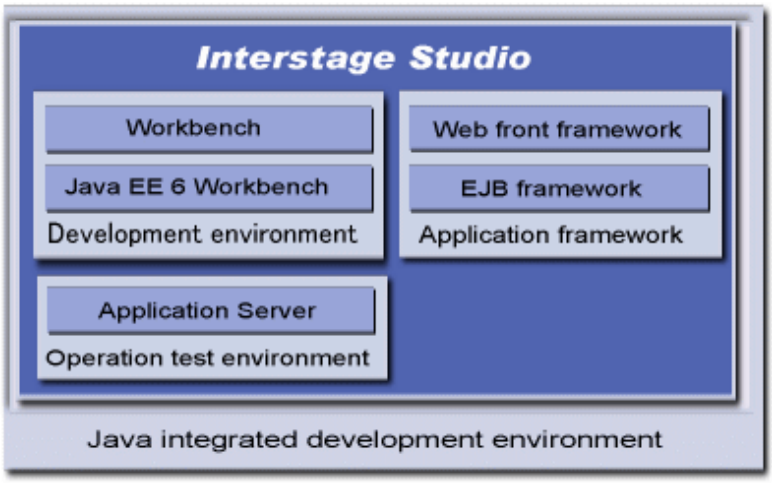


For details on applications that can be developed with Interstage Studio, refer to "1.4 Application Development Supported by Interstage Studio".

1.1.1 Function Configuration

Interstage Studio provides development environment, operation test environment, and application framework functions as the Java integrated development environment.

Figure 1.2 Function configuration of Interstage Studio



- Development environment

  Interstage Studio can be used to develop a wide variety of applications, including Web front ends and EJB/Web service applications. It is possible to develop Java EE/J2EE server side applications (JSP/Servlet/EJB/Web services) and client applications such as Applets. The seamless integration of source editors and debuggers simplifies operation, from coding through to testing. In addition, application development is made more efficient by the provision of a refactoring function for performing batched changes to class names, method names, and similar in other source files; a real-time syntax check function for source entered in the source editor; and various wizard functions. Linkage with Interstage Application Development Cycle Manager associates development resources with development processes and enables integrated management.

Information

The Java EE 6 workbench is provided for the development of Java EE 6 application and Java application that uses JDK 7.

- Operation test environment

  The Interstage Application Server execution environment is in-built, making it easy to use the one machine environment for development through to debugging when developing client/server Java applications. In addition, application deployment from the development environment to the operation machine is possible via a simple operation. This enables smooth application debugging and deployment of resources to the operation machine, thus enabling efficient application development.

- Application framework

  The application framework supports the development of the following applications:

  - Web applications
  - EJB

  The application framework allows you to build a high-quality Web system quickly and more efficiently.

With these functions, you can perform the series of operations from application development to the operational testing of the developed applications efficiently in a standalone environment.

1.1.2 Interstage Studio Product Line

Interstage Studio provides as a Java integrated development environment a variety of products corresponding to the types of applications to be developed and system configurations.

Table 1.1 Interstage Studio products

Product name	Description
Interstage Studio Standard-J Edition	This product provides functions for developing Java applications and Java EE/J2EE applications. It enables development of various types of systems, ranging from simple Web systems to three-tier large-scale business systems.
Interstage Studio Client Runtime	This product is required for operation of an application that was developed with Interstage Studio and runs on a client.

For information about the types of applications that can be developed with this product, refer to "Applications that Can be Developed".

1.1.3 Operating Environments

This section explains the operating environment of each Interstage Studio product and the operating environments of the applications developed with each Interstage Studio product. For additional information, refer to the software release guide for each product.

Operating environments of Interstage Studio products

The table below lists the operating environments of each Interstage Studio product.

Operating environment (Operating system)		Product type	
		Studio	Client
Windows XP	Professional	Y	Y
	Home Edition	Y (*1) (*2)	Y
Windows Server 2003		Y	Y
Windows Vista	Home Basic	Y (*1) (*2)	Y
	Home Premium	Y (*1) (*2)	Y
	Business	Y	Y
	Enterprise	Y	Y
	Ultimate	Y	Y
Windows Server 2008		Y	Y
Windows 7	Home Premium	Y (*1) (*2)	Y
	Professional	Y	Y
	Enterprise	Y	Y
	Ultimate	Y	Y
Windows Server 2012		Y	Y
Windows 8	Windows 8 (Pro and Enterprise are excluded.)	Y (*1) (*2)	Y
	Pro	Y	Y
	Enterprise	Y	Y

Product type

Studio: Interstage Studio Standard-J Edition
Client: Interstage Studio Client Runtime

Legend

Y: Can be used

Remarks

(*1): The following functionality cannot be used.

- Stand-alone debugging using Application Server

(*2): The following product is needed in the building of applications that use the Application Server function in these editions.

- Interstage Application Server Client Package

Operating environments of developed applications

Applications that have been developed will operate in the following environments.

Client applications

Developed client application will operate in the following environments:

- Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2012, Windows 8

Server applications

Interstage Application Server is required to run a server application developed with this product. The operating environment of the server application conforms to the Interstage Application Server operating environment.

1.2 Application Development Policy

This section explains the application development policy of Interstage Studio, specifically the design philosophy in application development and operation principles of applications.

1.2.1 Design Philosophy in Application Development

Interstage Studio adopts the three-tier model as the design philosophy for system development of Web applications and Web services.

Three-tier model

The design philosophy realized by the three-tier model has processing in a system logically divided into three processing logic layers, as listed below, for development and operation. These processing logic layers are flexibly distributed and linked together using open technologies.

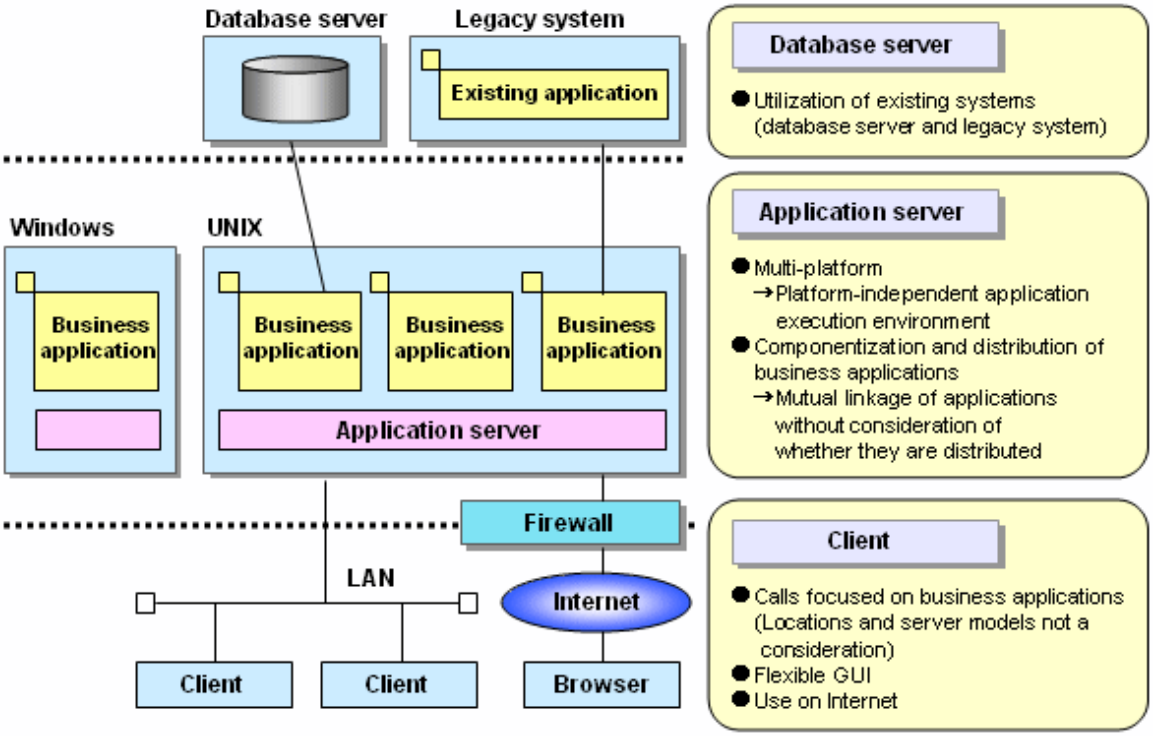
Table 1.2 Classification of logic layers in the three-tier model

Processing logic layer	Description
Presentation logic	Controls all user interfaces. For example, its processing includes: - Displaying an initial screen to a user - Receiving information entered by a user and checking the information at the syntax level - Displaying processing results and other information
Business logic	Executes processing of the application itself according to user's instructions received from screen display control. For example, its processing includes: - Financial accounting - Inventory control and related tasks If necessary, this layer is linked with the database logic layer to perform database operations.
Database logic	Performs database operations according to instructions from the business logic layer. This layer guarantees database consistency and executes reference, update, and other database processing tasks.

Processing logic layer	Description
	Normally, the database management system (DBMS), such as Symfoware or Oracle, processes the tasks.

The figure below outlines the three-tier model.

Figure 1.3 Outline of application development using the three-tier model



By distributing the process logic layers as follows, the three-tier model enables development of an efficient system that has high scalability and maintainability:

- Presentation logic (screen display control, etc.)
  Clients
- Business logic
  Application server
- Database logic
  Database server

The application server provides an execution environment for business logic and linkage between applications.

1.2.2 Operation Principles of Applications

Interstage Studio enables development of applications that conform to operation principles as follows:

Applications operating with Servlet services

These applications are industry standard Java program components activated through HTML. They are executed on servers, and they return HTML code to browsers.
These applications execute business processing that is linked with other applications at the back end and generate dynamic pages. Technologies that are used for these applications offer not only substitutes for the common gateway interface (CGI) but also various other advantages.

Applications operating with Web services

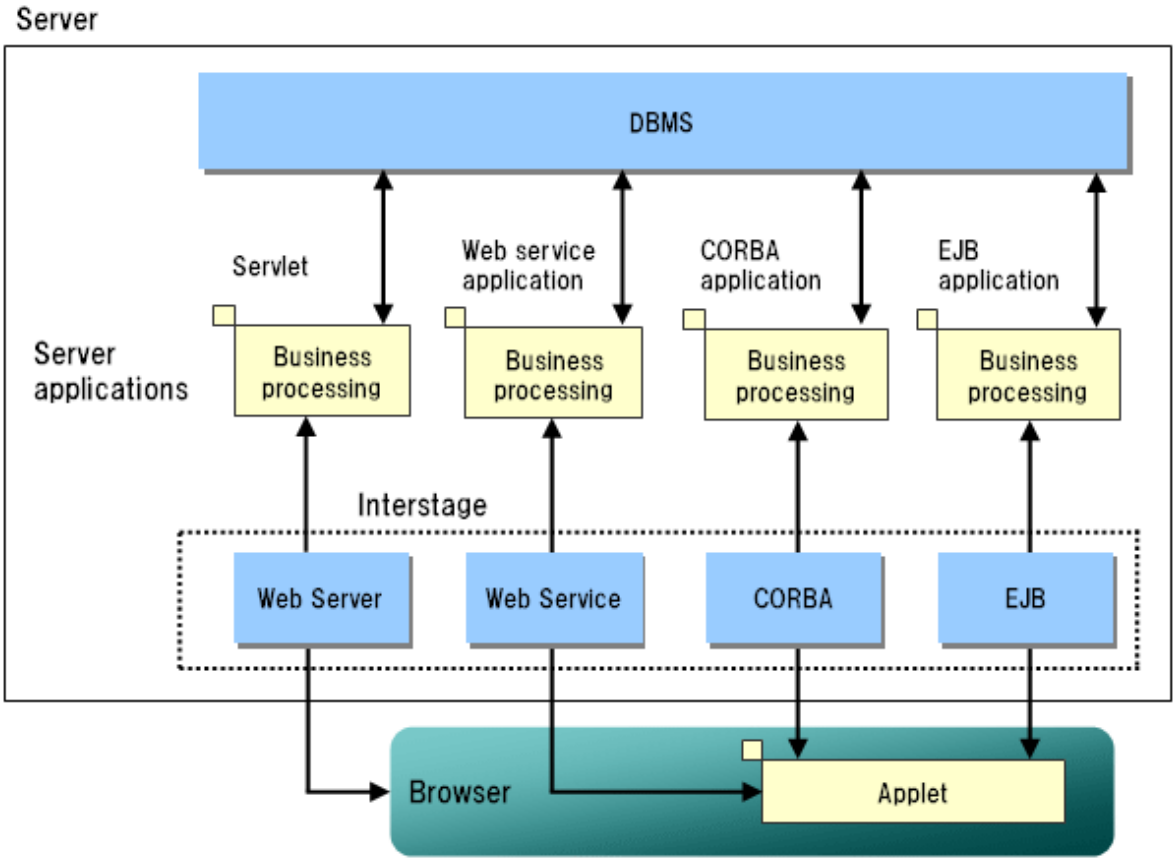
These include Web services that are developed according to the SOAP specification, and client applications that use the services.

Applications operating with EJB services

EJB applications are client/server applications using the industry standard component architecture.
They are suitable for development of full-fledged business applications that have high functionality and the potential for wide distribution.

The figure below outlines the operation principles of applications by application type.

Figure 1.4 Operation principles of applications



1.3 Development Environment of Interstage Studio

This section provides an overview of the development environment. For details on the development environment, refer to the following development environment manuals:

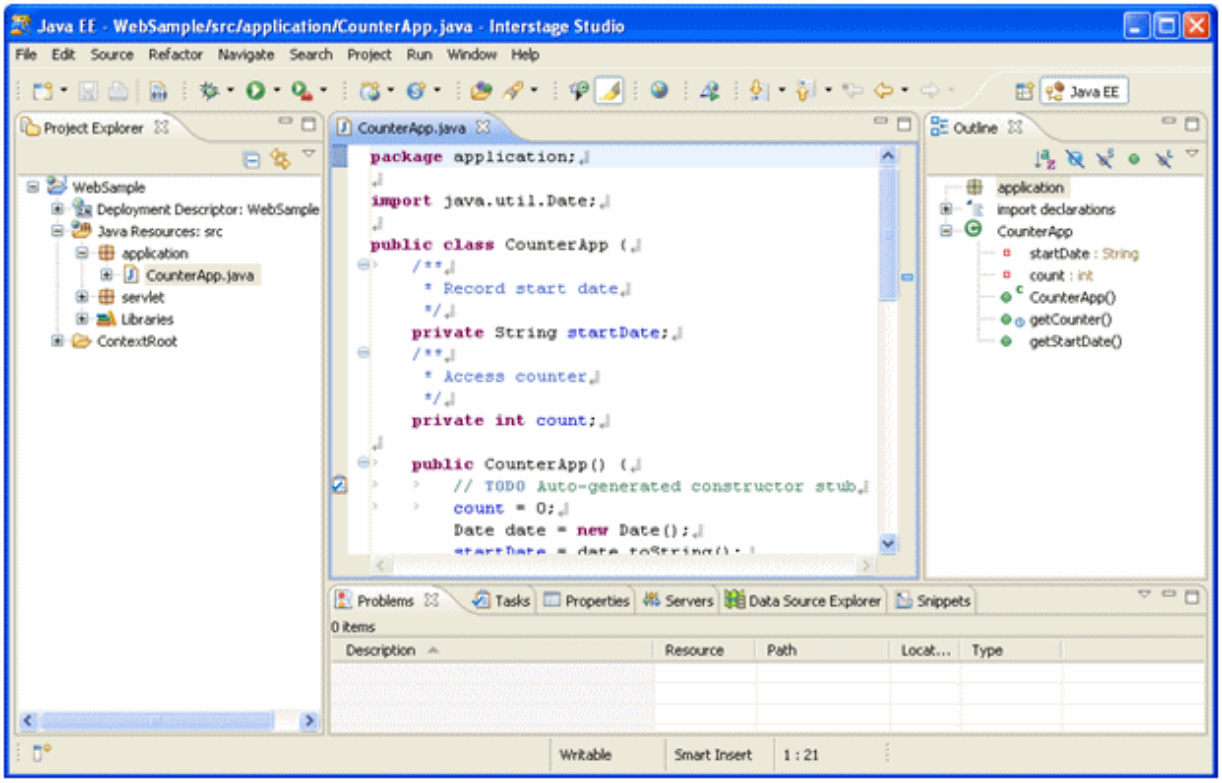
- Interstage Studio User's Guide

1.3.1 Overview of the Development Environment

Example of the development environment

Examples of the Interstage Studio workbench are displayed below. For details on the features of each development environment, refer to "Chapter 3 Functions of Interstage Studio", "3.1 Development Environment".

Figure 1.5 Overview of the Interstage Studio workbench



Configuration of the development environment

The development environment includes various design tools (development support tools) and components that are useful for efficient application development. With these tools and components, the development environment can be used as a comprehensive Java integrated development environment covering a wide range of development work, from application design and development to debugging.

For an overview of the various design tools that are integrated in the development environment, refer to "3.2 Design Tools" in "Chapter 3 Functions of Interstage Studio".

1.3.2 Units of Application Development

Interstage Studio manages each application under development as a single **project**. In other words, it collectively manages development resources (such as source files and libraries) necessary for application development and operation information in units of projects.

A project is the unit of application development in Interstage Studio. A project can be efficiently and easily defined with a wizard.

Project types

The project types listed below can be defined. For details on projects, refer to the manuals about the development environment.

- Java
  Java Application Project
  Java Project
- EJB
  EJB Project
- Java EE
  Application Client Project
  Enterprise Application Project
  Utility Project
- JPA
  JPA Project

- Web
  Dynamic Web Project

- Apcoordinator
  Enterprise JavaBeans Project (Apcoordinator)
  Web Application Project (Apcoordinator)

1.4 Application Development Supported by Interstage Studio

Various types of applications, ranging from standalone applications that run on client systems to multi-tier client/server applications, can be developed with Interstage Studio. Web-based server applications can also be developed.

1.4.1 Application List

This section explains the types of application that can be developed with Interstage Studio, by outlining each of the application types.

The table below lists and outlines the supported application types.

Table 1.3 Applications that can be developed using the Interstage Studio

Application type	Description of applications
Java application	Java applications, applets, and JavaBeans, all of which are coded in Java and can be run on various platforms
Web application	Servlets that run on Web servers and Web applications that use JSP
Enterprise JavaBeans	Enterprise Beans based on EJB, which is a distributed object architecture
Enterprise application	Multi-tier applications that are developed by combining EJB, Web applications, Java EE application clients, and J2EE application clients
Web service application	Server-side Web service applications that run on Interstage

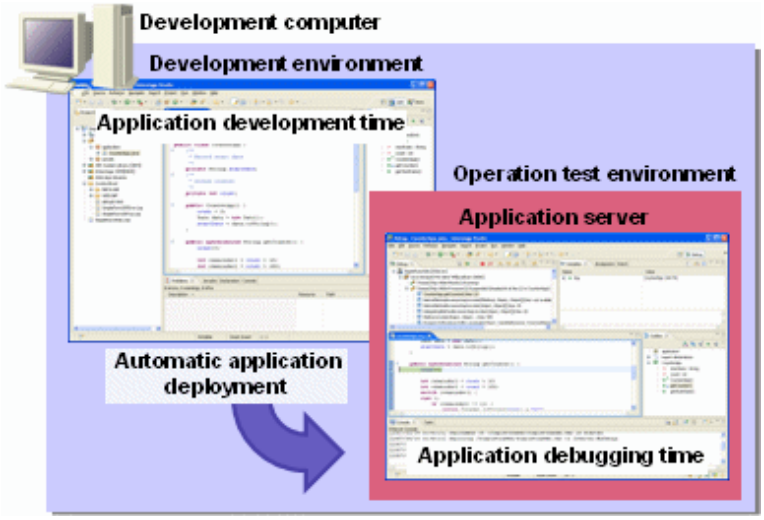
Chapter 2 Features of Interstage Studio

Interstage Studio is a component-oriented Java integrated development environment that enables development of various types of applications, ranging from applications running on client systems to server applications that operate in a multi-tier architecture (such as EJB).
This chapter explains features of Interstage Studio as a Java integrated development environment.

2.1 Development and Debugging in a Standalone Environment

Through seamless linkage with an application server providing an operation test environment, a series of work steps, from development to debugging, can be easily performed in a standalone environment.

Figure 2.1 Concept of Interstage Studio local debugging



The automatic application deployment function greatly reduces the amount of work required in the phases from development to debugging, thereby increasing the efficiency of development.

2.2 Developing Applications with an Application Framework

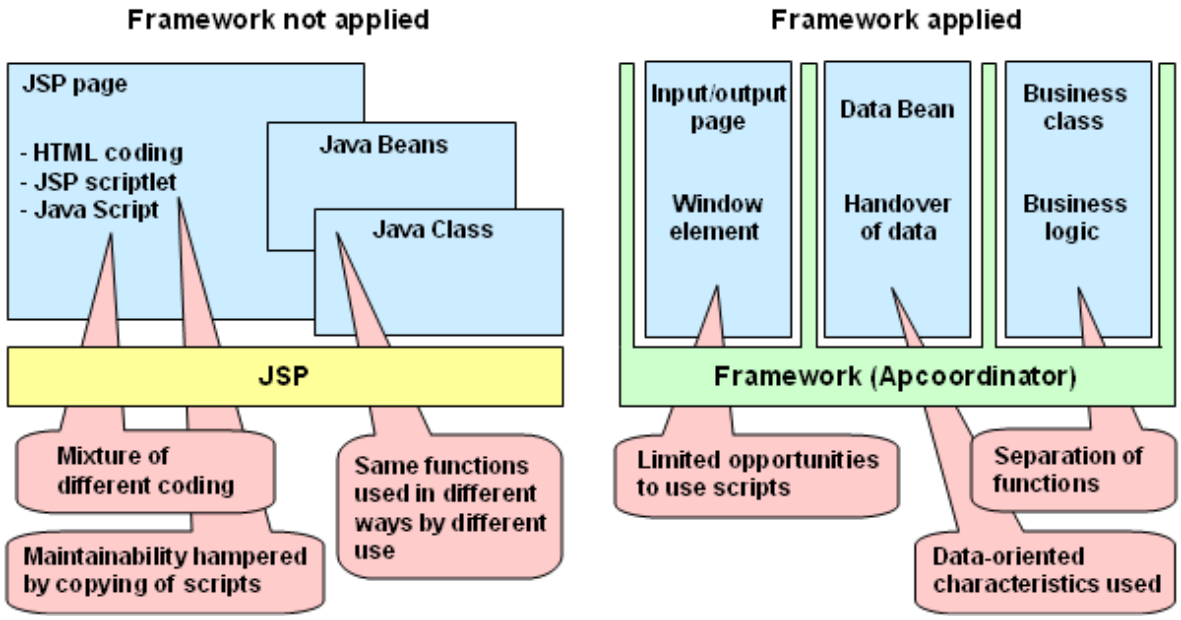
When you create a business application, you must first determine a framework for the application, and then create the application by building on the framework. This framework is called an application framework, or more simply, a framework.
When a framework is used for application development, application elements such as logic and screens are created in the defined form and then inserted into the framework. This process creates the application. The framework definition limits the degree of freedom, but the framework of the application takes shape, enabling you to create easy-to-maintain and easy-to-reuse applications.

Interstage Studio framework

The creation of applications in accordance with the Interstage Studio framework (Apcoordinator) has the following advantages:

- Standardize application structures, thus eliminating programming differences among developers.
- Independently create logical parts, screens, and other application constituents and improve maintainability.
- Easily separate applications into component parts and improve reusability.
- Hide communication-handling elements and other secondary elements by using frameworks and enable their automatic operation.

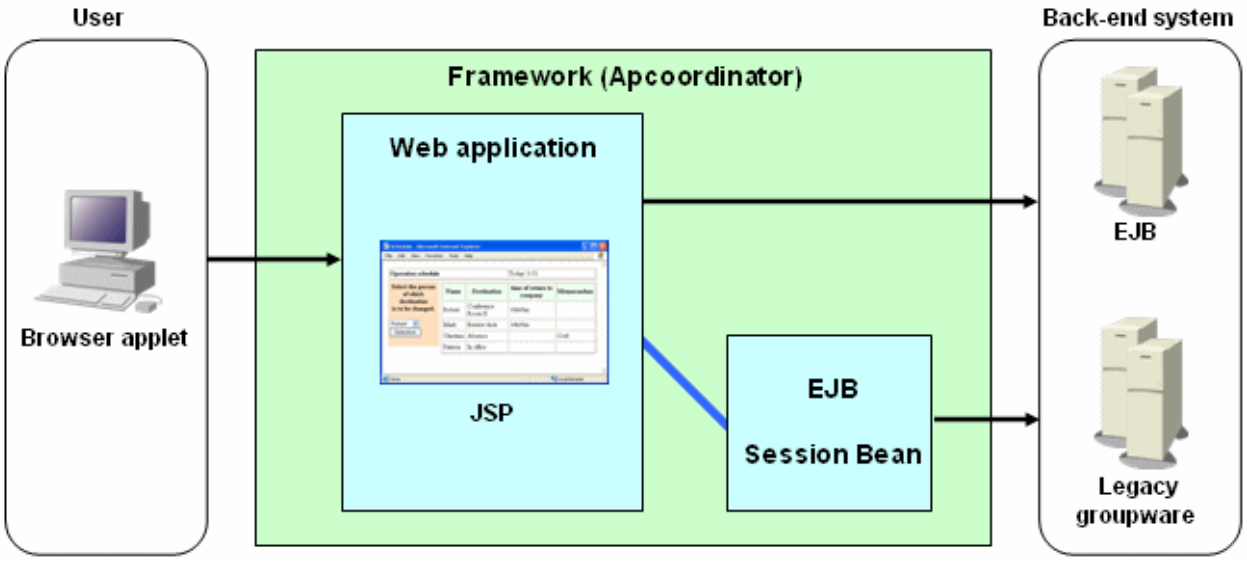
Figure 2.2 Applying the framework to Web applications



Applications supported by the framework

The Interstage Studio framework supports the development of the following types of applications:

- Web applications
  - Apcoordinator provides frameworks for Web applications with JSP (JavaServer Pages) or Servlets. Apcoordinator enables separation and structured configuration of screens and their logical parts.
  - Apcoordinator provides functions for communication with applets.
- EJB
  - Apcoordinator provides frameworks thereby increasing compatibility between EJB and Web front-applications.



For more information about the Interstage Studio framework and information about how to create an application using the Interstage Studio framework, refer to the "Apcoordinator User's Guide", which is an online manual for frameworks.

2.3 Various Tools Required in Application Development

The various design tools and editors listed below are integrated into Interstage Studio. The automatic Java Form creation function, automatic source code generation function, and other functions are provided in the form of wizards, to increase productivity in application development.

- Java Form Designer
- Editor
  - Java Editor
  - HTML/JSP/CSS Editor
  - JavaScript Editor
  - XML Editor
  - DTD Editor
  - XML Schema Editor
  - CMP Extension Information File Editor
  - Source Editor
- Debugger

Note

For details on the various design tools, refer to "Chapter 3 Functions of Interstage Studio", "3.2 Design Tools".

2.4 Components for Business Application Development

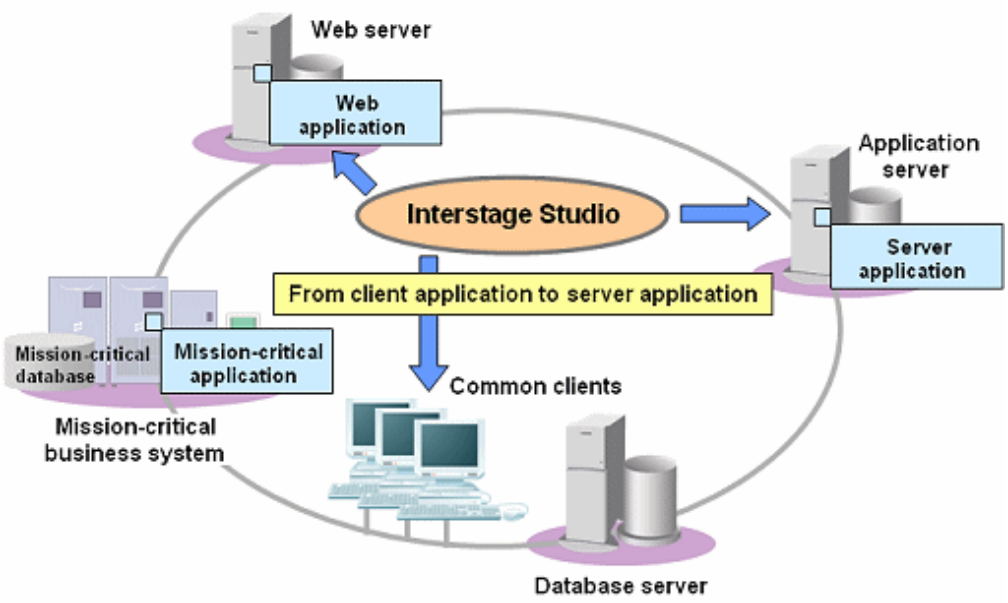
Interstage Studio provides a Java class library (J Business Kit) that is required for business application development. Applications can be quickly developed by combining these components.

2.5 Linkage with Various Servers

Server applications and client applications can be developed with Interstage Studio for diverse patterns of operation in linkage with various servers. The user can develop systems for linkage with the following servers:

- Web server
- Application server
- Database server

Figure 2.3 Overview of linkage with servers



Chapter 3 Functions of Interstage Studio

Interstage Studio provides the Java integrated development environment as development environment (workbench) that can be used for efficient development of applications. Various design tools are integrated into this development environment. These tools help users to efficiently perform a series of application development work steps, from screen design to application debugging.

This chapter explains features of the development environment provided by Interstage Studio and outlines the various design tools.

3.1 Development Environment

Interstage Studio provides the workbench that adopts Eclipse, which is one of the world-famous open-source development environments. Workbench usage enables the development of various types of applications, including Java applications and Java EE applications. This section explains the development environment features of workbench.

Integration of various design tools

Various applications can be efficiently created by using various wizards or design tools.

Advanced edit functions

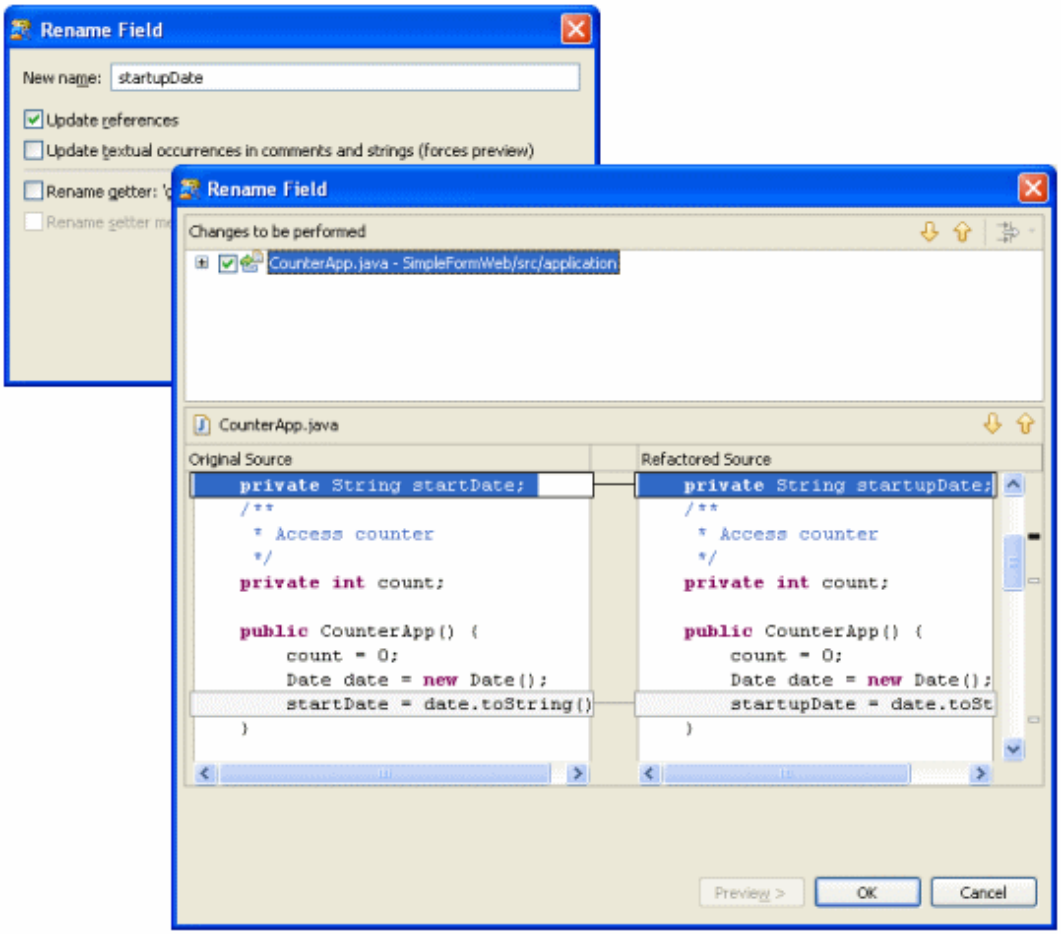
The typical ones are described below.

Refactoring

This function collectively makes changes in common code that is dispersed in different files. For example, the following changes can be made collectively without fail:

- Changing an element name (such as a class name or method name)
  All files with the element name written in them are collectively changed.
- Changing the order of arguments of a method
  All files that contain the arguments of the method are collectively changed so that the code change is matched in all the files.
- Moving a method to another class
  All files that contain this method invocation are collectively changed.

Figure 3.1 Refactoring (renaming) example

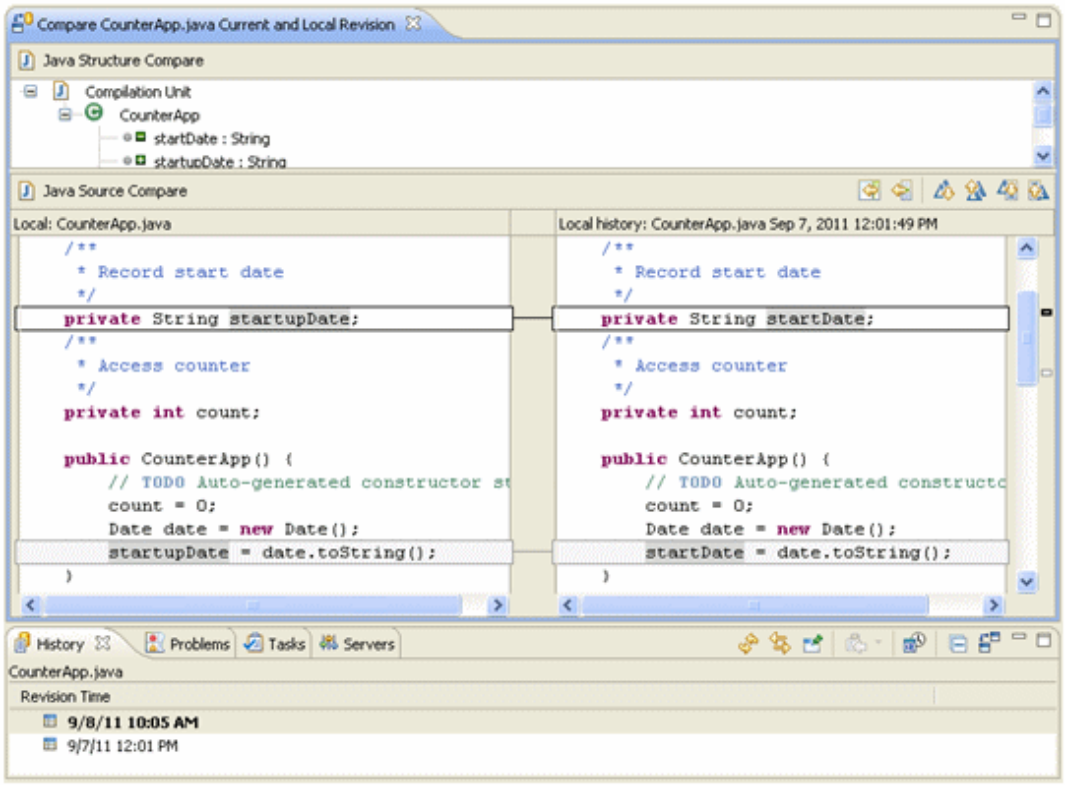


Local history

When a file is saved after editing, the location of each change that was made is recorded in a local history (change history). The local history enables the following operations:

- The contents of the current file can be compared with the older ones, and any differences can be displayed in a way that makes them easy to understand.
- The contents before editing can be restored based on a local history.

Figure 3.2 Example of a local history (comparison)



Code assist

To reduce users' editing workloads, Interstage Studio provides an advanced input support function that completes the Java code you are typing. At the start of tag or keyword entry, the code assist displays a popup list containing candidate tags, properties, and methods that are supported with associated keywords, so that the user can make a selection, as necessary.

Editing XML documents

The following editors are provided for efficient editing of XML documents:

- XML editor, DTD editor, and XML Schema editor

Each editor provides functions to reduce the amount of editing work by users:

- Edit functions in tree view and text view
- Tag insertion function using a DTD or XML schema definition

The XML files that can be edited include Deployment Descriptor, WSDL files, and Web server setting files.

3.2 Design Tools

This section provides an overview of each design tool that is integrated into workbench.

- Java Form Designer
- Editor
- Debugger

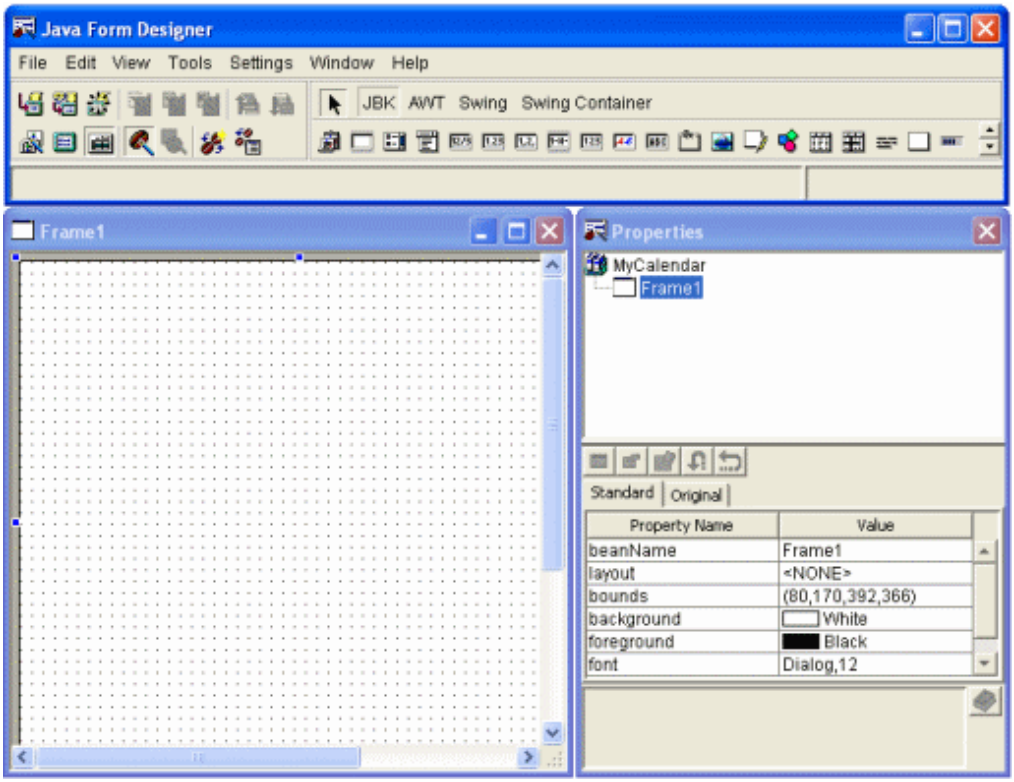
3.2.1 Java Form Designer

Java Form Designer is an editor that helps you edit Java Forms, applets, and JavaBeans. To use a Java Form, the following must be defined. For details, refer to "Interstage Studio User's Guide" > "Developing Java Applications".

- Java Form size

- Layout of controls and Beans contained in the Java Form
- Procedures used in response to different events

Figure 3.3 Java form definition for a Java application



3.2.2 Editors

Interstage Studio provides editors that reduce the amount of user editing for the efficient editing of the different types of development property. The following are the typical editors provided by Interstage Studio:

- Java editor

  This editor is used to code and edit source programs written in Java.

- Editors for Web application files

  The following editors are provided for creating, designing, and editing HTML, JSP, CSS, and JavaScript files:

  - HTML editor, JSP editor, CSS editor, JavaScript editor, and Web page editor

- Editors for XML documents

  The following editors are provided to enable efficient editing of XML documents, which include WSDL files:

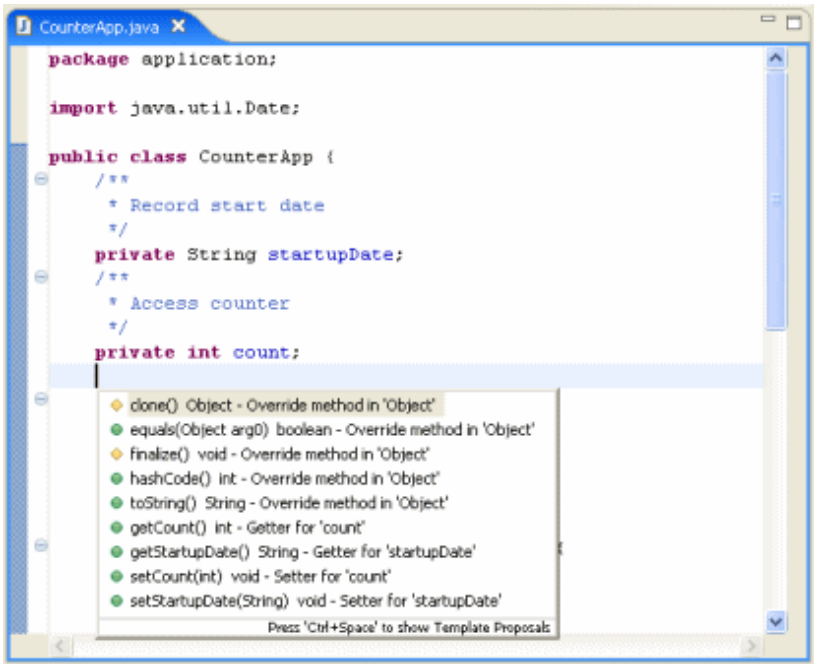
  - XML editor, DTD editor, and XML Schema editor

  Each editor provides edit functions in a tree view or text view.

Java editor

The Java editor provides advanced editing functions, such as the code assist function, which completes the Java code you are typing. The following shows an example of a Java editor.

Figure 3.4 Java editor



HTML editor/JSP editor

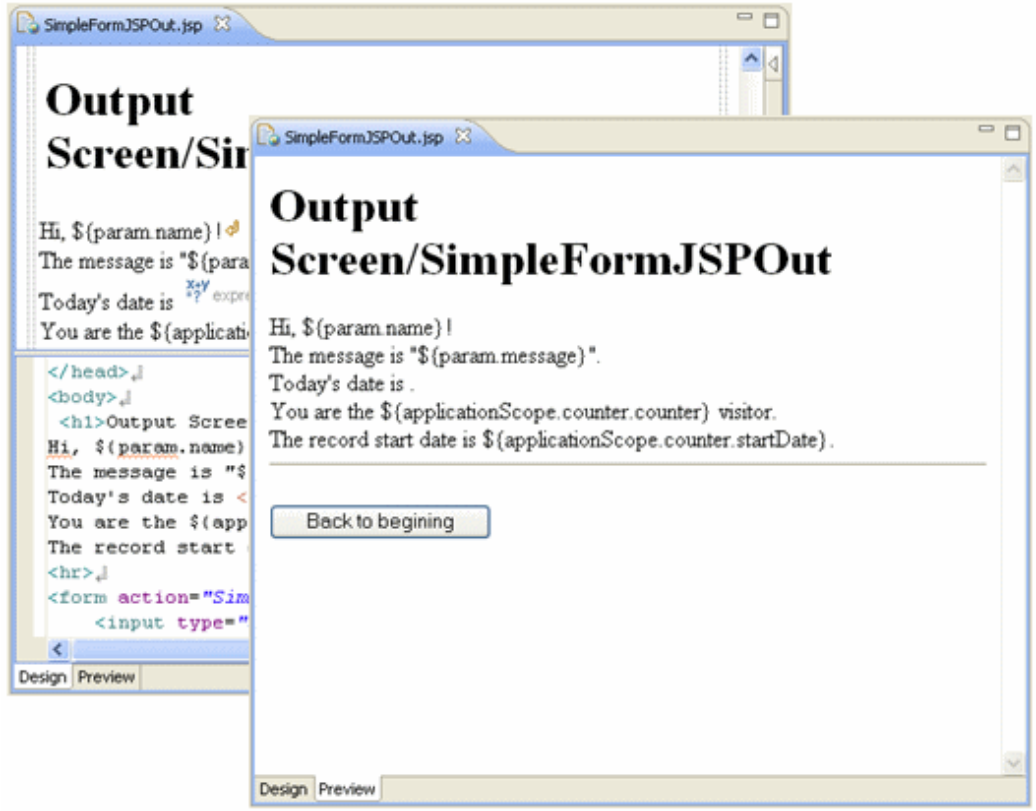
Use the HTML editor and JSP editor to edit the page layout of HTML files and JSP files.
The HTML editor and JSP editor have the following functions:

- Highlighted display of syntax
- Problem identification
- Contents Assist
- Tool tips display
- User-definable templates and snippets
- Tag selection

Using a Web page editor as the editor enables the design and layout to be checked during graphical editing.

The following shows an example of the Web page editor.

Figure 3.5 Web page editor



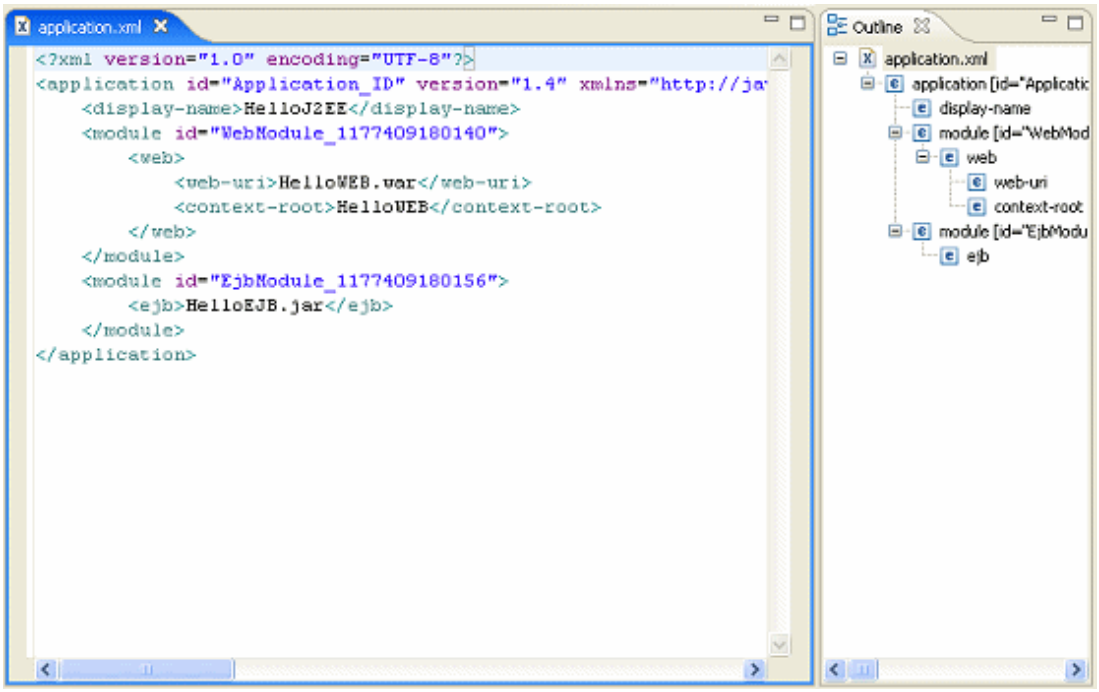
XML editor

The XML editor has the following functions:

- Display of the structure of an XML document in the XML document hierarchal view
- Insertion of elements into the DTD or XMLSchema definition of an XML document

The following shows an example of the XML editor.

Figure 3.6 XML editor



3.2.3 Debugger

Logical errors in the processing code can be detected with the debugger by executing a file registered in a project. The debugger includes the following functions:

- Setting breakpoints
- Displaying and changing data
- Displaying breakpoints
- Threads
- Setting field monitoring points
- Concurrent debugging of multiple applications

Chapter 4 Supported Server Linkages

This chapter explains the server linkages supported by the applications that can be developed with Interstage Studio.

4.1 Linkage with a Web Server

You can develop applications that operate in linkage with Interstage HTTP Server and Interstage HTTP Server 2.2.

A Web application runs on a Web server and uses a Web browser as a client. It can generate dynamic Web pages by using servlets or JSP.

- Servlet

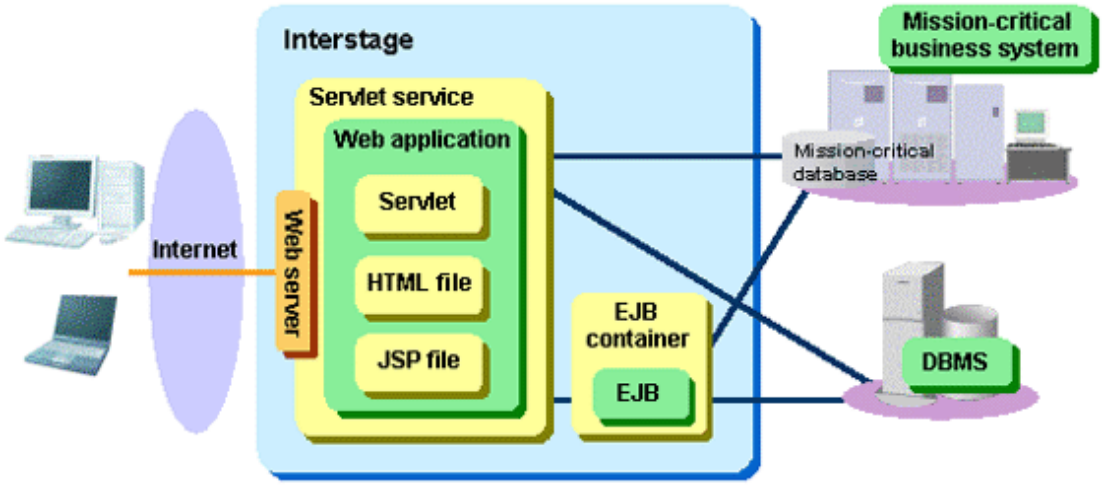
  Java program executed on a Java VM on a server

- JSP

  Web page configured by incorporating scripts or actions into an HTML-format file

The figure below shows the configuration of a Web application.

Figure 4.1 Web application configuration

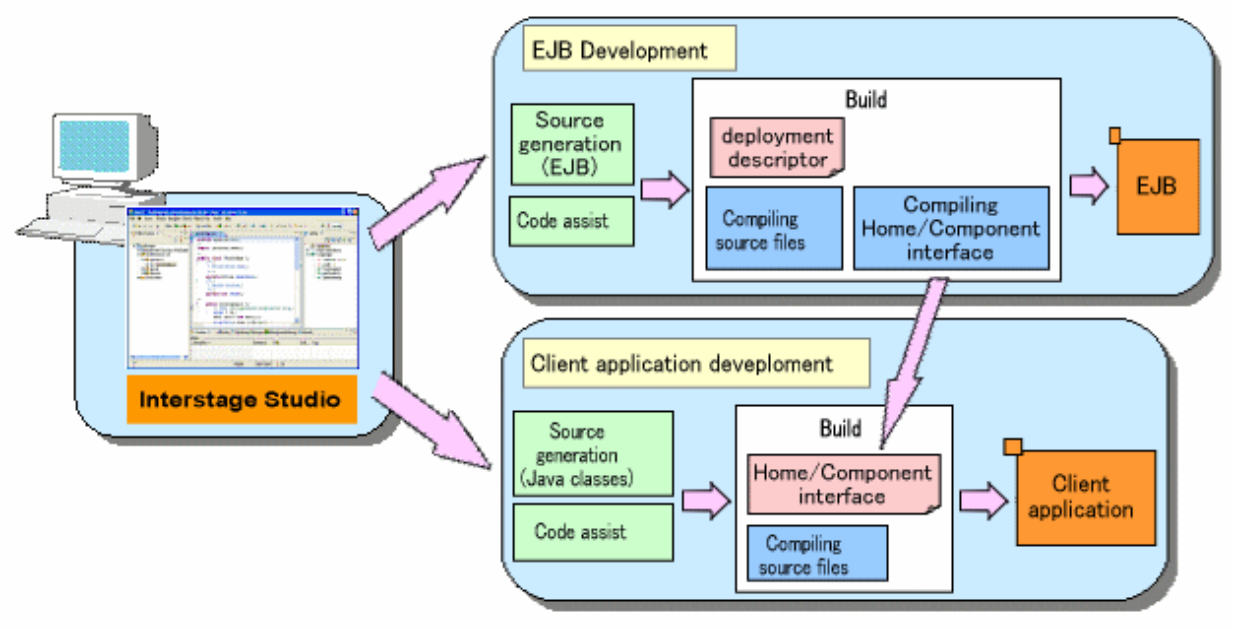


4.2 Linkage with an Application Server

Applications can be developed for operation on application servers using EJB, Web service, and CORBA technology, which are the industry standard interface for distributed application communication. Client applications can also be developed for operation in linkage with the above applications. Since these applications can be developed seamlessly, distributed objects and applications can be efficiently developed.

The figure below shows the overview of EJB development.

Figure 4.2 EJB development overview



4.3 Linkage with a Database Server

Applications can be developed for operation in linkage with database servers such as Oracle and Symfoware. A wizard can be used to automatically generate Java source codes for database access:

- The Connect or Disconnect method for a database can be created through the JDBC driver.
- The Search, Insert, Delete, or Update SQL statement can be created through interactive operation of a wizard.

Note

Use the Java Persistence API (JPA) when developing Java EE applications.

Chapter 5 Supported Components

This chapter explains, in detail, components that can be developed with Interstage Studio, in sections organized by application type.

5.1 Java Applications

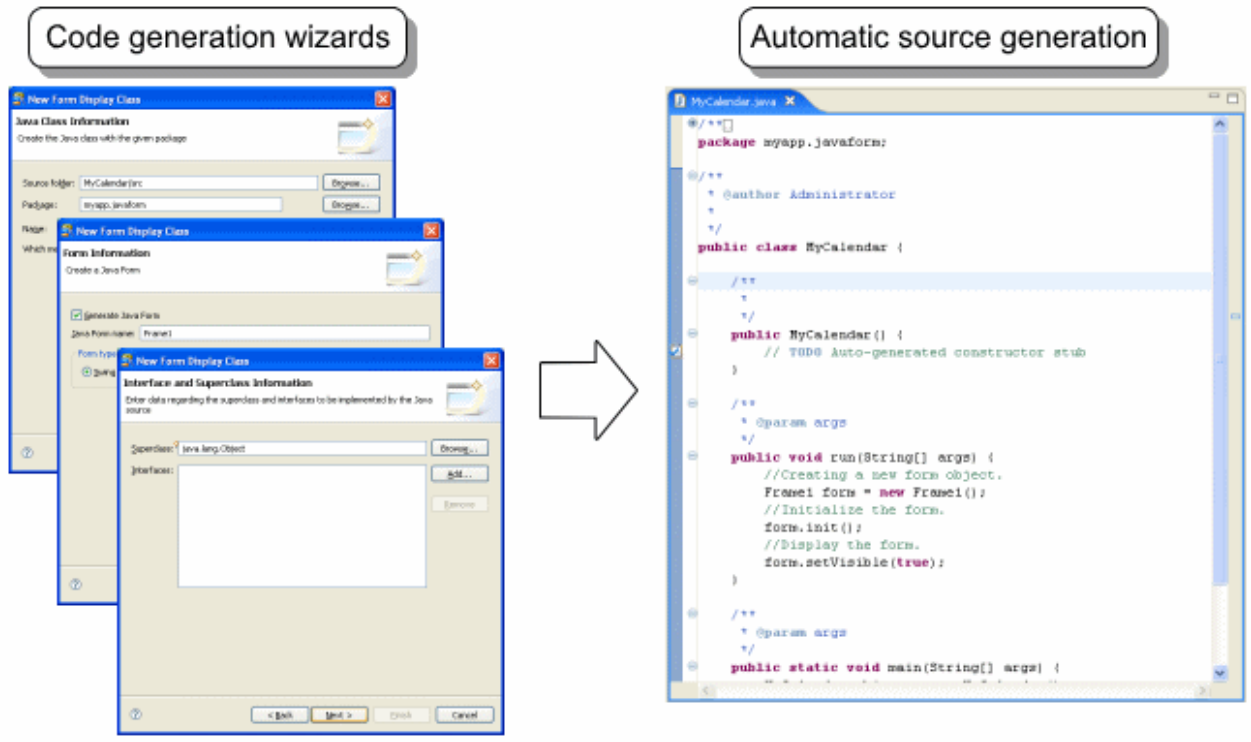
Interstage Studio can be used to develop the following Java applications:

5.1.1 Java Applications

Java applications are applications developed with Java and can run on Java compatible computers independently from platforms.

By specifying the method and the name of the class to inherit from, the automatic source generation wizard creates a template for Java source code. Then, using Java class parts provided by Interstage Studio, add specific processing procedures to this template source code when developing an application. Also, for developing applications that operate on the client side, you can create and use Java Forms on which JavaBeans, which are supplied by Interstage Studio, can be pasted.

Figure 5.1 Development of Java Applications



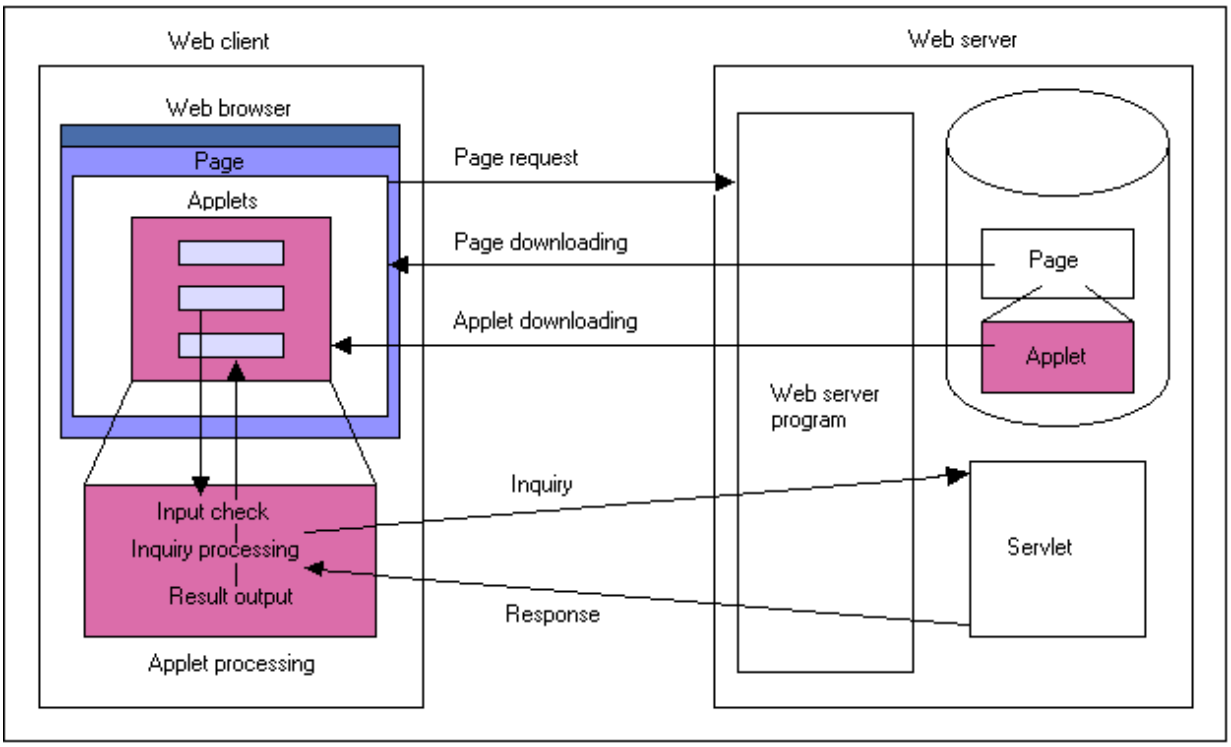
5.1.2 Applets

An applet is a program usually sent to a Web browser as part of an HTML file. They are usually downloaded from a Web server and run inside Web browsers. Applications created as applets can be collectively managed with a server.

Applets must be developed using Java. Applet samples can be created easily using the creation function provided in Interstage Studio. An applet GUI can be created by attaching a Bean (for example, edit function) to the sample created by using a Java form designer. Frequently used functions in the screen transition, inter-applet data linkage, and applet-servlet data linkage are provided as libraries.

Applets can efficiently be developed in Interstage Studio using these functions.

Figure 5.2 Development of Applets



5.1.3 JavaBeans

JavaBeans is the rule for creating Java class components, and is also the general name of a component. Each component is called a bean. A bean can be used with GUI. JavaBeans is a component model that has concepts such as properties, methods, and events.

Creating a Java class as a bean (or a component) enables us not only to reuse them, but also to develop applications with GUI. Thus, by using the Java forms supplied by Interstage Studio, you can paste a bean to a Java form or other container bean and set the property of a bean with GUI.

Beans must be developed using Java. Interstage Studio supports the bean sample creation function to easily create a bean sample. A new bean can be created based on an existing bean. For example, a new text-style bean can be created by using an existing text-style bean and by executing a specific error checking function. Registered in an object palette for Java form designer, the created bean can be used as a component like the ones provided by other systems. In this way, you can develop and reuse beans efficiently in Interstage Studio.

5.2 Web Applications

Web applications are application programs that are run on a Web server so that a Web browser can be used as a client.

A Web application screen displayed on the Web browser is called a Web page. On a Web page, data can be entered in the HTML form. You can display the processed result on the next Web page, or can display the linked Web page. Moreover, applets and scripts, which run on a client, can be built into a Web page.

The Web application development function of Interstage Studio supports construction of a Web site that consists of multiple Web pages (servlet, JSP and HTML pages). This section explains servlets and JSP.

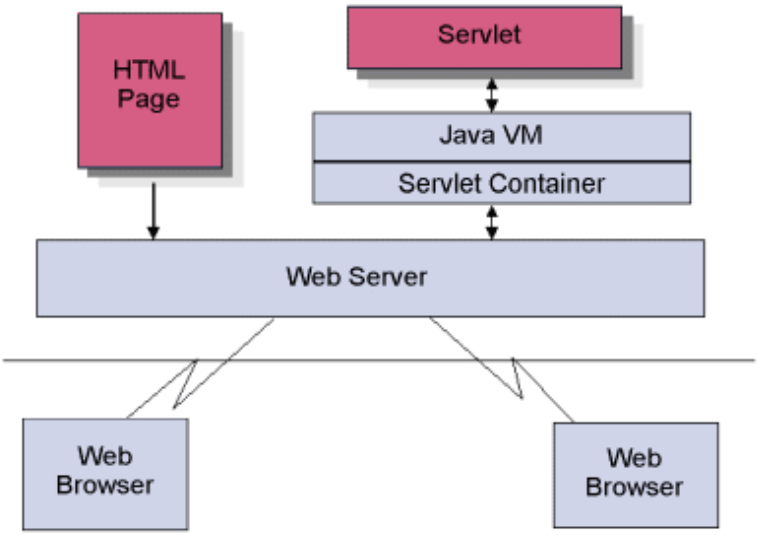
5.2.1 Servlets

Servlets are Java programs that are activated on a Web server, and run through Java VM. A servlet receives from a browser the data that was entered on one Web page, configures processing results as a new Web page, then outputs the new Web page to the browser.

The Web application development function uses Servlet 2.5 (Java EE 5), Servlet 3.0 (Java EE 6), or Servlet 2.4 (J2EE) as the servlet API executed by the servlet container. The servlet engine is a program that controls the processing between a Web server and a servlet. The servlet container loads a servlet program onto Java VM and calls the servlet API according to the access from the Web browser.

The figure below shows the configuration of a Web application that uses servlets.

Figure 5.3 Configuration of Web Application



5.2.2 JSP

JSP (JavaServer Pages) are Web pages with dynamic contents constructed by including scripts and actions in files written in HTML format. Like a servlet page, a JSP page receives from a browser the data that was entered on one Web page, configures a new Web page, and then outputs this to the browser. Through script processing, results executed by other Java classes and JavaBeans on the server can be output.

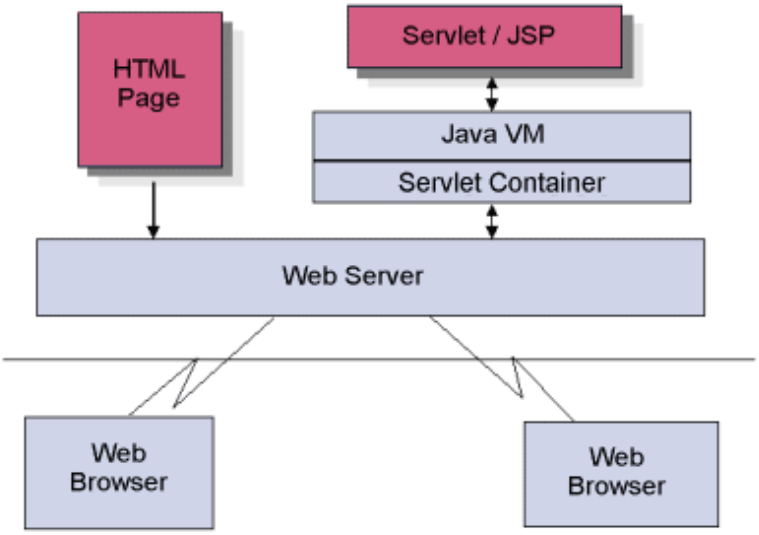
JSP page differ from servlet pages in that, whereas a servlet page is deployed on the Web server after being compiled as a servlet class, a JSP page does not need to be compiled as it run as a server function.

Point

The function that enables JSP pages to run on Web servers is known as the "JSP engine" or "JSP container". Following access through a Web browser, the JSP engine automatically converts the JSP page into a servlet program, compiles it, and runs process described in the page using a servlet engine.

In the Web application development function, JSP 2.1 (Java EE 5), JSP 2.2 (Java EE 6), or JSP 2.0 (J2EE) is used as the servlet container.

Figure 5.4 Configuration of a Web Application using JSP



5.3 Enterprise JavaBeans

Enterprise JavaBeans (EJB) is a server component for decentralized object-oriented processing. Interstage Studio can be used to develop Enterprise Beans (EB) using Java.
Covering the following low-level API and EB can create server-side program components by programming business logic processing alone:

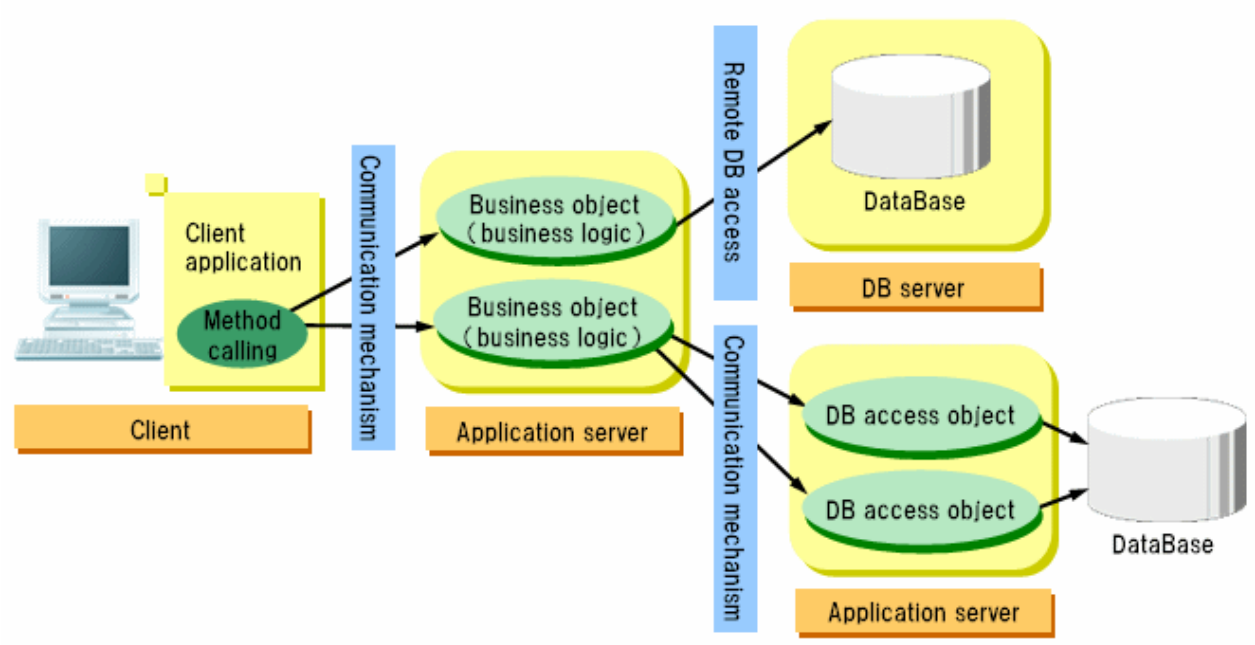
- Life cycle management of the components
- Transaction management
- Database processing

Interstage Studio can be used to develop the following using Java:

- EB (server-side components)
- Client applications that use EB

The EB execution environment covers over the differences in transaction processing based on the TP monitor and DBMS. Therefore, you can develop server program components that are independent from the platform to be used.

Figure 5.5 Development of EJB



5.4 Enterprise Applications

With the Java EE/J2EE platform, you have the technology to create multi-level applications that combine components that are easy to re-use, using component functions, and the various services and communication methods.

Enterprise JavaBeans, Web applications, Java EE application clients, and J2EE application clients can be created with Interstage Studio. In addition, they can be made into EAR files by creating them as enterprise applications.

5.5 Web Service Applications

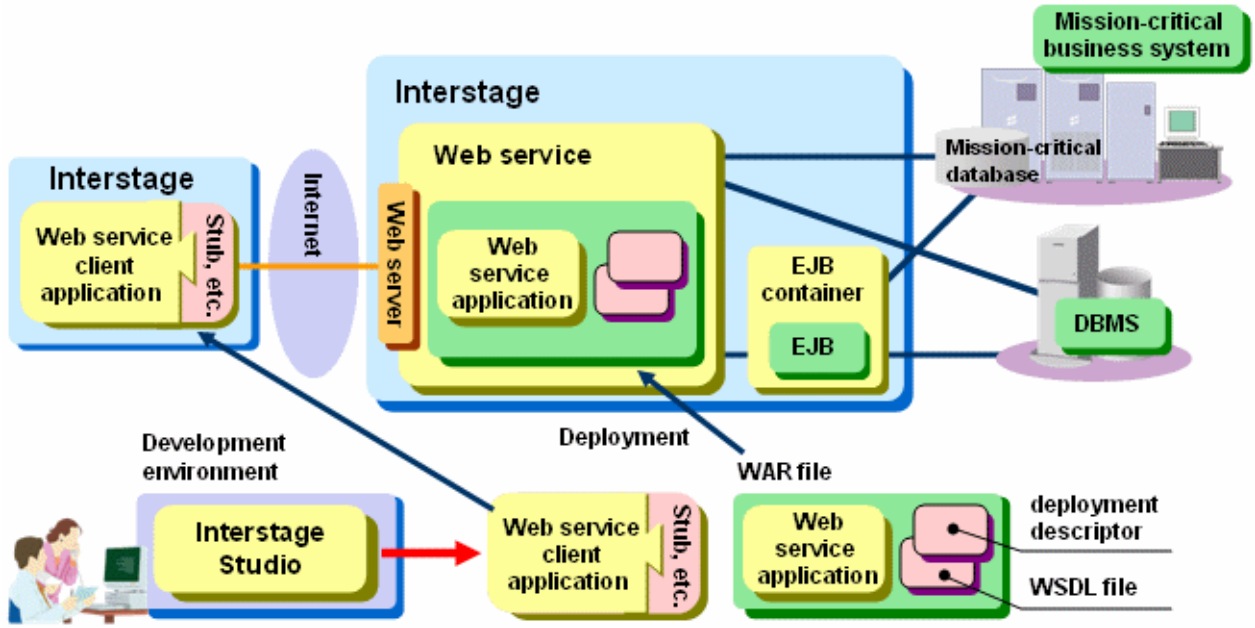
Web services are technologies used to make system functions publicly available as services over networks, and they have been attracting public attention as one method of leveraging existing systems and promoting reuse of software.

For those purposes, Web services make possible platform-independent remote procedure calls for applications by using WSDL to define interfaces and using SOAP for communication with HTTP as the underlying protocol.

The following applications, which use Interstage Application Server as an operation environment, can be developed with Interstage Studio:

- Web service applications
  Applications that make their services publicly available on networks
- Web service client applications
  Applications that call services that are publicly available on networks

Figure 5.6 Configuration of a Web service application



The files necessary for developing Web service applications, such as WSDL and deployment descriptor files, can be created from the service endpoint interface. The files necessary for developing Web service client applications, such as a service endpoint interface and stub files, can be created from the WSDL file.

5.6 Development of Components

Any processing can be coded as a package by defining a class in Java. A created class can be used by other programs (classes). Moreover, a new class based on an existing class can be created by using a technique called inheritance. This new class maintains the functions of the original class. This means that any file that has the file extension .class and was created with Java can be treated as a component.

The procedure for Java class component development with Interstage Studio is similar to that described in "5.1.1 Java Applications". Developed class components are stored at a location such that they can be referred to by their development team. By setting the component storage location in CLASSPATH, users of these components can efficiently code method calls by using the input template function of the Java editor.

Appendix A Online Manuals

Various online manuals are provided with this product. Refer to the online manuals as required.

Manual name	Manual overview
Interstage Studio Release Notes	Provides an overview of the additional functions of Interstage Studio and explains compatibility.
Interstage Studio Installation Guide	Explains how to install and uninstall Interstage Studio.
Interstage Studio General Description	Provides an overview of Interstage Studio and explains key concepts.
Interstage Studio User's Guide	Explains development methods and operation procedures and provides development-related notes for workbench functions.
Java Development Kit Documentation	Provides the information required to develop applications that use Java.
J Business Kit Online Manual	Explains how to develop applications that use business-oriented libraries (GUI, multi-media, etc.).
Framework Online Manual	Explains how to use Framework to develop applications.
Application Server Online Manual	Provides general descriptions of the Interstage Application Server operation methods.

Index