FUJITSU Software
Interstage Information Integrator
V11.0.0

System Design Guide

Windows

B1X1-0293-01ENZ0(00)
August 2013

Preface

This document deals with the following products:

- Interstage Information Integrator Standard Edition V11.0.0

Additionally, the optional products listed below are also included.

- Interstage Information Integrator Agent V11.0.0

Purpose of This Document

This document explains the outline of features and design method and system configuration that provided by Interstage Information Integrator (hereafter abbreviated as "Information Integrator").

Intended Readers

This document is intended for the following readers:

- Users who design the operational environment of the Information Integrator

The following knowledge is required to read this document:

- General knowledge about operating systems
- General knowledge about the internet
- General knowledge about database functions and how to use them
- General knowledge about public clouds
- General knowledge about forms

Structure of This Document

Chapter 1: Information Integrator Features

This chapter presents an overview of the features provided in this product.

Chapter 2: Designing the Process Flow

This chapter explains the designing the Process Flow of the collection, conversion, and delivery, etc. done with Information Integrator.

Chapter 3: Designing Using III Studio

This chapter explains the design procedure of the Process Flow using III Studio.

Chapter A: Function list that can be various use it

This appendix explains the functions of each communication protocol that can be used and the function of each database that to be extracted and to be stored and that can be used are shown by the list.

Positioning of This Document

FUJITSU Software Interstage Information Integrator Structure and Overview of Manuals

This document lists the manual structure for Information Integrator, as well as which manuals should be referred to for different user purposes and objectives.

FUJITSU Software Interstage Information Integrator System Design Guide [this document]

This document presents an overview of Information Integrator functions, as well as information about system configurations and design issues.

FUJITSU Software Interstage Information Integrator Setup Guide

This document explains how to create an environment after Information Integrator has been installed.

FUJITSU Software Interstage Information Integrator Operation Guide

This document explains how to operate Information Integrator.

FUJITSU Software Interstage Information Integrator Command Reference

This document explains the commands provided by Information Integrator.

FUJITSU Software Interstage Information Integrator Transfer Accelerator Setup Guide

This document explains the outline, the environmental construction method, and operation procedure of III Transfer Accelerator.

FUJITSU Software Interstage Information Integrator Message Guide

This document lists the messages output by Information Integrator.

FUJITSU Software Interstage Information Integrator Glossary

This document is a glossary of the terms used in the Information Integrator manuals.

FUJITSU Software Symfoware Server Message Guide

This guide explains lists the messages output by Symfoware Server.

Conventions Used in This Document

Manual titles

The manual titles listed in this document are abbreviated as below.

Information Integrator manual

Abbreviation	Official title
Structure and Overview of Manuals	FUJITSU Software Interstage Information Integrator Structure and Overview of Manuals
System Design Guide (This document)	FUJITSU Software Interstage Information Integrator System Design Guide
Setup Guide	FUJITSU Software Interstage Information Integrator Setup Guide
Operation Guide	FUJITSU Software Interstage Information Integrator Operation Guide
Command Reference	FUJITSU Software Interstage Information Integrator Command Reference
Transfer Accelerator Setup Guide	FUJITSU Software Interstage Information Integrator Transfer Accelerator Setup Guide
Message Guide	FUJITSU Software Interstage Information Integrator Message Guide
Glossary	FUJITSU Software Interstage Information Integrator Glossary

Related manual

Abbreviation	Official title
Symfoware Message Guide	FUJITSU Software Symfoware Server Message Guide

Abbreviations

The product names listed in this document are abbreviated as below.

Note that system names and product names in this document are not followed by trademark symbols such as ä or â.

Abbreviation	Official title
Windows	Microsoft® Windows Server® 2012 Foundation
	Microsoft® Windows Server® 2012 Essentials
	Microsoft® Windows Server® 2012 Standard
	Microsoft® Windows Server® 2012 Datacenter
	Microsoft Windows Server 2008 Foundation
	Microsoft Windows Server 2008 Standard

Abbreviation	Official title
	Microsoft Windows Server 2008 Enterprise
	Microsoft Windows Server 2008 Datacenter
	Microsoft Windows Server 2008 Standard without Hyper-V
	Microsoft Windows Server 2008 Enterprise without Hyper-V
	Microsoft Windows Server 2008 Datacenter without Hyper-V
	Microsoft Windows Server 2008 R2 Foundation
	Microsoft Windows Server 2008 R2 Standard
	Microsoft Windows Server 2008 R2 Enterprise
	Microsoft Windows Server 2008 R2 Datacenter
	Microsoft Windows Server 2003, Standard Edition
	Microsoft Windows Server 2003, Enterprise Edition
	Microsoft Windows Server 2003, Datacenter Edition
	Microsoft Windows Server 2003, Standard x64 Edition
	Microsoft Windows Server 2003, Enterprise x64 Edition
	Microsoft Windows Server 2003, Datacenter x64 Edition
	Microsoft Windows Server 2003 R2, Standard Edition
	Microsoft Windows Server 2003 R2, Enterprise Edition
	Microsoft Windows Server 2003 R2, Datacenter Edition
	Microsoft Windows Server 2003 R2, Standard x64 Edition
	Microsoft Windows Server 2003 R2, Enterprise x64 Edition
	Microsoft Windows Server 2003 R2, Datacenter x64 Edition
	Windows®8
	Windows®8 Pro
	Windows®8 Enterprise
	Windows®7 Home Premium
	Windows®7 Professional
	Windows®7 Enterprise
	Windows®7 Ultimate
	Windows® Vista Home Basic
	Windows® Vista Home Premium
	Windows® Vista Business
	Windows® Vista Ultimate
	Windows® Vista Enterprise
	Windows® XP Home Edition
	Windows® XP Professional
	Windows® XP
Windows XP	Windows® XP Home Edition
	Windows® XP Professional
Windows Vista	Windows® Vista Home Basic

Abbreviation	Official title
	Windows® Vista Home Premium
	Windows® Vista Business
	Windows® Vista Ultimate
	Windows® Vista Enterprise
Windows 7	Windows®7 Home Premium
	Windows®7 Professional
	Windows®7 Enterprise
	Windows®7 Ultimate
Windows 8	Windows®8
	Windows®8 Pro
	Windows®8 Enterprise
Windows Server 2003	Microsoft Windows Server 2003, Standard Edition
	Microsoft Windows Server 2003, Enterprise Edition
	Microsoft Windows Server 2003, Datacenter Edition
	Microsoft Windows Server 2003, Standard x64 Edition
	Microsoft Windows Server 2003, Enterprise x64 Edition
	Microsoft Windows Server 2003, Datacenter x64 Edition
	Microsoft Windows Server 2003 R2, Standard Edition
	Microsoft Windows Server 2003 R2, Enterprise Edition
	Microsoft Windows Server 2003 R2, Datacenter Edition
	Microsoft Windows Server 2003 R2, Standard x64 Edition
	Microsoft Windows Server 2003 R2, Enterprise x64 Edition
	Microsoft Windows Server 2003 R2, Datacenter x64 Edition
Windows Server 2008	Microsoft Windows Server 2008 Foundation
	Microsoft Windows Server 2008 Standard
	Microsoft Windows Server 2008 Enterprise
	Microsoft Windows Server 2008 Datacenter
	Microsoft Windows Server 2008 Standard without Hyper-V
	Microsoft Windows Server 2008 Enterprise without Hyper-V
	Microsoft Windows Server 2008 Datacenter without Hyper-V
	Microsoft Windows Server 2008 R2 Foundation
	Microsoft Windows Server 2008 R2 Standard
	Microsoft Windows Server 2008 R2 Enterprise
	Microsoft Windows Server 2008 R2 Datacenter
Windows Server 2012	Microsoft Windows Server 2012 Foundation
	Microsoft Windows Server 2012 Essentials
	Microsoft Windows Server 2012 Standard
	Microsoft Windows Server 2012 Datacenter
Failover Clusters	Windows Server 2008 Failover Clusters

Abbreviation	Official title
	Microsoft Cluster Service
Excel	Microsoft® Excel
Word	Microsoft® Word
SQL Server	Microsoft® SQL Server™
Oracle	Oracle® Enterprise Edition
	Oracle® Database Standard Edition
	Oracle® Database Standard Edition One
	Oracle® Database Enterprise Edition
Solaris	Oracle Solaris
Linux	Red Hat Enterprise Linux
UNIX	Solaris
	Linux
XL-DATA/MV	XL DATA MOVER
	Softek XL DATA MOVER
	ETERNUS SF XL-DATA/MV
	OSIV MSP XL-DATA/MV
Interstage Application Server	Interstage® Application Server Standard-J Edition
	Interstage® Application Server Enterprise Edition
Shunsaku	Interstage® Shunsaku Data Manager Enterprise Edition
Data Effector	Interstage® Data Effector Standard Edition
Charset Manager	Interstage® Charset Manager
Linkexpress	Linkexpress
	Linkexpress Advanced Edition
	Linkexpress Standard Edition
	Linkexpress Enterprise Edition
Symfoware Server	Symfoware® Server Standard Edition
	Symfoware® Server Enterprise Edition
	Symfoware® Server Enterprise Extended Edition
DB2	IBM® DB2 9.7 for Linux, UNIX and Windows, Express Edition
	IBM® DB2 9.7 for Linux, UNIX and Windows, Workgroup Server Edition
	IBM® DB2 9.7 for Linux, UNIX and Windows, Enterprise Server Edition

Note

In this manual, please read **Start** menu of Windows in a different way as **Start** Window or **All Apps** Window when you use Windows 8 or Windows Server 2012.

Export Restriction

If this document is to be exported or provided overseas, confirm legal requirements for the Foreign Exchange and Foreign Trade Act as well as for other laws and regulations including U.S. Export Administration Regulations, and follow any necessary procedures.

Trademarks

Microsoft, Windows Azure, Windows, Windows Vista, Windows Server, and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds.

Red Hat, RPM, and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.

HULFT is a registered trademark of Saison Information Systems Co., Ltd.

Salesforce is a registered trademark of salesforce.com, Inc.

DB2 is a trademark of International Business Machines Corp., registered in many jurisdictions worldwide.

PostgreSQL is a trademark of PostgreSQL in the United States and other countries.

Interstage, Linkexpress, Symfoware, Shunsaku, and Systemwalker are registered trademarks of FUJITSU LIMITED.

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

Issue Date and Version

August 2013

August 2013: First Edition

Copyright

Contents

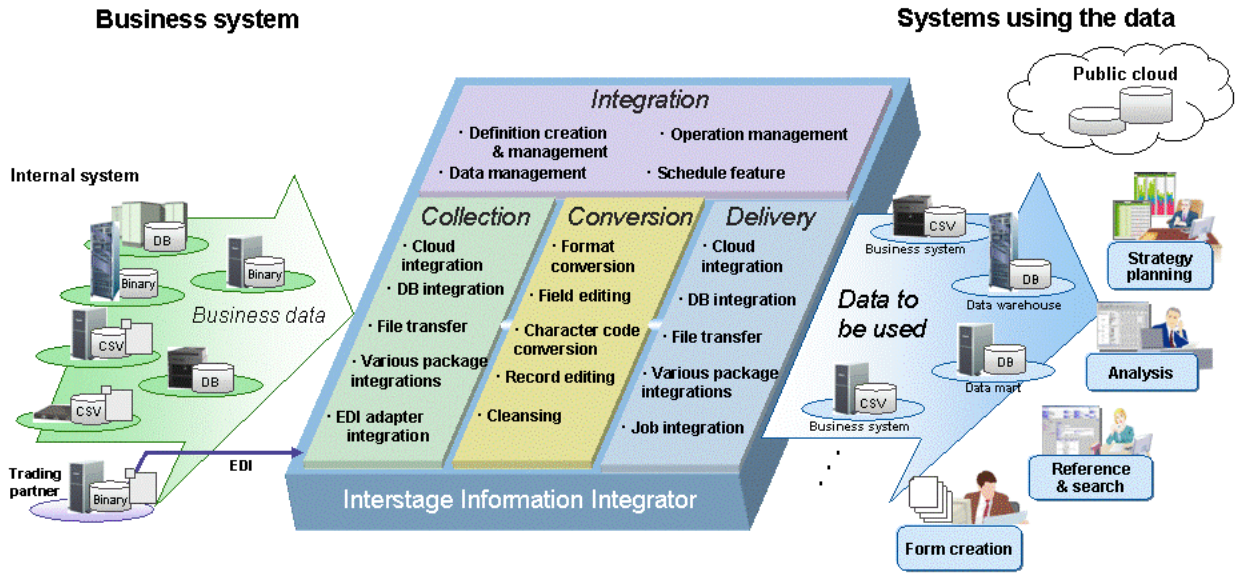
Chapter 1 Information Integrator Features

This chapter describes the features of Information Integrator.

1.1 Feature Outline

Information Integrator is a data integration product that collects and integrates a diverse range of data and delivers it in the required format at the required time. Data is collected from multiple business systems within the corporation comprising differing elements such as platforms, operating times and data management. The data is then integrated in a format suited to the targeted use and stored in the business system planning to use it.

Figure 1.1 Feature outline



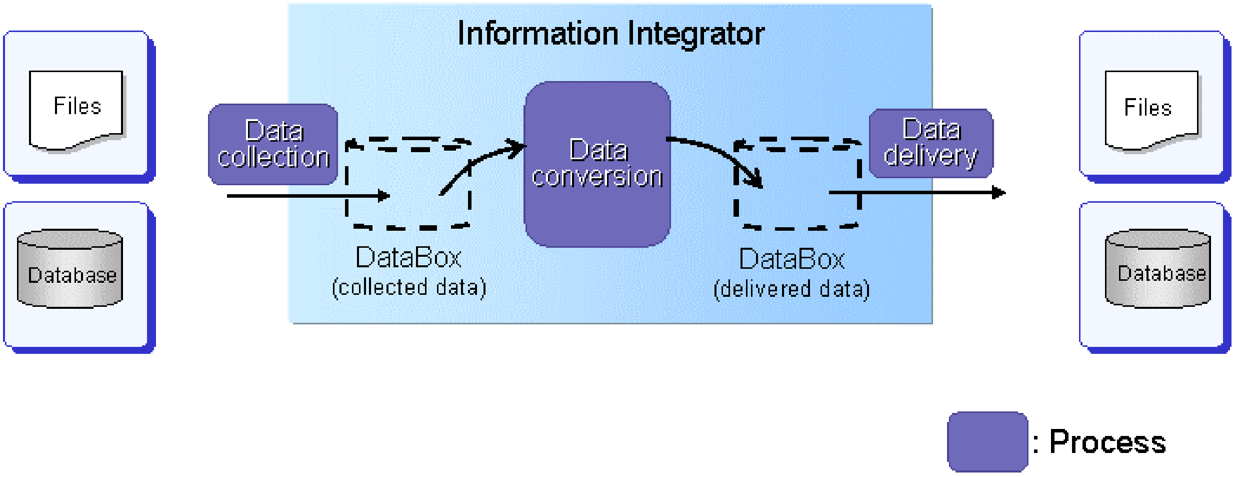
Point

Information Integrator Transfer Accelerator (III Transfer Accelerator)

File transfer of Information Integrator can be done at high speed or more by using III Transfer Accelerator. In addition, the data communication of not only Information Integrator but also the applications that use TCP/IP can be sped up by using III Transfer Accelerator.

Information Integrator executes a series of processes involving data collection, data conversion and data delivery, in the flow outlined below.

Figure 1.2 Process flow



Information Integrator's data management feature stores the collected data temporarily in the DataBox. Various required conversion processes are then performed on the data stored in the DataBox, where the data to be delivered is created. The converted data is then extracted from the DataBox and delivered.

1.2 Data Collection

This section explains the data collection process.

1.2.1 Data Formats that can be Collected by Information Integrator

The data formats that can be collected by Information Integrator are described below.

- File formats
  - CSV files
  - Binary files
  - XML files
- Database formats
  - Symfoware Server
  - Oracle
  - SQL Server
  - DB2
  - PostgreSQL
  - SQL Azure
  - Other databases (ODBC connection)
- Object formats
  - Salesforce objects

    This refers to objects extracted by Salesforce.com, Inc. (hereafter, 'Salesforce').
- BLOB format

  This refers to data collected from Azure Storage.

Point

Compressed data collection

For BLOB data collection, data compressed in zip format is collected and then decompressed using a plug-in provided by Information Integrator, after which it can be used.

Refer to "3.10.3 Compressing or Decompressing Data" for information on plug-ins.

1.2.2 Data Collection Methods

When collecting files, use 'Receive files' to collect files from remote systems, or 'File input' to collect files from your local system.

Use 'Extract from DB' to retrieve data from the database.

When collecting data placed on a public cloud, use 'Extract object' to extract public cloud objects.

Use 'BLOB collection' to retrieve data from Azure Storage.

A description of each of the data collection methods is provided below.

1.2.3 Receive Files

This section describes the 'Receive files' feature.

Use the 'Receive files' feature when collecting files located on remote systems.

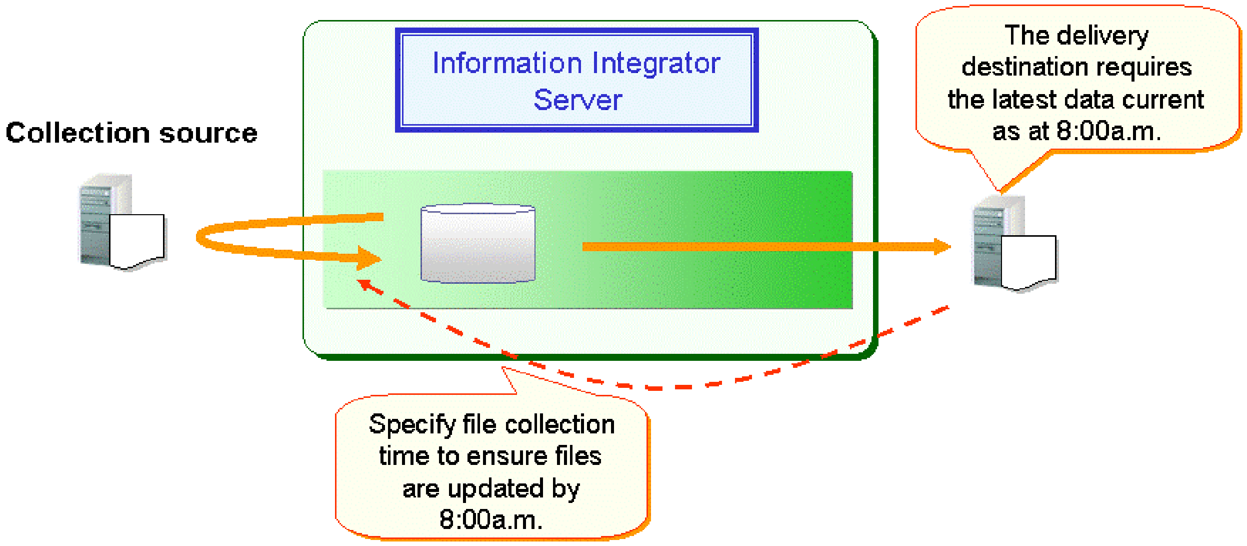
There are two types of 'Receive files' collection methods: 'Receive files (as initiator)' and 'Receive files (as responder)'.

Receive files (as initiator)

This feature collects files located on a business system (collection source) from Information Integrator.

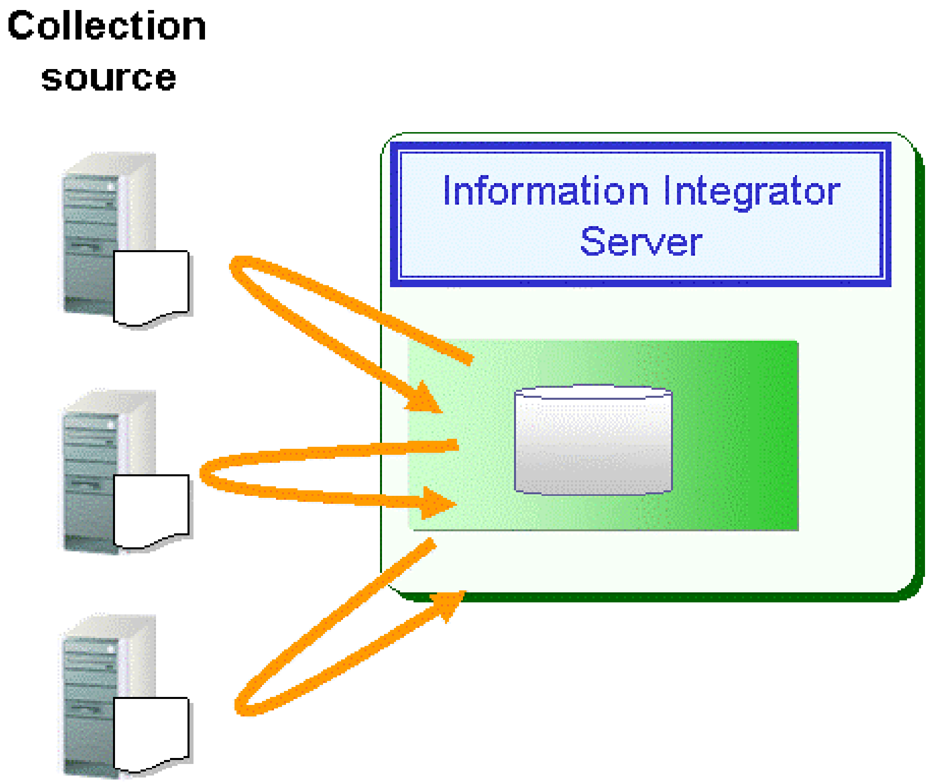
Use this feature when bringing information required fitting the schedule of the system using the information (delivery destination). For example, if the system using the information requires the latest information to be held at a certain point every morning, this feature will calculate the time required for data to reach the delivery destination from 'Receive files', and will specify a schedule for 'Receive files' accordingly.

Figure 1.3 Receive files (as initiator)


Information Integrator Server
Collection source
The delivery destination requires the latest data current as at 8:00a.m.
Specify file collection time to ensure files are updated by 8:00a.m.

The 'Receive files (as initiator)' feature can collect files located on multiple collection sources.

Figure 1.4 Collecting files from multiple collection sources



Receive files (as responder)

Information Integrator monitors files sent from the collection source and confirms they have been sent before executing subsequent processes.

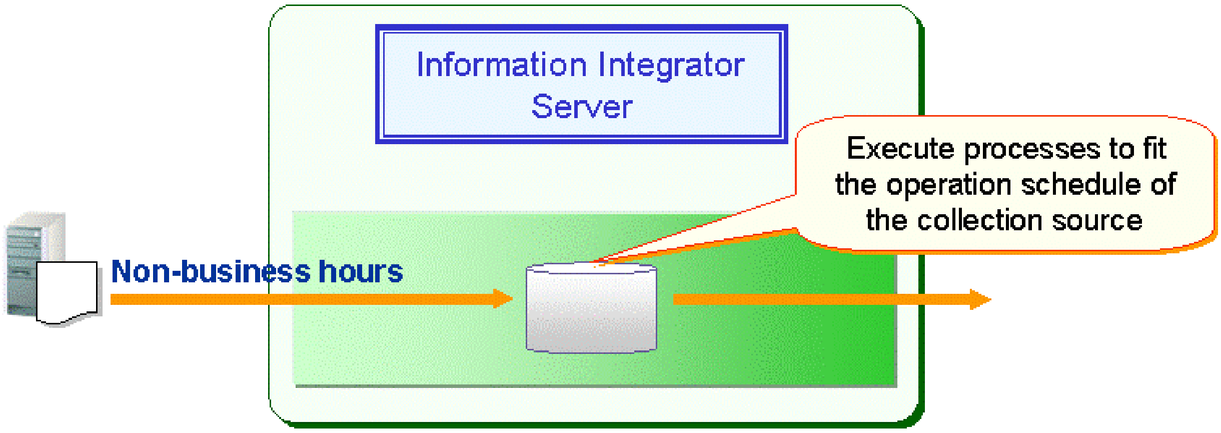
The main feature of 'Receive files (as responder)' is that processes can be executed to fit the operational schedule of the collection source. For example, 'Receive files (as responder)' can be used when waiting for all data sent from multiple collection sources to arrive, or when waiting for multiple files to arrive from a single collection source.

The following kinds of operations can be performed by using 'Receive files (as responder)':

- Execute processes to fit the collection source operation schedule

  Once preparation of the required data is complete at the collection source, a file is sent to Information Integrator. The latest data generated at the collection source can be processed as required.

Figure 1.5 Executing processes to fit the operation schedule of the collection source

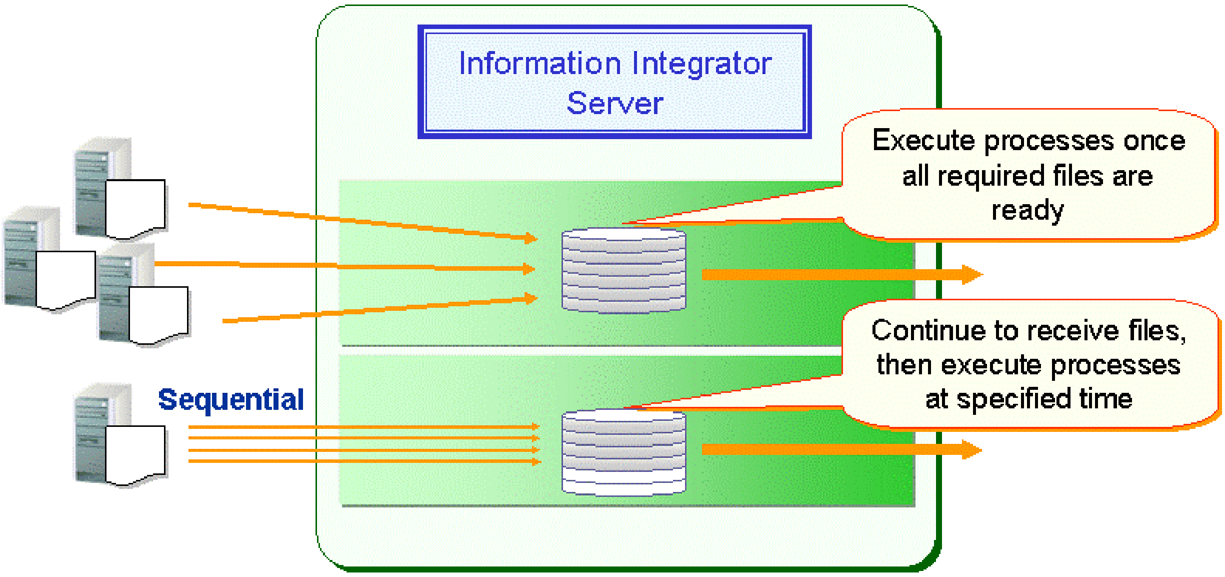

Information Integrator Server
Execute processes to fit the operation schedule of the collection source
Non-business hours

- Execute subsequent processes in accordance with the file receipt status

  Required data is sent as soon as preparation at the collection source is complete, and Information Integrator executes processes to fit its own operation schedule. Information Integrator continues to receive the latest data and can also send data in accordance with the delivery destination operation.

  This feature can also be used when collecting data from multiple branch locations or when collecting files that are joined and used with each other.

  Figure 1.6 Executing subsequent processes in accordance with the file receipt status


Information Integrator Server
Execute processes once all required files are ready
Sequential
Continue to receive files, then execute processes at specified time

1.2.3.1 Optional Features for 'Receive Files'

Automatic compression and decompression

The time required for file transfer is proportionate to the size of the data being transferred. Transfers involving large-scale data or networks with slow connection speeds will require a noticeably longer transfer time.

In such cases, both the volume of data being sent over the network and the time required to transfer it can be reduced by compressing files and then decompressing them after transfer.

In Information Integrator, files are compressed and decompressed automatically in conjunction with file transfer.

The objective of compression is to reduce data transfer times. Specify this option when using a network with a transfer rate that is slower than the processing speed for data compression and decompression. Note also that if the data to be transferred is already compressed, it should be transferred without further data compression. This feature can be used when the communication protocol is FTP+.

Point

When the communication protocol is other than FTP+ and the data is in zip format

Data compressed in zip format can be handled by decompressing after collection using a plug-in provided by Information Integrator.

Refer to "3.10.3 Compressing or Decompressing Data" for information on plug-ins.

Concurrency control

Transfer concurrency refers to the number of networks in concurrent use by a system and/or individual communications channel for 'File transfer', 'Execute remote jobs', and/or to 'Notify job results'. In Information Integrator, the upper limit value can be set for this transfer concurrency.

When multiple file transfers are being executed concurrently with many remote systems, protocol response message notifications can become delayed in accordance with the increased load placed on the Information Integrator Server. This causes timeout errors or network errors to be detected on the remote system, which may lead to multiple file transfer errors. Response delays for received file transfers can be prevented by using the concurrency control to limit the number of file transfers processed concurrently.

Note also that HULFT (Host UNIX Linkage File Transfer) product settings should be adhered to for transfer concurrency using HULFT.

1.2.3.2 High-speed file transfer

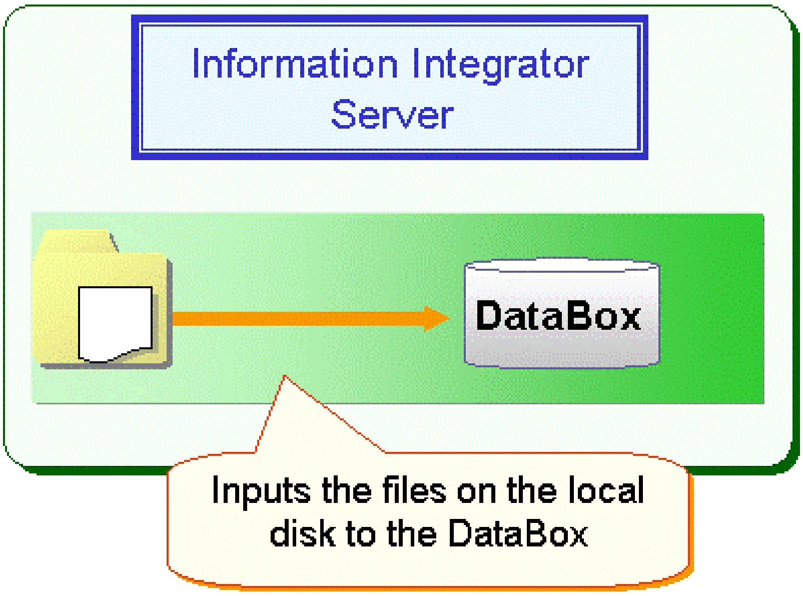
The transfer time of file transfer is proportional to the forwarding distance. In forwarding between remote places and forwarding that uses the network that line speed is slow, an increase in transfer time becomes remarkable.

Information Integrator speed up the data transfer between remote places greatly by building in the technology that uses the WAN band to its maximum.

1.2.4 File Input

Enter the files on the local disk into the DataBox to enable subsequent processes such as data conversion and data delivery to be performed.

Figure 1.7 File input


Information Integrator Server
DataBox
Inputs the files on the local disk to the DataBox

1.2.5 Extract from DB

'Extract from DB' is a feature for extracting the data stored in the database of the collection source from the Information Integrator Server-side.

Data can also be extracted from databases within servers with an Information Integrator Server installed.

Use 'Extract from DB' to extract data stored in the database of the collection source in the following ways:

- Extract all items

  This extracts full database tables.

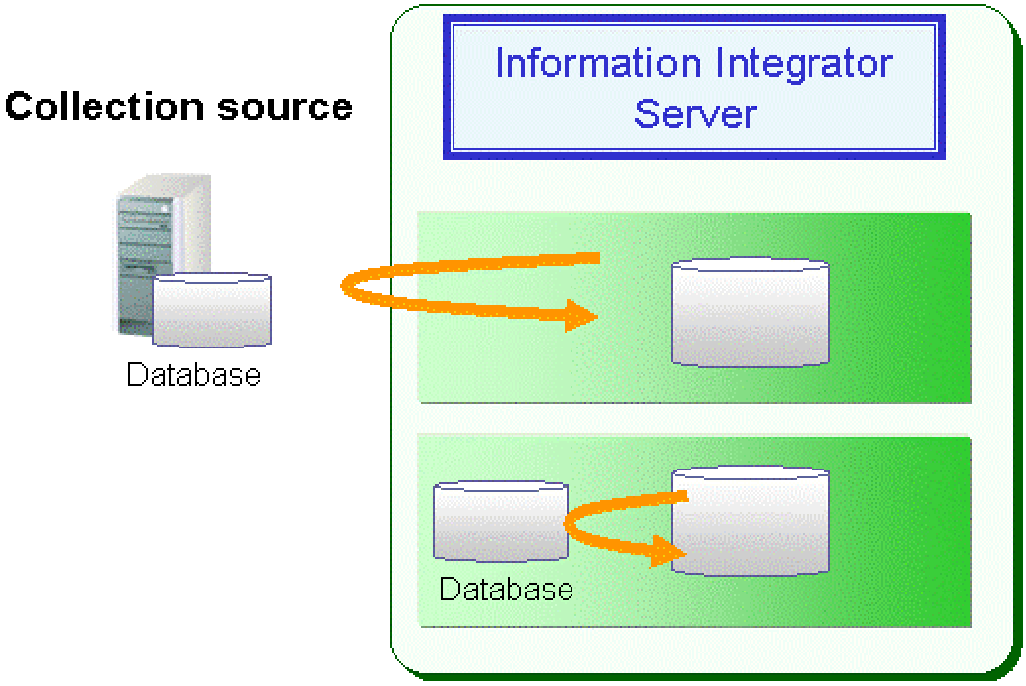
- Conditional extraction

  This extracts selected records.

- Conditions file

  This uses a file in which extraction conditions have been specified to extract selected records from the Information Integrator Server-side.

Figure 1.8 Extract from DB



Sequence control

Use this feature to preserve the sequence of collected data right through to the storage destination (delivery destination), such as when delivering data extracted from a database in the transaction order.

- Process Flow units

  This feature controls the sequence of Process Flow units.

  If the same Process Flow is already being executed, the new process will not begin until the preceding Process Flow is complete.

- Monitoring the wait state

- Processes that are in a wait state due to the sequence control will remain waiting until they can be processed. However, the operation management feature can be used to check whether or not a process is waiting.

- Although checks using the operation management feature require constant attention on the part of the user, the warning message indicated below will be output to the event log and system log if the process enters a waiting state, so monitoring can be performed using message monitoring products.

```
A process is waiting for its sequence - process ID (~~) service number (~~~)
```

1.2.6 Extract Object

Use this feature when collecting objects.

In the extraction of the Salesforce object, the data stored in collection source system (Salesforce) is extracted from the Information Integrator Server side.

The user can extract data using the following methods:

- Extract all items

  This extracts everything from the Information Integrator Server-side.

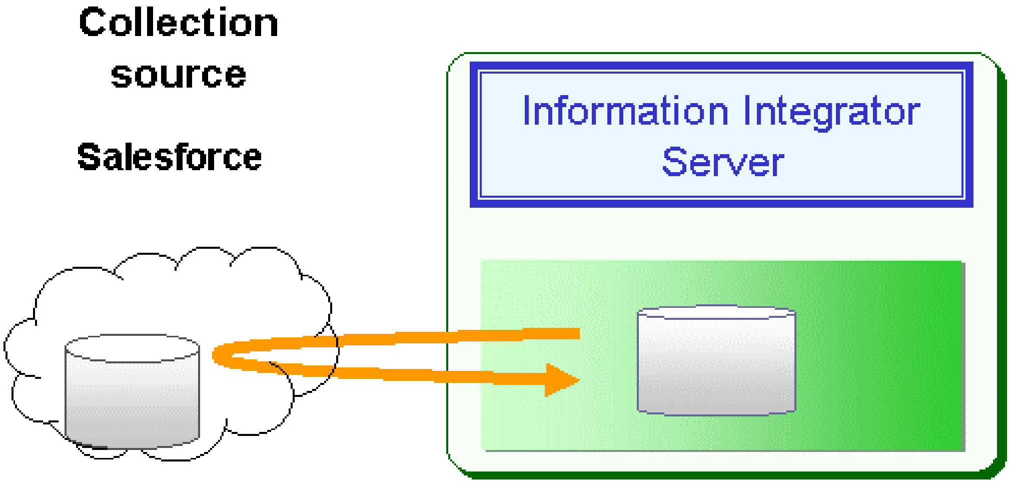
- Conditional extraction

  This extracts selected records from the Information Integrator Server-side.

- Conditions file

  This uses a file in which extraction conditions have been specified to extract selected records from the Information Integrator Server-side.

Figure 1.9 Extract data (as initiator)

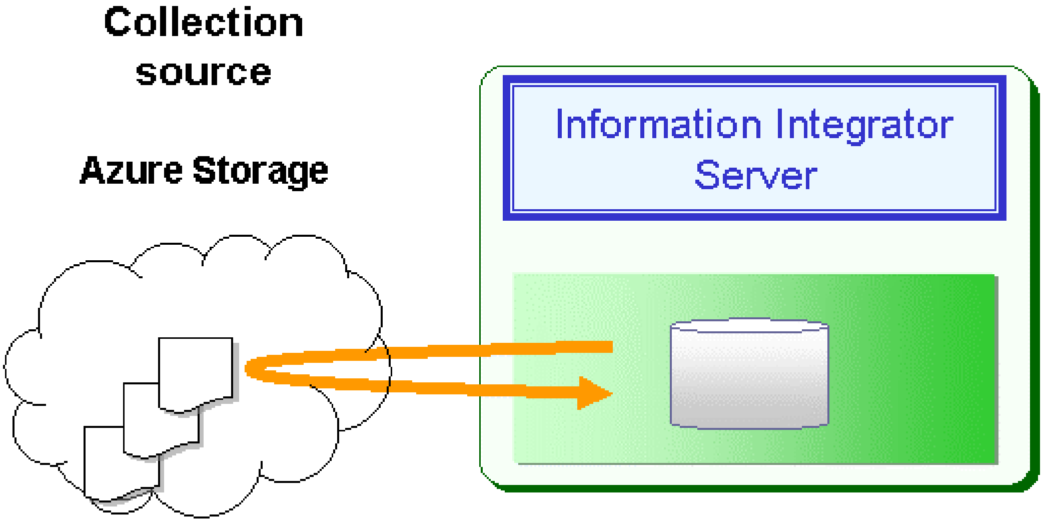


1.2.7 BLOB Collection

Use this feature when collecting Azure Storage data.

BLOB collection collects data stored in Azure Storage at the collection source from the Information Integrator Server-side in BLOB units.

Figure 1.10 BLOB collection



1.3 Data Delivery

This section describes the data delivery process.

1.3.1 Data Formats that can be Delivered by Information Integrator

The data sources that can be delivered by Information Integrator are indicated below.

- File formats
  - CSV files

    - Binary files
    - XML files
- Database formats
    - Symfoware Server
    - Oracle
    - SQL Server
    - DB2
    - PostgreSQL
    - SQL Azure
    - Other databases (ODBC connection)
- Object formats
    - Salesforce objects

      This refers to objects stored in Salesforce.
- BLOB format

  This refers to data delivered to Azure Storage.

Point

Compressed data collection

Prior to sending BLOB format data, files can be compressed in zip format using a plug-in provided by Information Integrator and then delivered accordingly.

Refer to "3.10.3 Compressing or Decompressing Data" for information on plug-ins.

1.3.2 Data Delivery Methods

When delivering files, use 'Send files' to deliver files to a remote system, or 'File output' to place files on the local disk of the local system.

Use 'Store in DB' to store data in the database.

Use 'Store object' to store data in a public cloud.

Use 'BLOB delivery' to store data in Azure Storage.

A description of each of the data storage methods is provided below.

1.3.3 Send Files

The 'Send files' feature is described below.

Two methods can be used to send files: 'Send files (as initiator)' and 'Send files (as responder)'.

Send files (as initiator)

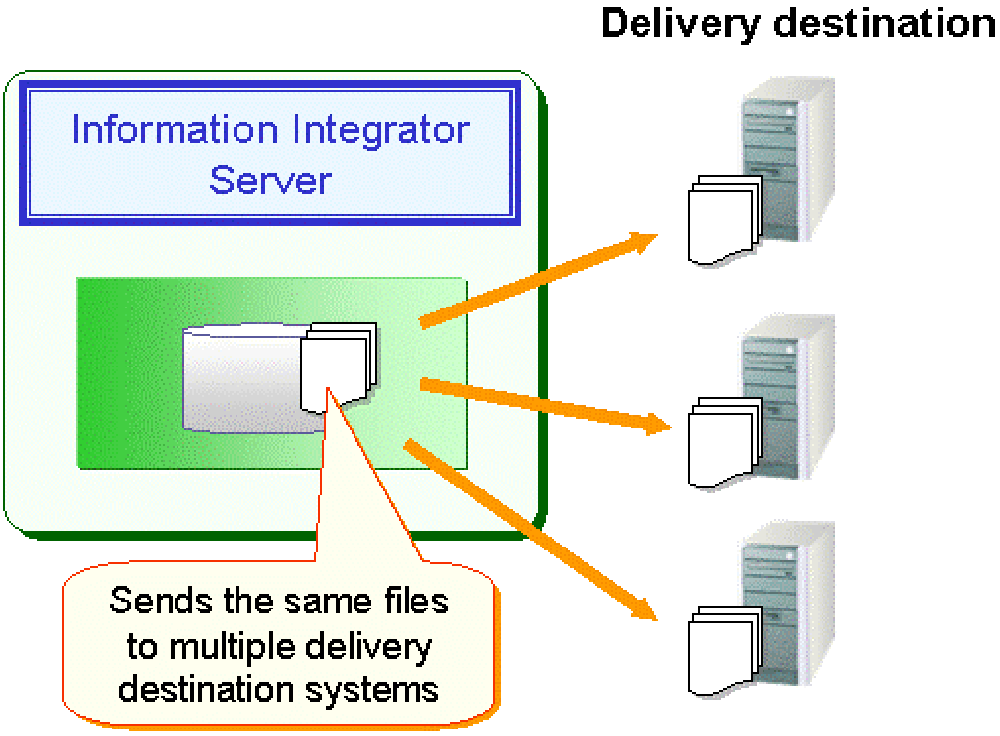
Files for which all processes have been completed are sent to the delivery destination.

'Send files (as initiator)' comprises the following features:

- Broadcast delivery

  This feature sends the same data to multiple delivery destinations, and is used for master data sending tasks.

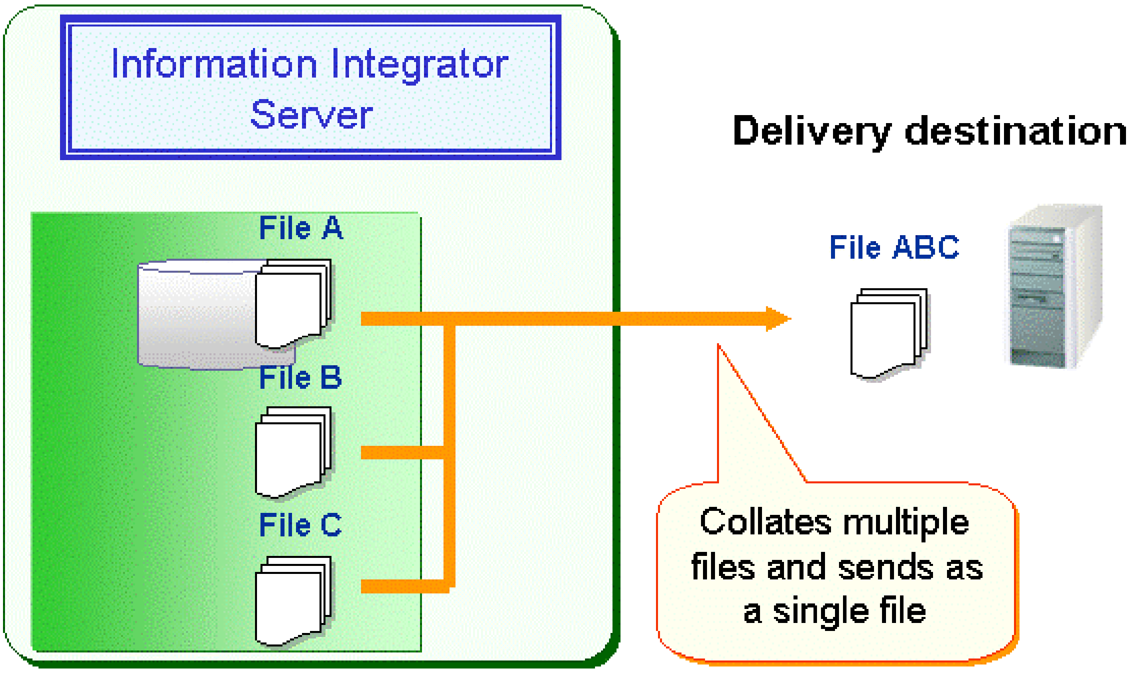
  Figure 1.11 Broadcast delivery


Delivery destination
Information Integrator Server
Sends the same files to multiple delivery destination systems

- Batch delivery

  This feature consolidates multiple files for which all processes have been completed into a single file and then sends it to the delivery destination. Use this feature to consolidate and send single slip data during slip processing.

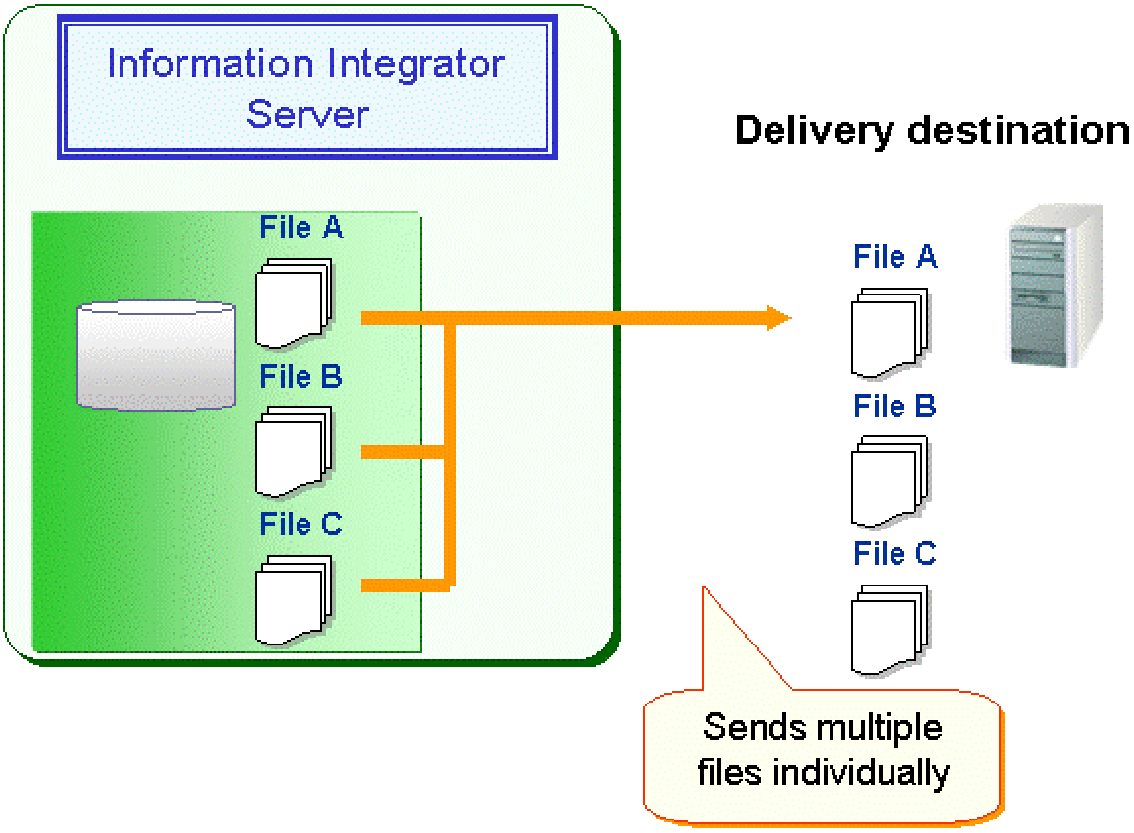
  Figure 1.12 Batch delivery


Information Integrator Server
File A
File B
File C
Delivery destination
File ABC
Collates multiple files and sends as a single file

- Individual delivery

  This feature sends multiple files for which all processes have been completed to the delivery destination as individual files.

Figure 1.13 Individual delivery

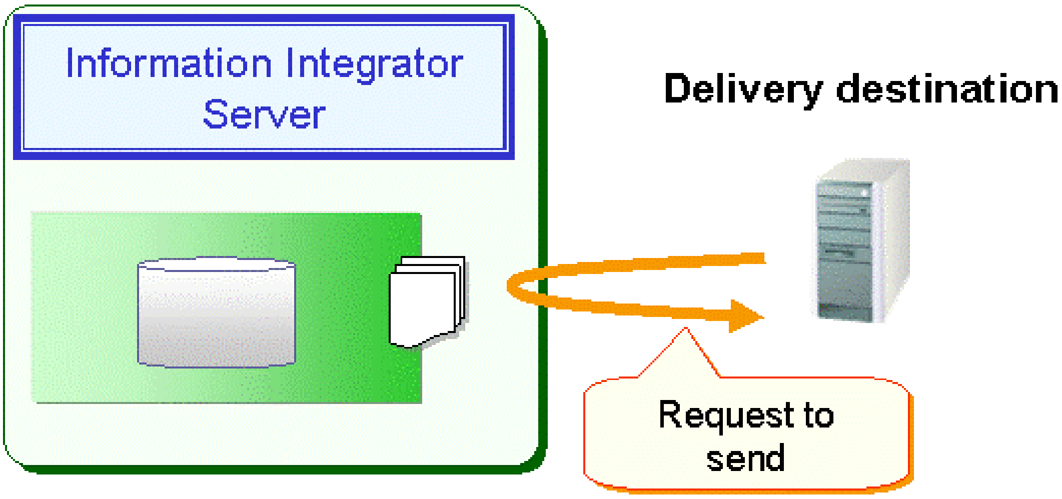

Information Integrator Server
Delivery destination
File A
File B
File C
File A
File B
File C
Sends multiple files individually

Send files (as responder)

Files on Information Integrator are sent when prompted by a send request from the delivery destination.

All 'Send files (as responder)' processes must have been completed at the point the request from the delivery destination is received.

Figure 1.14 Send files (as responder)



1.3.3.1 Optional Features for 'Send Files'

Automatic compression and decompression

The time required for file transfer is proportionate to the size of the data being transferred. Transfers involving large-scale data or networks with slow connection speeds will require a noticeably longer transfer time.

In such cases, both the volume of data being sent over the network and the time required to transfer it can be reduced by compressing files and then decompressing them after transfer.

In Information Integrator, files are compressed and decompressed automatically in conjunction with file transfer.

The objective of compression is to reduce data transfer times. Specify this option when using a network with a transfer rate that is slower than the processing speed for data compression and decompression. Note also that if the data to be transferred is already compressed, it should be transferred without further data compression. This feature can be used when the communication protocol is FTP+.

Point

When the communication protocol is other than FTP+ and the data is in zip format

Prior to sending, files can be compressed in zip format using a plug-in provided by Information Integrator and then delivered accordingly.

Refer to "3.10.3 Compressing or Decompressing Data" for detailed information on plug-ins.

Concurrency control

Transfer concurrency refers to the number of networks in concurrent use by a system and/or individual communications channel for 'File transfer', 'Execute remote jobs', and/or to 'Notify job results'. In Information Integrator, the upper limit value can be set for this transfer concurrency.

When multiple file transfers are being executed concurrently with many remote systems, protocol response message notifications can become delayed in accordance with the increased load placed on the Information Integrator Server. This causes timeout errors or network errors to be detected on the remote system, which may lead to multiple file transfer errors. Response delays for received file transfers can be prevented by using the concurrency control to limit the number of file transfers processed concurrently.

Note also that HULFT product settings should be adhered to for transfer concurrency using HULFT.

Transfer rate control

When the network speed is slow for high-volume file transfers, the load placed on the communications channel can be alleviated by controlling the data transfer rate. If the data transfer rate is not controlled in such cases, the network will become monopolized by the large volume of files and this could have an adverse effect on other business tasks.

Note

This feature is for setting the maximum transfer rate when sending files. It does not guarantee that the transfer rate will be equal to the specified value. Transfer rates may be slower than the specified value depending on other factors such as the system load and network load.

Overwriting or appending files

When using 'Send files', the user can specify whether to overwrite the destination files or to append them.

1.3.3.2 High-speed file transfer

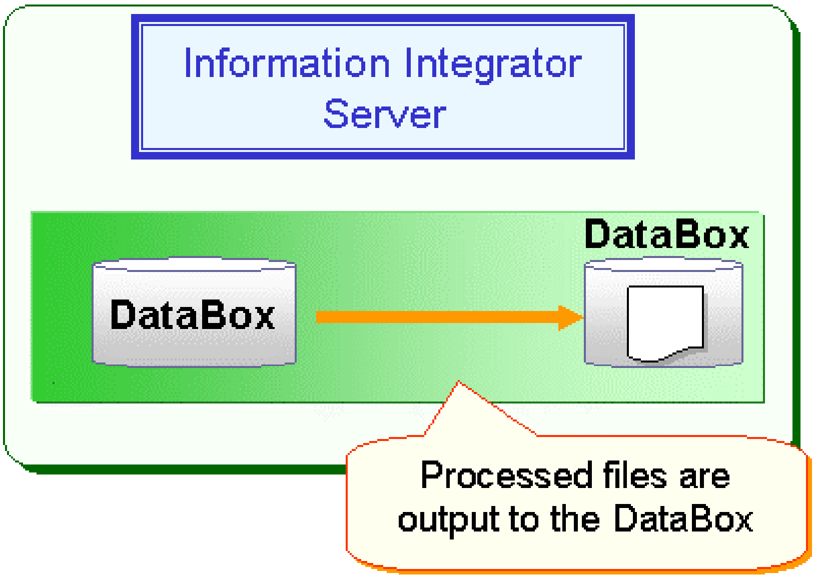
The transfer time of file transfer is proportional to the forwarding distance. In forwarding between remote places and forwarding that uses the network that line speed is slow, an increase in transfer time becomes remarkable.

Information Integrator speed up the data transfer between remote places greatly by building in the technology that uses the WAN band to its maximum.

1.3.4 File Output

This feature outputs files to the DataBox for which processes (such as conversion) are complete.

Figure 1.15 File output


Information Integrator Server
DataBox
DataBox
Processed files are output to the DataBox

1.3.5 Store in DB

This feature stores data directly in the database of the delivery destination. It is also possible to store data in multiple tables with a single action.

The four methods for 'Store in DB' are listed below.

- Replace data

  Replaces all data on the DB table with the resulting data from collection or conversion processes.

- Add data

  Adds the resulting data from collection and conversion processes to the data on the DB table.

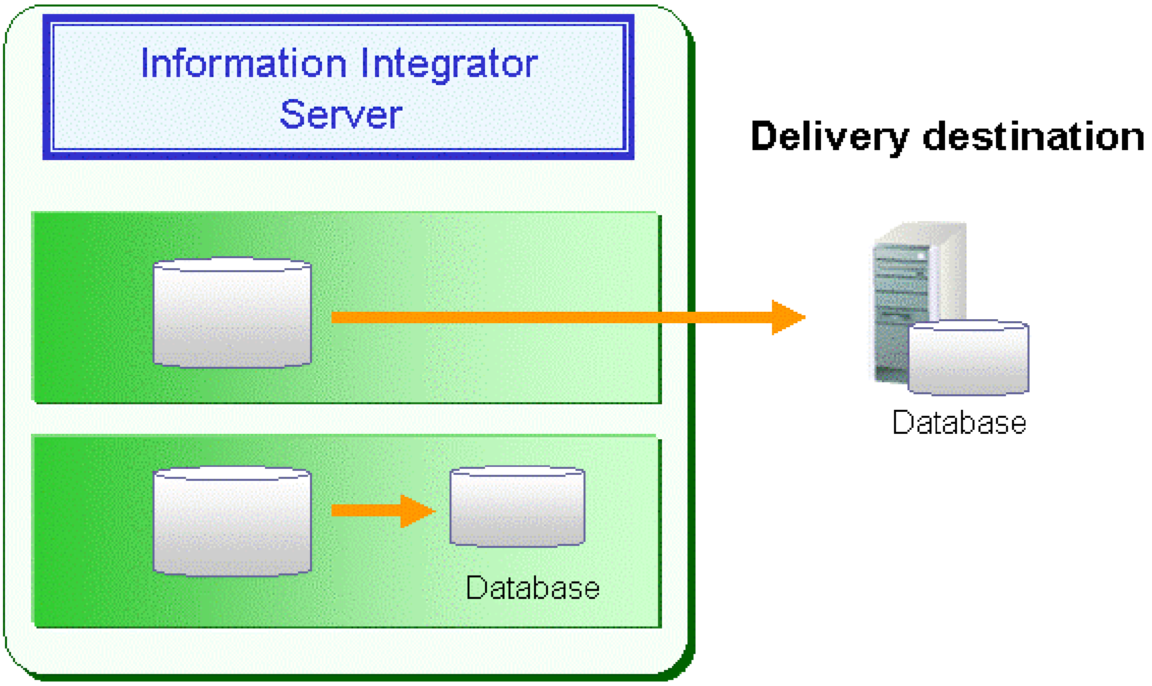
- Update data

  Updates all items in relevant records on a DB table with the resulting data from collection or conversion processes.

- Add/update data

  Updates all items in relevant records on a DB table with the resulting data from collection and conversion processes. If a relevant record is not present, it will be added.

Figure 1.16 Store in DB



1.3.6 Store Object

Use this feature when storing objects.

When storing Salesforce objects, data is stored in the (Salesforce) delivery destination from the Information Integrator Server-side.

The data storage methods are outlined below.

- Add data

  Adds resulting objects, for which collection and conversion processes have been executed, to Salesforce objects.

- Update data

  All relevant items in Salesforce objects are updated with resulting data from collection or conversion processes.

- Add/Update data

  All relevant items in Salesforce objects are updated with resulting data from collection and conversion processes. If a relevant item is not present, the object will be added.

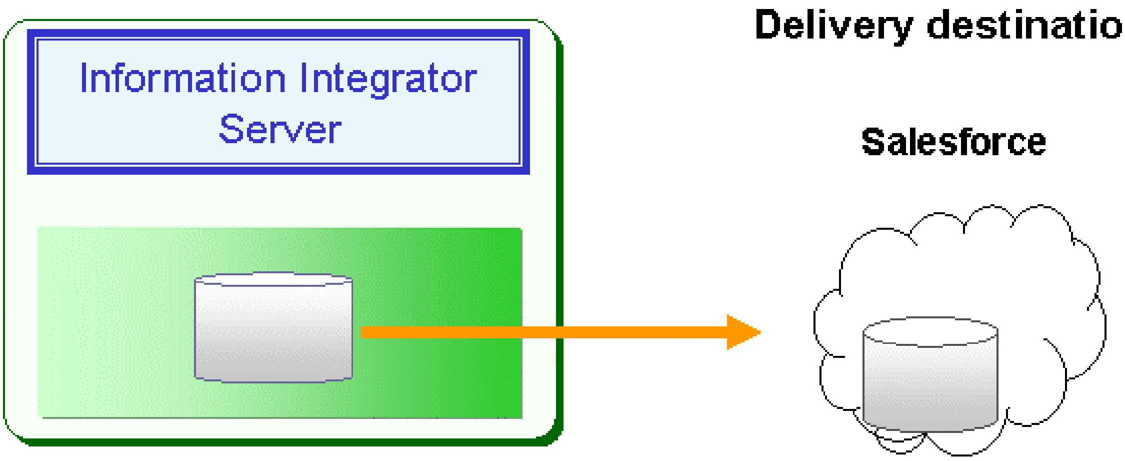
Point

Adding data

We recommend storing objects that have utilized the 'Add/Update data' feature when adding data.

If any kind of error occurs when storing using 'Add data', it will be necessary to restore the Salesforce data (perform a backup beforehand).

Figure 1.17 'Store object' for Salesforce



1.3.6.1 Optional Features for 'Store Object'

Large volumes of data can be split prior to transferring. Refer to "1.3.8 Cloud Integrated Data Fragment Transfer" for detailed information.

1.3.7 BLOB Delivery

Use this feature when delivering BLOB data.

The BLOB data delivery feature delivers data from the Information Integrator Server-side to the (Azure Storage) delivery destination.

Refer to "" for information on connection topology with Azure Storage.

If a relevant BLOB is not present in Azure Storage, a new BLOB will be created and the BLOB data delivered. If a BLOB is already present, the BLOB data will be overwritten and delivered.

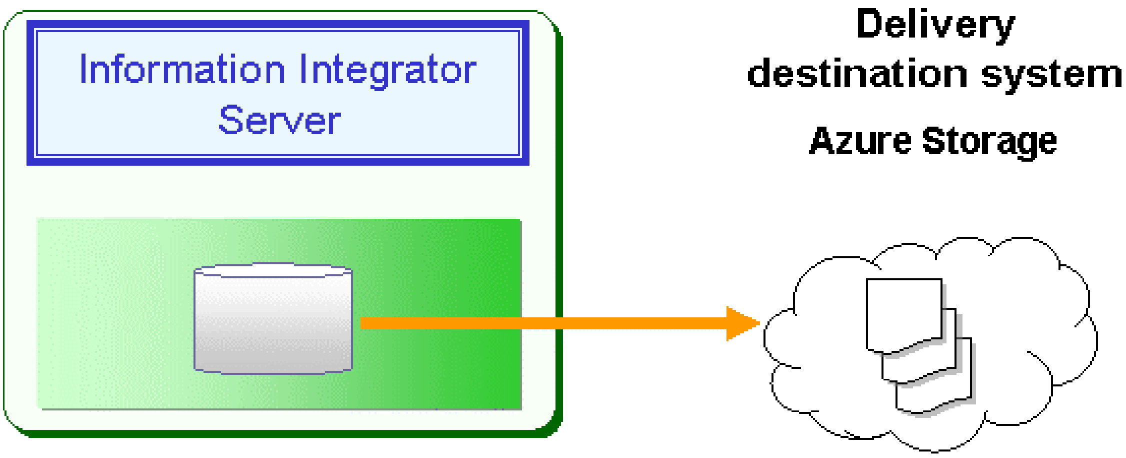
Point

Prior to executing BLOB delivery

Prior to executing BLOB delivery, ensure that data on Azure Storage has been backed up using an appropriate backup method, to insure against any kind of error occurring at the time of data delivery.

Similarly, ensure that rollback is configured in the operation using Azure Storage features, also to insure against any kind of error occurring at the time of data delivery.

Figure 1.18 BLOB delivery



1.3.7.1 Optional Features for BLOB Delivery

Large volumes of BLOB data can be split prior to transferring. Refer to "1.3.8 Cloud Integrated Data Fragment Transfer" for detailed information.

1.3.8 Cloud Integrated Data Fragment Transfer

Large volumes of data can be split prior to transferring in order to alleviate the load placed on the communications channel.

If the data transfer volume is not controlled in such cases, the network will become monopolized by the large volume of data being transferred and this could have an adverse effect on other business tasks.

1.4 Data Conversion and Editing

Information Integrator can be used to convert and edit collected data.

Note that it is also possible to convert and edit only specific records and/or items selected from the collected data.

The data conversion and editing features in Information Integrator are listed below.

- File format conversion
- Character code conversion
- Edit items
- Edit records

Point

- The Information Integrator data conversion and editing features perform the following data checks on both pre- and post-conversion data at the time of conversion and editing, for each data type (CSV files or binary files).
    - Data formats

      The following input value checks will be performed:
        - Full-width character string attributes
        - Date/time attributes
        - Numeric attributes
        - Binary attributes
        - Bit attributes
    - Data checks to determine the correct number of digits.
    - Mandatory input checks

  Note

    - These data checks are not performed for XML files if the pre- or post-conversion data is an XML file.
    - It is not possible to perform data checks on collected data (pre-conversion data) in the case of data extracted from Salesforce in the following data types:
        - int
        - double
        - currency
        - percent

- Conversion applications that were in use on existing systems can be embedded in Information Integrator using a plug-in feature. Refer to "1.7 Plug-ins" for detailed information on plug-in features.
- When handling post-conversion data in CSV or XML files, assign enclosing characters (quotation marks) only to character-type data. Alternatively, the user can choose not to assign enclosing characters to any data.

1.4.1 File Format Conversion

This feature converts files from one format to another. Conversions can be performed for all collected data. Format conversion can be performed reciprocally on binary files, CSV files and XML files to convert files between these three types.

Note

XML file conversion

To convert XML files to XML files in Information Integrator, the XML files must first be converted once to CSV file format and then re-converted to XML file format.

Error processing during data conversion

It is possible to select the action taken by Information Integrator after an error occurs during data conversion when performing file format conversion or standard code conversion.

The range of actions that may be selected is listed below.

Table 1.1 Cause of error during data conversion and follow-up action to be taken

Attribute	Cause of error	Action	Error message
String operation	Character code conversion	Abort	Aborted (error)
		Continue	Enter substitute characters and continue processing (warning)
	Character overflow	Abort	Aborted (error)
		Continue	Delete overflow characters and continue processing (warning)
Internal decimal	First 4 bits do not contain 0	Abort	Aborted (error)
		Continue	Make the first 4 bits '0' and continue processing (warning)
Numeric operation	Digit overflow	Abort	Aborted (error)
	Division by zero	Abort	Aborted (error)

1.4.2 Character Code Conversion

This feature converts the specified character code to a different character code.

Conversions can be performed for all collected data.

In Information Integrator, the 'First standard' and 'Second standard' of JIS (Japanese Industrial Standard) are available as standard features for character code conversion. Possible conversion combinations are indicated below.

Table 1.2 Possible character code conversion combinations

		Output character code														
Input character code		Shift-JIS	EUC			JEF-EBCDIC				Unicode						
			U90	S90	HP	Kana (shape based)	Kana (area based)	ASCII (shape based)	ASCII (area based)	UCS2	UCS2(MS)	CFSD (*2)	UTF8	UTF8-4	UTF16	UTF16(MS)
Shift-JIS		Y(*1)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
EUC	U90	Y	-	-	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	S90	Y	-	Y (*1)	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	HP	Y	-	-	-	Y	Y	Y	Y	-	-	-	-	-	-	-
JEF-EBCDIC	Kana (shape based)	Y	Y	Y	Y	-	-	-	-	Y	Y	Y	Y	Y	Y	Y
	Kana (area based)	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-
	ASCII (shape based)	Y	Y	Y	Y	-	-	-	-	Y	Y	Y	Y	Y	Y	Y
	ASCII (area based)	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-
Unicode	UCS2	Y	Y	Y	-	Y	-	Y	-	Y (*1)	Y	Y	Y	-	-	-
	UCS2(MS)	Y	Y	Y	-	Y	-	Y	-	Y	Y (*1)	Y	Y	-	-	-
	CFSD (*2)	Y	Y	Y	-	Y	-	Y	-	Y	Y	Y (*1)	Y	-	-	-
	UTF8	Y	Y	Y	-	Y	-	Y	-	Y	Y	Y	Y (*1)	-	-	-
	UTF8-4	Y	Y	Y	-	Y	-	Y	-	Y	Y	Y	Y	Y (*1)	Y	Y
	UTF16	Y	Y	Y	-	Y	-	Y	-	Y	Y	Y	Y	Y	Y (*1)	Y
	UTF16 (MS)	Y	Y	Y	-	Y	-	Y	-	Y	Y	Y	Y	Y	Y	Y (*1)
IBM Japanese Character set	Kana	Y	Y	Y	-	Y	Y	-	-	-	-	-	-	-	-	-
	Lower-case alphabetic characters	Y	Y	Y	-	-	-	-	-	-	-	-	-	-	-	-

Y: conversion possible. -: Non-convertible combination.

*1: May be specified when changing the numeric representation format.

*2: Cannot be specified when using the ificnvdat command.

Point

The character code for old JEF character forms or vendor native characters can be changed by toggling the character code conversion table. Furthermore, it is also possible to convert the character codes of user-defined characters by customizing the character code conversion table.

Note

If Charset Manager has been implemented in the system operated by Information Integrator, and the Charset Manager character code conversion table has been customized, please incorporate this character code conversion table into the Information Integrator character code conversion table. Refer to *Command Reference* for more information.

Error processing during data conversion

It is possible to select the action taken by Information Integrator after an error occurs during data conversion when performing file format conversion or character code conversion.

The range of actions that may be selected is listed below.

Table 1.3 Cause of error during data conversion and follow-up action to be taken

Attribute	Cause of error	Action	Error message
String operation	Character code conversion	Abort	Aborted (error)
		Continue	Enter substitute characters and continue processing (warning)
	Character overflow	Abort	Aborted (error)
		Continue	Delete overflow characters and continue processing (warning)
Internal decimal	First 4 bits do not contain 0	Abort	Aborted (error)
		Continue	Make the first 4 bits '0' and continue processing (warning)
Numeric operation	Digit overflow	Abort	Aborted (error)
	Division by zero	Abort	Aborted (error)

1.4.3 Edit Items

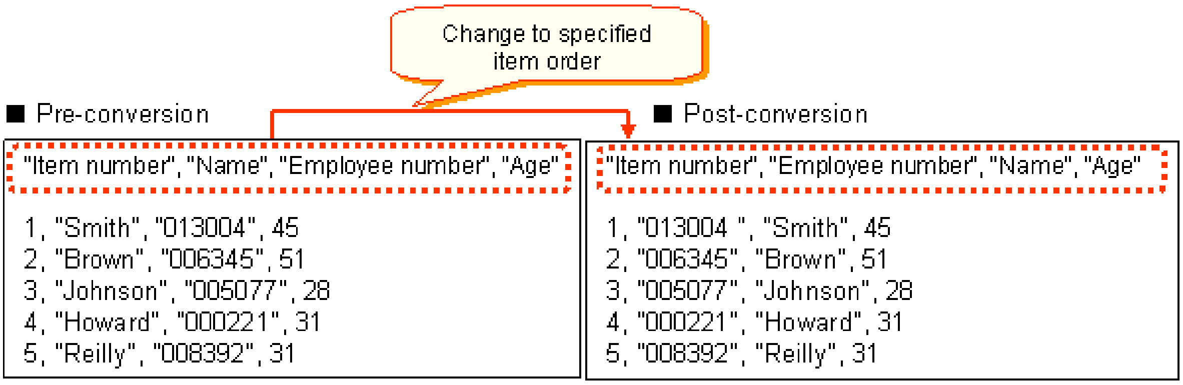
Editing can be performed for data items.

1.4.3.1 Sort Items

Items can be sorted in a specified order.

[Example]

'Item number', 'Employee number', 'Name' and 'Age' is the sort order of these items in the source data.



Change to specified item order

■ Pre-conversion

```
"Item number", "Name", "Employee number", "Age"

1, "Smith", "013004", 45
2, "Brown", "006345", 51
3, "Johnson", "005077", 28
4, "Howard", "000221", 31
5, "Reilly", "008392", 31
```

■ Post-conversion

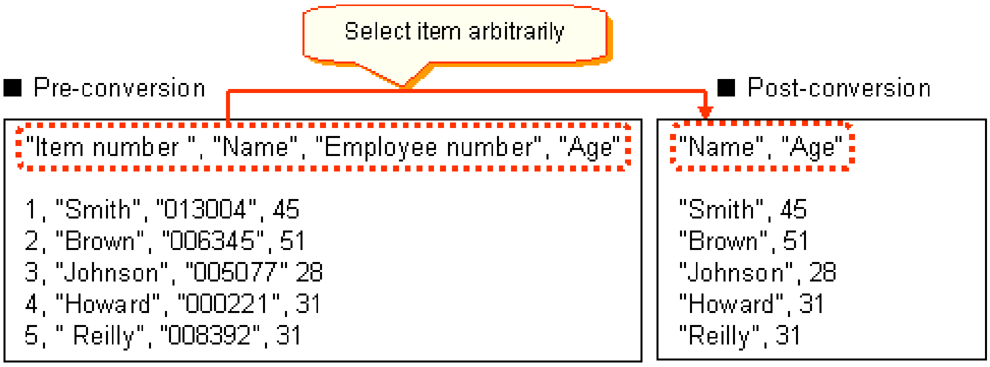
```
"Item number", "Employee number", "Name", "Age"

1, "013004 ", "Smith", 45
2, "006345", "Brown", 51
3, "005077", "Johnson", 28
4, "000221", "Howard", 31
5, "008392", "Reilly", 31
```

1.4.3.2 Item Selection

It is possible to output only specified items from items in the source data.

[Example]

It is possible to output only the 'Name' and 'Age' items from the 'Item number', 'Name', 'Employee number', and 'Age' items in the source data. The specified items will be output to the output data.



Select item arbitrarily

■ Pre-conversion

```
"Item number ", "Name", "Employee number", "Age"

1, "Smith", "013004", 45
2, "Brown", "006345", 51
3, "Johnson", "005077" 28
4, "Howard", "000221", 31
5, " Reilly", "008392", 31
```

■ Post-conversion

```
"Name", "Age"

"Smith", 45
"Brown", 51
"Johnson", 28
"Howard", 31
"Reilly", 31
```

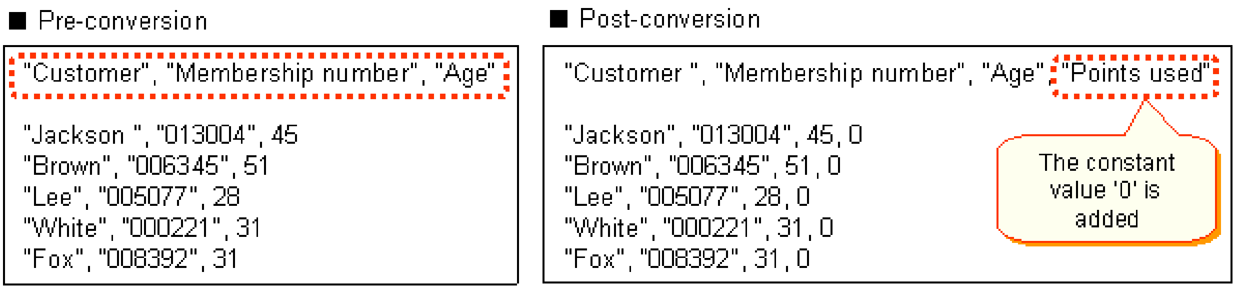
1.4.3.3 Adding Items

Item values not present in the source data (such as constants or operation results) can be stored in new items.

[Example]

This example illustrates the constant value '0' being stored in an item (points used) not present in the source data.

Additionally, operation results such as substring extractions can also be stored in new items.



■ Pre-conversion

```
"Customer", "Membership number", "Age"

"Jackson ", "013004", 45
"Brown", "006345", 51
"Lee", "005077", 28
"White", "000221", 31
"Fox", "008392", 31
```

■ Post-conversion

```
"Customer ", "Membership number", "Age", "Points used"

"Jackson", "013004", 45, 0
"Brown", "006345", 51, 0
"Lee", "005077", 28, 0
"White", "000221", 31, 0
"Fox", "008392", 31, 0
```

The constant value '0' is added

1.4.3.4 Attribute Conversion

This feature performs the data type conversion for both the input and output sides and converts the data in accordance with the data type of the output side. Note that this feature is valid for binary record types.

Table 1.4 Attribute conversion matrix

	outdata															
indata	SQL_CHAR	SQL_VARCHAR	SQL_NCHAR	SQL_VARNCHAR	SQL_INTEGER	SQL_SMALLINT	SQL_NUMERIC	SQL_DECIMAL	SQL_REAL	SQL_FLOAT	SQL_DATE	SQL_TIME	SQL_TIMESTAMP	SQL_LONGVARBINARY	SQL_BINARY	SQL_BIT
SQL_CHAR	Y 1	Y 2	CP 3	CP 4	Y 5	Y 6	Y 7	Y 8	Y 9	Y 10	Y 11	Y 12	Y 13	-	-	-
SQL_VARCHAR	Y 14	Y 15	CP 16	CP 17	Y 18	Y 19	Y 20	Y 21	Y 22	Y 23	Y 24	Y 25	Y 26	-	-	-
SQL_NCHAR	CP 27	CP 28	Y 29	Y 30	CP 31	CP 32	CP 33	CP 34	-	-	-	-	-	-	-	-
SQL_VARNCHAR	CP 35	CP 36	Y 37	Y 38	CP 39	CP 40	CP 41	CP 42	-	-	-	-	-	-	-	-
SQL_INTEGER	Y 43	Y 44	CP 45	CP 46	Y 47	Y 48	Y 49	Y 50	-	-	-	-	-	-	-	-
SQL_SMALLINT	Y 51	Y 52	CP 53	CP 54	Y 55	Y 56	Y 57	Y 58	-	-	-	-	-	-	-	-
SQL_NUMERIC	Y 59	Y 60	CP 61	CP 62	Y 63	Y 64	Y 65	Y 66	-	-	-	-	-	-	-	-
SQL_DECIMAL	Y 67	Y 68	CP 69	CP 70	Y 71	Y 72	Y 73	Y 74	-	-	-	-	-	-	-	-
SQL_REAL	Y 75	Y 76	-	-	-	-	-	-	Y 77	Y 78	-	-	-	-	-	-
SQL_FLOAT	Y 79	Y 80	-	-	-	-	-	-	Y 81	Y 82	-	-	-	-	-	-
SQL_DATE	Y 83	Y 84	-	-	-	-	-	-	-	-	Y 85	-	-	-	-	-
SQL_TIME	Y 86	Y 87	-	-	-	-	-	-	-	-	-	Y 88	-	-	-	-
SQL_TIMESTAMP	Y 89	Y 90	-	-	-	-	-	-	-	-	-	-	Y 91	-	-	-
SQL_LONGVARBINARY	-	-	-	-	-	-	-	-	-	-	-	-	-	Y 92	-	-
SQL_BINARY	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y 93	-
SQL_BIT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y 94

Y: Conversion possible.

CP: Conversion is conditionally possible if the following is performed: 'Full-width/half-width conversion' is specified in the data conversion definition 'Operation expression-operation type', and either 'Full-width character conversion' or 'Half-width character conversion' is specified in the data conversion definition 'Operation expression-argument'.

-: Conversion is not possible.

Numeric information matches the 'Item number' in the "Attribute conversion data example".

Point

Important notes for attribute conversion

- Conversion between binary files and CSV files for SQL_LONGVARBINARY types and/or SQL_BINARY types is as indicated below.
  - When converting binary files to CSV files:

    Binary data is converted to CSV file data using BASE64 conversion.
  - When converting CSV files to binary files:

    Data in CSV files that was from binary files to CSV files using BASE64 conversion is converted to binary data. Data that does not satisfy the specified total number of digits is padded with zeroes.
- Conversion between binary files and CSV files for SQL_BIT types is as indicated below:
  - When converting binary files to CSV files:
    - Binary data is converted to strings (0 or 1 as half-width digits) expressed in binary numbers.
    - If the specified total number of digits is not a multiple of 8, ignore any data at the end of the CSV file input data.

    For example, if the total number of digits has been defined as 6 digits and the binary data is '0x0A (bit sequence) 00001010', the data following conversion to CSV file data will be '000010'.
  - When converting CSV files to binary files:
    - CSV file strings (0 or 1 as half-width digits) expressed in binary numbers are converted to binary data.
    - If the specified total number of digits is not equal to a number that is a multiple of 8, pad the end of the binary file output data with zeroes.

    For example, if the total number of digits has been defined as 6 digits and the CSV file data is '000010', the data following conversion to binary file data will be '0x08 (bit sequence) 00001000'.

The following table provides examples of data (output data) following conversion of input data types listed in the previous attribute conversion matrix, which denoted whether conversion was possible with 'Y' or 'CP' (Conditionally Possible). The 'Item number' matches the numeric information in the "attribute conversion matrix".

Table 1.5 Attribute conversion data examples

Item No.	Category	Input data	Input data example	Output data	Output data example	Remarks
1	Character attribut	0x31323334	SQL_CHAR(4)	0x31323334	SQL_CHAR(4)	(Half-width) characters 1234
2				0x040031323334	SQL_VARCHAR(4)	(Half-width) characters 1234
3				0x8250825182528253	SQL_NCHAR	(Full-width) characters 1234
4				0x04008250825182528253	SQL_VARNCHAR	(Full-width) characters 1234
5				0xD2040000	SQL_INTEGER	Numbers 1234
6				0xD204	SQL_SMALLINT	Numbers 1234

Item No.	Category	Input data	Input data example	Output data	Output data example	Remarks
7	es			0x31323334	SQL_NUMERIC	Unsigned
				0x31323344	SQL_NUMERIC	Signed, end, bit format
				0x41323334	SQL_NUMERIC	Signed, beginning, bit format
				0x2B31323334	SQL_NUMERIC	Signed, beginning, character format
				0x313233342B	SQL_NUMERIC	Signed, end, character format
8				0x012343	SQL_DECIMAL	Numbers 1234
9				0x449a4000	SQL_REAL	Numbers 1234
10				0x4093480000000000	SQL_FLOAT	Numbers 1234
11		0x323030392F30372F3031	SQL_CHAR(10)	0xD9070701	SQL_DATE	Date 2009/07/01
12		0x31323A33343A3536	SQL_CHAR(8)	0x0C2238	SQL_TIME	Time 12:34:56
13		0x323030392F30372F30312031323A33343A3536	SQL_CHAR(19)	0xD90707010C2238	SQL_TIMESTAMP	Date/time 2009/07/01 12:34:56
14		0x040031323334	SQL_VARCHAR(4)	0x31323334	SQL_CHAR(4)	(Half-width) characters 1234
15				0x040031323334	SQL_VARCHAR(4)	(Half-width) characters 1234
16				0x8250825182528253	SQL_NCHAR	(Full-width) characters 1234
17				0x04008250825182528253	SQL_VARNCHAR	(Full-width) characters 1234
18				0xD2040000	SQL_INTEGER	Numbers 1234
19				0xD204	SQL_SMALLINT	Numbers 1234
20				0x31323334	SQL_NUMERIC	Unsigned
				0x31323344	SQL_NUMERIC	Signed, end, bit format
				0x41323334	SQL_NUMERIC	Signed, beginning, bit format
				0x2B31323334	SQL_NUMERIC	Signed, beginning, character format
				0x313233342B	SQL_NUMERIC	Signed, end, character format

Item No.	Category	Input data	Input data example	Output data	Output data example	Remarks
21				0x012343	SQL_DECIMAL	Numbers 1234
22				0x449a4000	SQL_REAL	Numbers 1234
23				0x4093480000000000	SQL_FLOAT	Numbers 1234
24		0x0a00323030392F30372F3031	SQL_VARCHAR(10)	0xD9070701	SQL_DATE	Date 2009/07/01
25		0x080031323A33343A3536	SQL_VARCHAR(8)	0x0C2238	SQL_TIME	Time 12:34:56
26		0x1300323030392F30372F30312031323A33343A3536	SQL_VARCHAR(19)	0xD90707010C2238	SQL_TIMESTAMP	Date/time 2009/07/01 12:34:56
27		0x8250825182528253	SQL_NCHAR(4)	0x31323334	SQL_CHAR(4)	(Half-width) characters 1234
28				0x040031323334	SQL_VARCHAR(4)	(Half-width) characters 1234
29				0x8250825182528253	SQL_NCHAR	(Full-width) characters 1234
30				0x04008250825182528253	SQL_VARNCHAR	(Full-width) characters 1234
31				0xD2040000	SQL_INTEGER	Numbers 1234
32				0xD204	SQL_SMALLINT	Numbers 1234
33				0x31323334	SQL_NUMERIC	Unsigned
				0x31323344	SQL_NUMERIC	Signed, end, bit format
				0x41323334	SQL_NUMERIC	Signed, beginning, bit format
				0x2B31323334	SQL_NUMERIC	Signed, beginning, character format
				0x313233342B	SQL_NUMERIC	Signed, end, character format
34				0x012343	SQL_DECIMAL	Numbers 1234
35		0x04008250825182528253	SQL_VARNCHAR(4)	0x31323334	SQL_CHAR(4)	Characters 1234
36				0x040031323334	SQL_VARCHAR(4)	Characters 1234
37				0x8250825182528253	SQL_NCHAR	(Full-width) characters 1234
38				0x04008250825182528253	SQL_VARNCHAR	(Full-width) characters 1234

Item No.	Category	Input data	Input data example	Output data	Output data example	Remarks
39				0xD2040000	SQL_INTEGER	Numbers 1234
40				0xD204	SQL_SMALLINT	Numbers 1234
41				0x31323334	SQL_NUMERIC	Unsigned
				0x31323344	SQL_NUMERIC	Signed, end, bit format
				0x41323334	SQL_NUMERIC	Signed, beginning, bit format
				0x2B31323334	SQL_NUMERIC	Signed, beginning, character format
				0x313233342B	SQL_NUMERIC	Signed, end, character format
42				0x012343	SQL_DECIMAL	Numbers 1234
43	Numeric attributes	0xD2040000	SQL_INTEGER	0x31323334	SQL_CHAR(4)	Characters 1234
44				0x040031323334	SQL_VARCHAR(4)	Characters 1234
45				0x8250825182528253	SQL_NCHAR	(Full-width) characters 1234
46				0x04008250825182528253	SQL_VARNCHAR	(Full-width) characters 1234
47				0xD2040000	SQL_INTEGER	Numbers 1234
48				0xD204	SQL_SMALLINT	Numbers 1234
49				0x31323334	SQL_NUMERIC	Unsigned
				0x31323344	SQL_NUMERIC	Signed, end, bit format
				0x41323334	SQL_NUMERIC	Signed, beginning, bit format
				0x2B31323334	SQL_NUMERIC	Signed, beginning, character format
				0x313233342B	SQL_NUMERIC	Signed, end, character format
50				0x012343	SQL_DECIMAL	Numbers 1234
51		0xD204	SQL_SMALLINT	0x31323334	SQL_CHAR(4)	Characters 1234
52				0x040031323334	SQL_VARCHAR(4)	Characters 1234
53				0x8250825182528253	SQL_NCHAR	(Full-width) characters 1234

Item No.	Category	Input data	Input data example	Output data	Output data example	Remarks
54				0x04008250825182528253	SQL_VARNCHAR	(Full-width) characters 1234
55				0xD2040000	SQL_INTEGER	Numbers 1234
56				0xD204	SQL_SMALLINT	Numbers 1234
57				0x31323334	SQL_NUMERIC	Unsigned
				0x31323344	SQL_NUMERIC	Signed, end, bit format
				0x41323334	SQL_NUMERIC	Signed, beginning, bit format
				0x2B31323334	SQL_NUMERIC	Signed, beginning, character format
				0x313233342B	SQL_NUMERIC	Signed, end, character format
58				0x012343	SQL_DECIMAL	Numbers 1234
59		0x31323334	SQL_NUMERIC Unsigned	0x31323334	SQL_CHAR(4)	Characters 1234
60				0x040031323334	SQL_VARCHAR(4)	Characters 1234
61				0x82508251825282 53	SQL_NCHAR	(Full-width) characters 1234
62				0x04008250825182528253	SQL_VARNCHAR	(Full-width) characters 1234
63				0xD2040000	SQL_INTEGER	Numbers 1234
64				0xD204	SQL_SMALLINT	Numbers 1234
65				0x31323334	SQL_NUMERIC	Unsigned
				0x31323344	SQL_NUMERIC	Signed, end, bit format
				0x41323334	SQL_NUMERIC	Signed, beginning, bit format
				0x2B31323334	SQL_NUMERIC	Signed, beginning, character format
				0x313233342B	SQL_NUMERIC	Signed, end, character format
66				0x012343	SQL_DECIMAL	Numbers 1234
		0x31323344	SQL_NUMERIC	0x31323334	SQL_CHAR(4)	Characters 1234
				0x040031323334	SQL_VARCHAR(4)	Characters 1234

Item No.	Category	Input data	Input data example	Output data	Output data example	Remarks
			Signed, end, bit format	0x8250825182528253	SQL_NCHAR	(Full-width) characters 1234
				0x04008250825182528253	SQL_VARNCHAR	(Full-width) characters 1234
				0xD2040000	SQL_INTEGER	Numbers 1234
				0xD204	SQL_SMALLINT	Numbers 1234
				0x31323334	SQL_NUMERIC	Unsigned
				0x31323344	SQL_NUMERIC	Signed, end, bit format
				0x41323334	SQL_NUMERIC	Signed, beginning, bit format
				0x2B31323334	SQL_NUMERIC	Signed, beginning, character format
				0x313233342B	SQL_NUMERIC	Signed, end, character format
				0x012343	SQL_DECIMAL	Numbers 1234
		0x41323334	SQL_NUMERIC Signed, beginning, bit format	0x31323334	SQL_CHAR(4)	Characters 1234
				0x040031323334	SQL_VARCHAR(4)	Characters 1234
				0x8250825182528253	SQL_NCHAR	(Full-width) characters 1234
				0x04008250825182528253	SQL_VARNCHAR	(Full-width) characters 1234
				0xD2040000	SQL_INTEGER	Numbers 1234
				0xD204	SQL_SMALLINT	Numbers 1234
				0x31323334	SQL_NUMERIC	Unsigned
				0x31323344	SQL_NUMERIC	Signed, end, bit format
				0x41323334	SQL_NUMERIC	Signed, beginning, bit format
				0x2B31323334	SQL_NUMERIC	Signed, beginning, character format
				0x313233342B	SQL_NUMERIC	Signed, end, character format
				0x012343	SQL_DECIMAL	Numbers 1234

Item No.	Category	Input data	Input data example	Output data	Output data example	Remarks
		0x2B31323334	SQL_NUMERIC Signed, beginning, character format	0x31323334	SQL_CHAR(4)	Characters 1234
				0x040031323334	SQL_VARCHAR(4)	Characters 1234
				0x8250825182528253	SQL_NCHAR	(Full-width) characters 1234
				0x04008250825182528253	SQL_VARNCHAR	(Full-width) characters 1234
				0xD2040000	SQL_INTEGER	Numbers 1234
				0xD204	SQL_SMALLINT	Numbers 1234
				0x31323334	SQL_NUMERIC	Unsigned
				0x31323344	SQL_NUMERIC	Signed, end, bit format
				0x41323334	SQL_NUMERIC	Signed, beginning, bit format
				0x2B31323334	SQL_NUMERIC	Signed, beginning, character format
				0x313233342B	SQL_NUMERIC	Signed, end, character format
				0x012343	SQL_DECIMAL	Numbers 1234
		0x313233342B	SQL_NUMERIC Signed, end, character format	0x31323334	SQL_CHAR(4)	Characters 1234
				0x040031323334	SQL_VARCHAR(4)	Characters 1234
				0x8250825182528253	SQL_NCHAR	(Full-width) characters 1234
				0x04008250825182528253	SQL_VARNCHAR	(Full-width) characters 1234
				0xD2040000	SQL_INTEGER	Numbers 1234
				0xD204	SQL_SMALLINT	Numbers 1234
				0x31323334	SQL_NUMERIC	Unsigned
				0x31323344	SQL_NUMERIC	Signed, end, bit format
				0x41323334	SQL_NUMERIC	Signed, beginning, bit format
				0x2B31323334	SQL_NUMERIC	Signed, beginning, character format

Item No.	Category	Input data	Input data example	Output data	Output data example	Remarks
				0x313233342B	SQL_NUMERIC	Signed, end, character format
				0x012343	SQL_DECIMAL	Numbers 1234
67		0x012343	SQL_DECIMAL	0x31323334	SQL_CHAR(4)	Characters 1234
68				0x040031323334	SQL_VARCHAR(4)	Characters 1234
69				0x8250825182528253	SQL_NCHAR	(Full-width) characters 1234
70				0x04008250825182528253	SQL_VARNCHAR	(Full-width) characters 1234
71				0xD2040000	SQL_INTEGER	Numbers 1234
72				0xD204	SQL_SMALLINT	Numbers 1234
73				0x31323334	SQL_NUMERIC	Unsigned
				0x31323344	SQL_NUMERIC	Signed, end, bit format
				0x41323334	SQL_NUMERIC	Signed, beginning, bit format
				0x2B31323334	SQL_NUMERIC	Signed, beginning, character format
				0x313233342B	SQL_NUMERIC	Signed, end, character format
74				0x012343	SQL_DECIMAL	Numbers 1234
75		0x449a4000	SQL_REAL	0x31323334	SQL_CHAR(4)	Characters 1234
76				0x040031323334	SQL_VARCHAR(4)	Characters 1234
77				0x449a4000	SQL_REAL	Numbers 1234
78				0x4093480000000000	SQL_FLOAT	Numbers 1234
79		0x4093480000000000	SQL_FLOAT	0x31323334	SQL_CHAR(4)	Characters 1234
80				0x040031323334	SQL_VARCHAR(4)	Characters 1234
81				0x449a4000	SQL_REAL	Numbers 1234
82				0x4093480000000000	SQL_FLOAT	Numbers 1234
83	Date/	0xD9070701	SQL_DATE	0x323030392F30372F3031	SQL_CHAR(10)	Date 2009/07/01
84				0x0a00323030392F30372F3031	SQL_VARCHAR(10)	Date 2009/07/01

Item No.	Category	Input data	Input data example	Output data	Output data example	Remarks
85	time attributes			0xD9070701	SQL_DATE	
86		0x0C2238	SQL_TIME	0x31323A33343A3536	SQL_CHAR(8)	Time 12:34:56
87				0x080031323A33343A3536	SQL_VARCHAR(8)	Time 12:34:56
88				0x0C2238	SQL_TIME	
89		0xD90707010C2238	SQL_TIMESTAMP	0x323030392F30372F30312031323A33343A3536	SQL_CHAR(19)	Date/time 2009/07/01 12:34:56
90				0x1300323030392F30372F30312031323A33343A3536	SQL_VARCHAR(19)	Date/time 2009/07/01 12:34:56
91				0xD90707010C2238	SQL_TIMESTAMP	
92		0x00000000020000001234	SQL_LONGVARBINARY(2)	0x00000000020000001234	SQL_LONGVARBINARY(2)	BLOB format
93		0x0102030405	SQL_BINARY(5)	0x0102030405	SQL_BINARY(5)	
94		0x0a	SQL_BIT(8)	0x0a	SQL_BIT(8)	
			SQL_BIT(6)	0x08	SQL_BIT(6)	

1.4.3.5 Trim

This feature removes all arbitrary characters comprising one or more consecutive characters from a string. The following two types of trim can be used:

- Right trim

  This feature removes all arbitrary characters comprising one or more consecutive characters from the end of a string.

  This feature can be used when data has been extracted from the database to remove any spaces embedded at the end of strings in char-type attribute items.

- Left trim

  This feature removes all arbitrary characters comprising one or more consecutive characters from the beginning of a string.

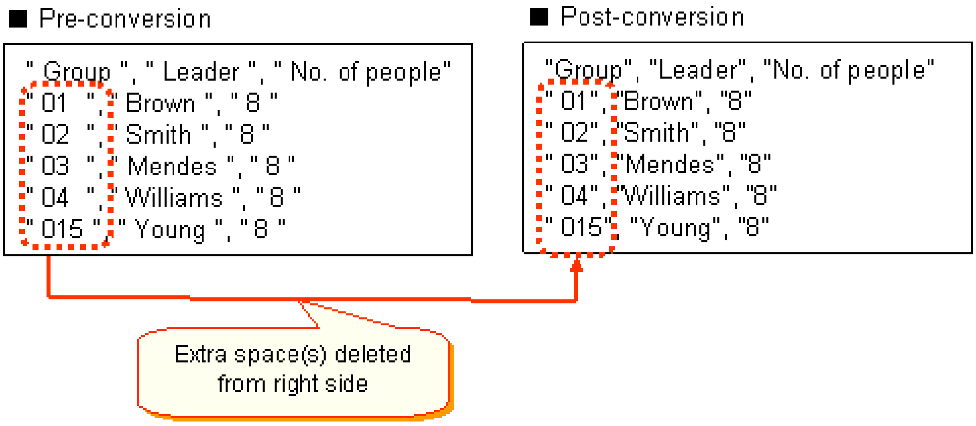
  This feature can be used to remove characters such as those at the beginning of right-aligned numeric characters when the user wishes to handle data as numeric values.

Note that items that can be trimmed should have string-type attributes and ASCII characters.

[Example] Right trim

The following example illustrates the removal of spaces from the end of a string.

The ends of numbers in each group contain spaces, and in the example below, the spaces attached to the end of the numbers will be removed.



■ Pre-conversion

```
" Group ", " Leader ", " No. of people"
" 01  ", " Brown ", " 8 "
" 02  ", " Smith ", " 8 "
" 03  ", " Mendes ", " 8 "
" 04  ", " Williams ", " 8 "
" 015 ", " Young ", " 8 "
```

■ Post-conversion

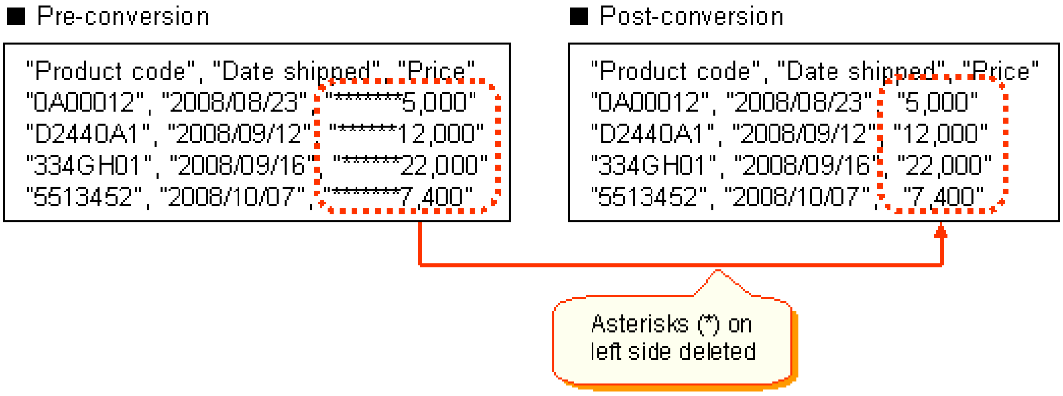
```
"Group", "Leader", "No. of people"
" 01", "Brown", "8"
" 02", "Smith", "8"
" 03", "Mendes", "8"
" 04", "Williams", "8"
" 015", "Young", "8"
```

Extra space(s) deleted from right side

[Example] Left trim

The following example illustrates the removal of characters from the beginning of a string.

In the example below, the asterisk ('*') characters will be removed from the beginning of the listed prices.



■ Pre-conversion

```
"Product code", "Date shipped", "Price"
"0A00012", "2008/08/23", "********5,000"
"D2440A1", "2008/09/12", "*******12,000"
"334GH01", "2008/09/16", "*******22,000"
"5513452", "2008/10/07", "********7,400"
```

■ Post-conversion

```
"Product code", "Date shipped", "Price"
"0A00012", "2008/08/23", "5,000"
"D2440A1", "2008/09/12", "12,000"
"334GH01", "2008/09/16", "22,000"
"5513452", "2008/10/07", "7,400"
```

Asterisks (*) on left side deleted

1.4.3.6 Padding

This feature is used to add 0 or more characters to a string. The following two types of padding can be used:

- Right padding

  This feature is used to add 0 or more consecutive arbitrary characters to the end of strings in item data. This action ensures the string will be the specified length.

  This feature can be used when spaces needs to be embedded at the end of strings in char-type attribute items when storing data in the database.

- Left padding

  This feature is used to add 0 or more consecutive arbitrary characters to the beginning of strings in item data. This action ensures the string will be the specified length.

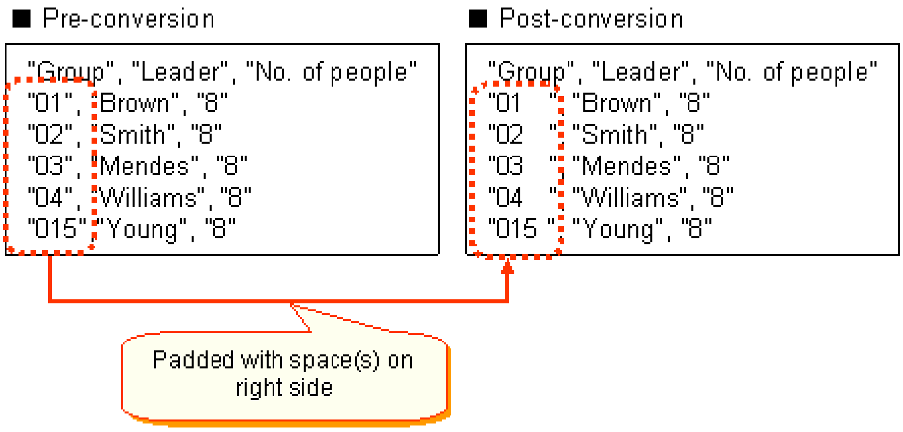
  This feature can be used when converting to right-aligned numeric characters.

Note that items that can be padded should have string-type attributes and ASCII characters.

[Example] Right padding

The following example illustrates the addition of spaces to the end of a string.

In the example below, spaces are added to the end of group numbers.



■ Pre-conversion

```
"Group", "Leader", "No. of people"
"01", "Brown", "8"
"02", "Smith", "8"
"03", "Mendes", "8"
"04", "Williams", "8"
"015","Young", "8"
```

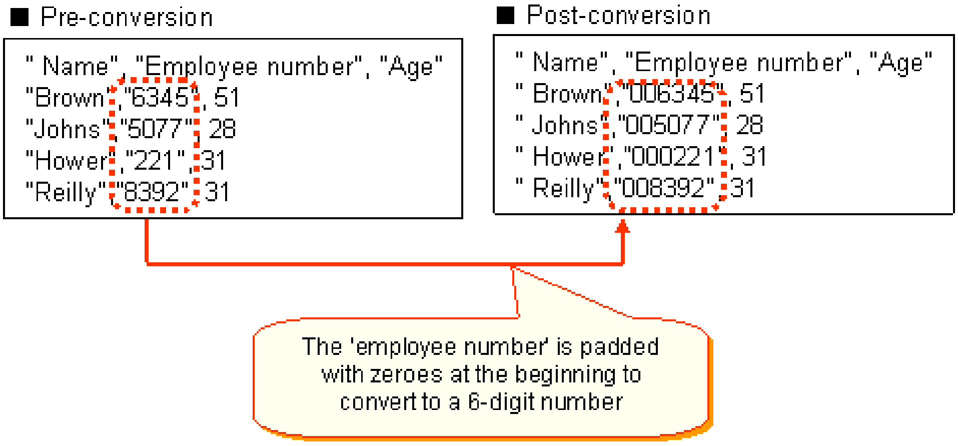
■ Post-conversion

```
"Group", "Leader", "No. of people"
"01  ", "Brown", "8"
"02  ", "Smith", "8"
"03  ", "Mendes", "8"
"04  ", "Williams", "8"
"015 ", "Young", "8"
```

[Example] Left padding

The following example illustrates the addition of characters to the beginning of a string.

In the example below, a '0' character is added to the beginning of employee numbers to convert these numbers to a length of 6 digits.



■ Pre-conversion

```
" Name", "Employee number", "Age"
"Brown", "6345", 51
"Johns", "5077", 28
"Hower", "221", 31
"Reilly", "8392", 31
```

■ Post-conversion

```
" Name", "Employee number", "Age"
" Brown", "006345", 51
" Johns", "005077", 28
" Hower", "000221", 31
" Reilly", "008392", 31
```

The 'employee number' is padded with zeroes at the beginning to convert to a 6-digit number

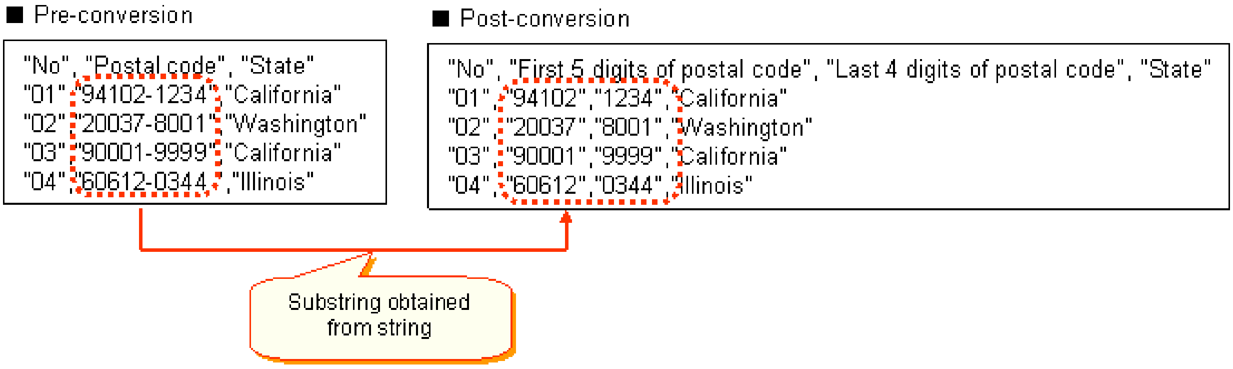
1.4.3.7 Substring Extraction

This feature enables partial acquisition of characters in strings.

[Example]

The following example illustrates a substring extraction:

In the example below, 9-digit US postal codes are split into the top 5 digits and bottom 4 digits.



■ Pre-conversion

```
"No", "Postal code", "State"
"01", "94102-1234", "California"
"02", "20037-8001", "Washington"
"03", "90001-9999", "California"
"04", "60612-0344", "Illinois"
```

■ Post-conversion

```
"No", "First 5 digits of postal code", "Last 4 digits of postal code", "State"
"01", "94102", "1234", "California"
"02", "20037", "8001", "Washington"
"03", "90001", "9999", "California"
"04", "60612", "0344", "Illinois"
```

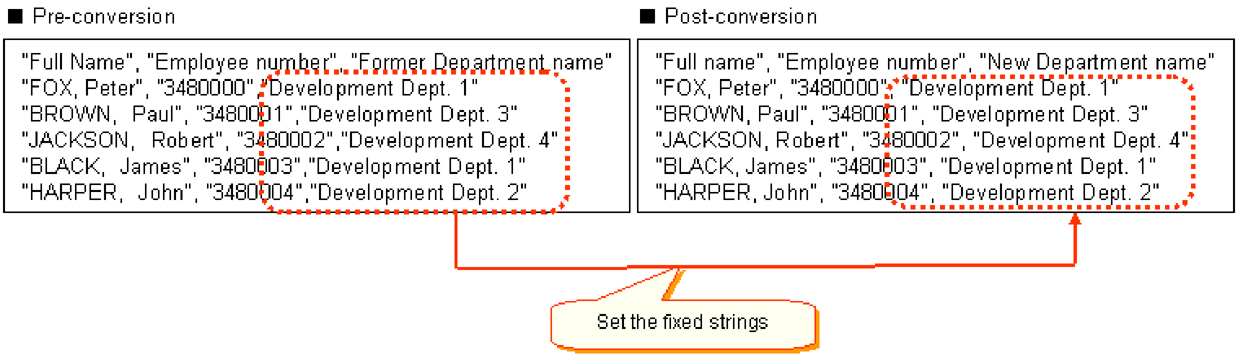
1.4.3.8 String Settings

Settings can be performed for fixed strings.

The following is an example of string settings:

[Example]

In the string settings example below, the string representing the department name will be set to 'Development Department 2'.



■ Pre-conversion

```
"Full Name", "Employee number", "Former Department name"
"FOX, Peter", "3480000","Development Dept. 1"
"BROWN,  Paul", "3480001","Development Dept. 3"
"JACKSON,  Robert", "3480002","Development Dept. 4"
"BLACK,  James", "3480003","Development Dept. 1"
"HARPER,  John", "3480004","Development Dept. 2"
```

■ Post-conversion

```
"Full name", "Employee number", "New Department name"
"FOX, Peter", "3480000", "Development Dept. 1"
"BROWN, Paul", "3480001", "Development Dept. 3"
"JACKSON, Robert", "3480002", "Development Dept. 4"
"BLACK, James", "3480003", "Development Dept. 1"
"HARPER, John", "3480004", "Development Dept. 2"
```

Note

- It is not possible to set only a NULL value or a half-width space.
- A string will be invalid even if a half-width space is set at the beginning of the string. Half-width spaces elsewhere within the string, or at the end of the string, will be valid.

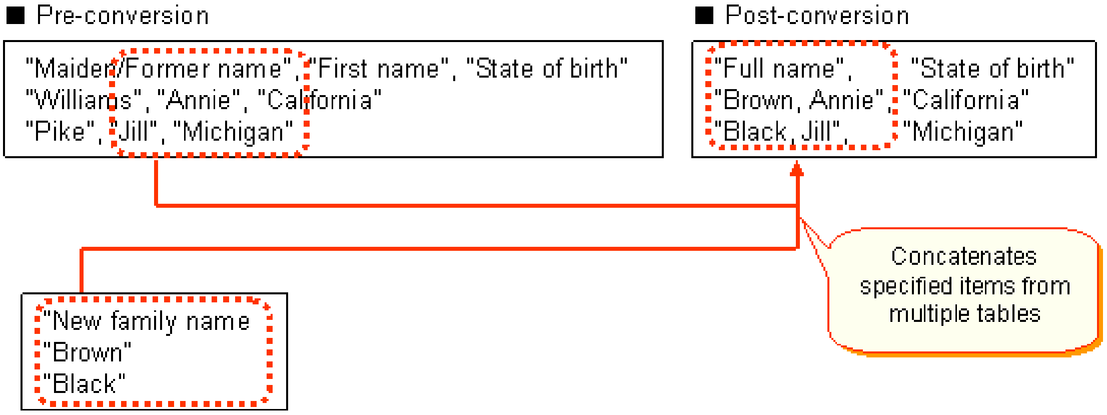
1.4.3.9 Joining Strings

This feature enables joined acquisition of characters in strings.

The following example illustrates some joined strings:

[Example 1]

In the example below, family names and first names present in separate tables will be joined to form full names.



■ Pre-conversion

```
"Maiden/Former name", "First name", "State of birth"
"Williams", "Annie", "California"
"Pike", "Jill", "Michigan"
```

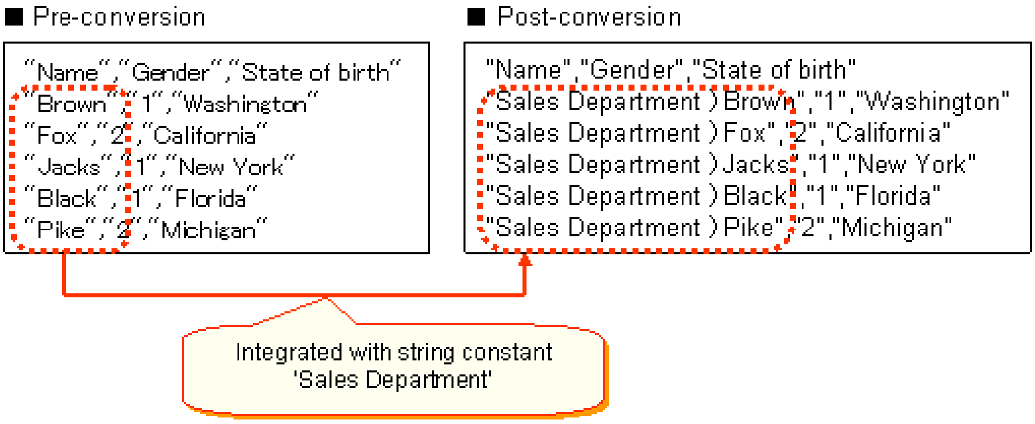
```
"New family name
"Brown"
"Black"
```

■ Post-conversion

```
"Full name",     "State of birth"
"Brown, Annie", "California"
"Black, Jill",  "Michigan"
```

[Example 2]

In the example below, each person's department name will be joined prior to their full name.



■ Pre-conversion

```
“Name”,“Gender”,“State of birth”
“Brown”,“1”,“Washington”
“Fox”,“2”,“California”
“Jacks”,“1”,“New York”
“Black”,“1”,“Florida”
“Pike”,“2”,“Michigan”
```

■ Post-conversion

```
"Name","Gender","State of birth"
"Sales Department）Brown","1","Washington"
"Sales Department）Fox","2","California"
"Sales Department）Jacks","1","New York"
"Sales Department）Black","1","Florida"
"Sales Department）Pike","2","Michigan"
```

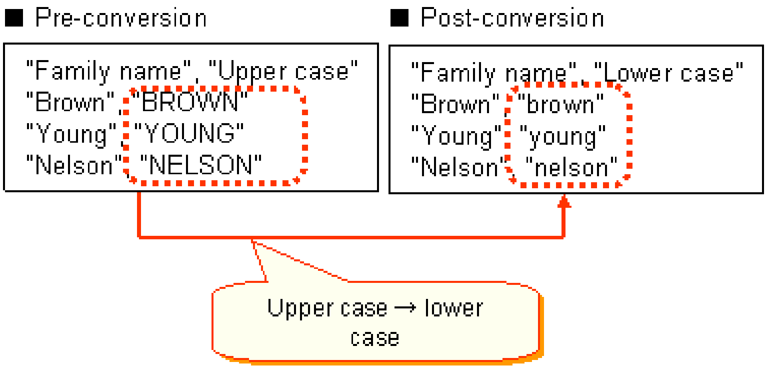
Integrated with string constant 'Sales Department'

1.4.3.10 Uppercase/Lowercase Conversion

This feature enables conversion from upper case characters to lower case, and from lower case characters to upper case.

[Example]

Converting the 'Alphabetic reading' of an item from upper case characters to lower case.



■ Pre-conversion

```
"Family name", "Upper case"
"Brown","BROWN"
"Young","YOUNG"
"Nelson","NELSON"
```

■ Post-conversion

```
"Family name", "Lower case"
"Brown","brown"
"Young","young"
"Nelson","nelson"
```

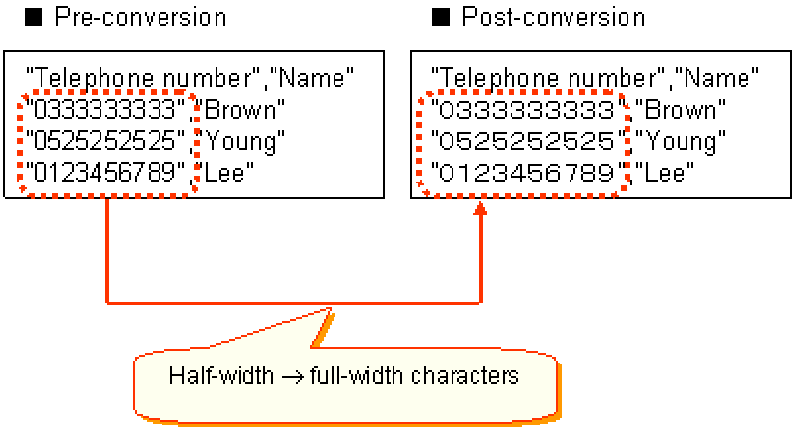
Upper case → lower case

1.4.3.11 Half-width/Full-width Conversion

This feature enables conversion from full-width characters to half-width, and from half-width characters to full-width.

[Example]

Converting the 'Telephone number' item from half-width to full-width characters



■ Pre-conversion

```
"Telephone number","Name"
"0333333333","Brown"
"0525252525","Young"
"0123456789","Lee"
```

■ Post-conversion

```
"Telephone number","Name"
"０３３３３３３３３３","Brown"
"０５２５２５２５２５","Young"
"０１２３４５６７８９","Lee"
```

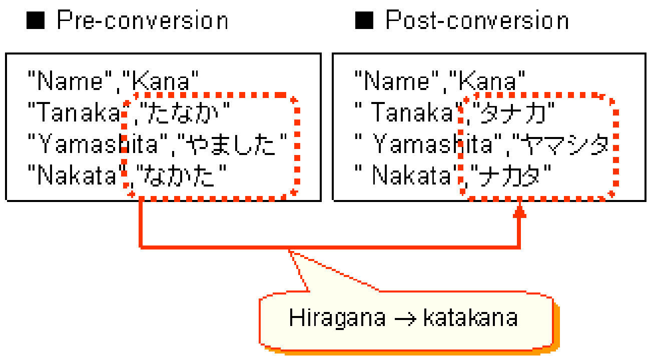
Half-width → full-width characters

1.4.3.12 Hiragana/Katakana Conversion

This feature enables conversion from hiragana characters to katakana, and from katakana characters to hiragana.

[Example]

Converting the 'Kana' item from hiragana to katakana characters



■ Pre-conversion

```
"Name","Kana"
"Tanaka","たなか"
"Yamashita","やました"
"Nakata","なかた"
```

■ Post-conversion

```
"Name","Kana"
" Tanaka","タナカ"
" Yamashita","ヤマシタ"
" Nakata","ナカタ"
```

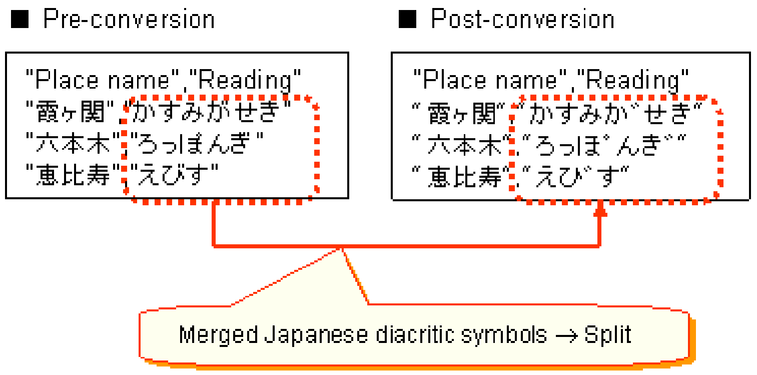
Hiragana → katakana

1.4.3.13 Japanese Diacritic Symbol Conversion

This feature enables the splitting and joining of Japanese diacritic symbols in full-width kana characters.

[Example]

'か゛' and 'ほ゜' when splitting, and 'か゛' and 'ほ゜' when joining



■ Pre-conversion

```
"Place name","Reading"
"霞ヶ関","かすみがせき"
"六本木","ろっぽんぎ"
"恵比寿","えびす"
```

■ Post-conversion

```
"Place name","Reading"
" 霞ヶ関"," かすみか゛せき"
" 六本木"," ろっほ゜んき゛"
" 恵比寿"," えひ゛す"
```

Merged Japanese diacritic symbols → Split

1.4.3.14 Four Arithmetic Operations

This feature enables the four arithmetic operations 'Addition', 'Subtraction', 'Multiplication' and 'Division' to be performed.

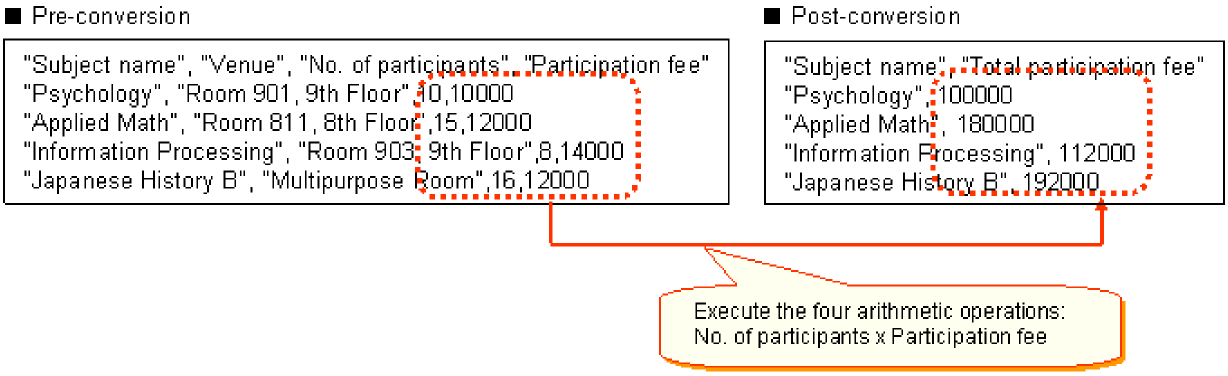
The four arithmetic operations feature facilitates calculations to be performed using either one or multiple numeric attribute items. Similarly, the output item will also be a numeric item attribute.

The four operators that may be used are addition '+', subtraction '-', multiplication '*' and division '/'. Furthermore, as with regular arithmetic expressions, '( )' can also be used to give priority to an expression.

Operations can also be performed for collected data or for data that has had some kind of conversion process applied to it.

[Example]

The example below illustrates a calculation multiplying the number of participants by the participation fee to output the total participation fee.


■ Pre-conversion
■ Post-conversion
Execute the four arithmetic operations:
No. of participants x Participation fee

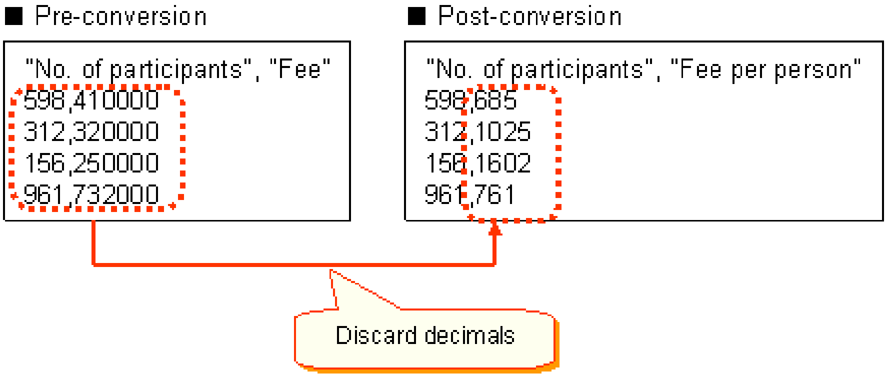
```
"Subject name", "Venue", "No. of participants", "Participation fee"
"Psychology", "Room 901, 9th Floor",10,10000
"Applied Math", "Room 811, 8th Floor",15,12000
"Information Processing", "Room 903, 9th Floor",8,14000
"Japanese History B", "Multipurpose Room",16,12000
```

```
"Subject name", "Total participation fee"
"Psychology", 100000
"Applied Math", 180000
"Information Processing", 112000
"Japanese History B", 192000
```

It is also possible to truncate the value of the calculation result. The location at which the value should be truncated may be specified at either the integer part or the decimal part.

[Example]

The example below illustrates a value with the decimal part truncated after calculating the participation fee per person.



```
"No. of participants", "Fee"
598,410000
312,320000
156,250000
961,732000
```

```
"No. of participants", "Fee per person"
598,685
312,1025
156,1602
961,761
```

Note

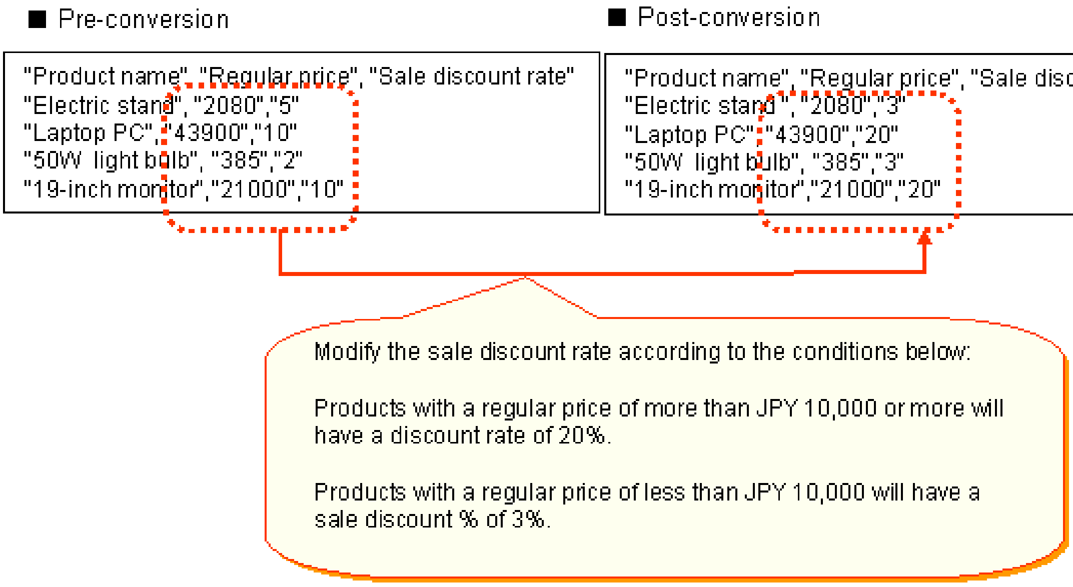
- The numeric range is a maximum of 36 digits, comprising 18 digits in the integer part and 18 digits in the decimal part. If the integer part exceeds 18 digits, with the exception of any zeroes at the beginning, an error will be output.

1.4.3.15 Item Conditions

Setting conditions for the items enables settings for fixed values and data operation processes to be performed, based on these conditions.

[Example]

The example below illustrates changes to a product discount percentage, based on the conditions.



■ Pre-conversion

```
"Product name", "Regular price", "Sale discount rate"
"Electric stand", "2080","5"
"Laptop PC", "43900","10"
"50W light bulb", "385","2"
"19-inch monitor","21000","10"
```

■ Post-conversion

```
"Product name", "Regular price", "Sale discount rate"
"Electric stand", "2080","3"
"Laptop PC", "43900","20"
"50W light bulb", "385","3"
"19-inch monitor","21000","20"
```

Modify the sale discount rate according to the conditions below:

Products with a regular price of more than JPY 10,000 or more will have a discount rate of 20%.

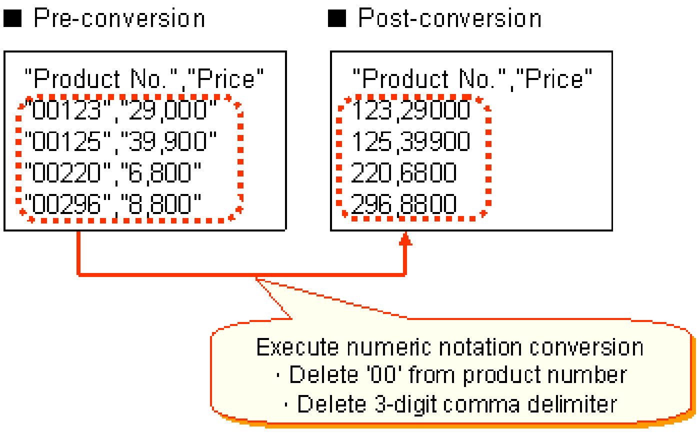
Products with a regular price of less than JPY 10,000 will have a sale discount % of 3%.

1.4.3.16 Numeric Notation Conversion

This feature converts character attributes to numeric attributes by deleting embedded zeroes and comma delimiters (positioners inserted every 3 digits in numbers).

[Example]

Delete the zeroes embedded in the 5-digit product number string and output a 3-digit numeric value. A further example illustrates a price notated as 3-digit comma-delimited, from which the comma delimiter is removed to convert to a numeric value.



■ Pre-conversion

```
"Product No.","Price"
"00123","29,000"
"00125","39,900"
"00220","6,800"
"00296","8,800"
```

■ Post-conversion

```
"Product No.","Price"
123,29000
125,39900
220,6800
296,8800
```

Execute numeric notation conversion
･Delete '00' from product number
･Delete 3-digit comma delimiter

1.4.3.17 Date and Time Format Conversion

Date and time-type attributes are notated as 'YYYY-MM-DD hh:mm:ss'.

The date and time format conversion feature is used to unify the format of differing types of date and time data.

The formats for data, both pre-conversion and post-conversion, are listed below.

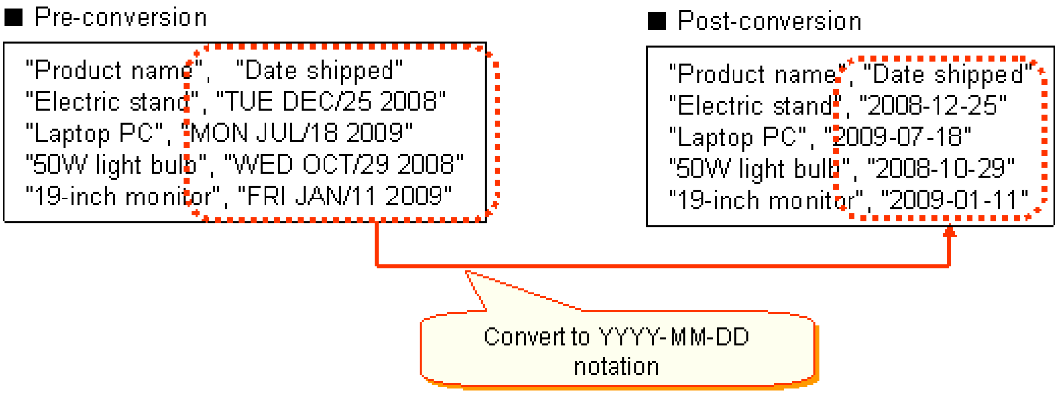
Table 1.6 Table of pre- and post-conversion data formats for date and time formats

Category	Pre-conversion data format	Post-conversion data format in Information Integrator
Year	30~29 (Meaning 1930~2029)	YYYY (Year in 4 digits)
	1~9999	

Category	Pre-conversion data format	Post-conversion data format in Information Integrator
Month	1~12	MM (Month in 2 digits)
	JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER	
	January, February, March, April, May, June, July, August, September, October, November, December	
	JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC	
	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec	
Date	1~31	DD (Date in 2 digits)
Day	SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY	
	Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday	
	SUN, MON, TUE, WED, THU, FRI, SAT	
	Sun, Mon, Tue, Wed, Thu, Fri, Sat	
Hour	0~23	hh (hour in 2 digits)
Minutes	0~59	mm (minutes in 2 digits)
Seconds	0~59	ss (seconds in 2 digits)

[Example]

The following example illustrates the conversion of the shipping date format to 'YYYY-MM-DD':



■ Pre-conversion

```
"Product name",  "Date shipped"
"Electric stand", "TUE DEC/25 2008"
"Laptop PC", "MON JUL/18 2009"
"50W light bulb", "WED OCT/29 2008"
"19-inch monitor", "FRI JAN/11 2009"
```

■ Post-conversion

```
"Product name", "Date shipped"
"Electric stand", "2008-12-25"
"Laptop PC", "2009-07-18"
"50W light bulb", "2008-10-29"
"19-inch monitor", "2009-01-11"
```

1.4.3.18 Obtain Date and Time Fragment

It is possible to obtain one of the following elements from the date and time data: 'Year', 'Month', 'Date', 'hours', 'minutes', and 'seconds'.

The following 2 methods can be used to obtain a date and time fragment.

- Obtaining a piece of data from the date and time data

  Example: To obtain 'Year'

  2009-07-10 -> 2009

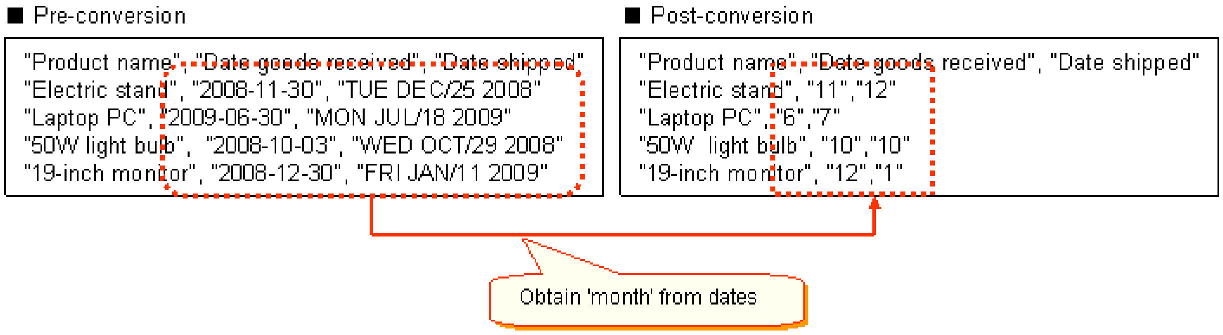
- To obtain data for a certain specified piece of data

  Example: Obtain all data for the month of July

Prior to obtaining	After obtaining
2008-07-22 2009-06-12 2009-07-11 2009-08-01	2008-07-22 2009-07-11

[Example]

The following example illustrates all 'month' data being obtained from both 'YYYY-MM-DD' format and 'DAY MON/DD YYYY' format.



■ Pre-conversion

```
"Product name", "Date goods received", "Date shipped"
"Electric stand", "2008-11-30", "TUE DEC/25 2008"
"Laptop PC", "2009-06-30", "MON JUL/18 2009"
"50W light bulb",  "2008-10-03", "WED OCT/29 2008"
"19-inch monitor", "2008-12-30", "FRI JAN/11 2009"
```

■ Post-conversion

```
"Product name", "Date goods received", "Date shipped"
"Electric stand", "11","12"
"Laptop PC", "6","7"
"50W  light bulb", "10","10"
"19-inch monitor", "12","1"
```

Note

- After obtaining the data, the notation format is converted to 'YYYY-MM-DD hh:mm:ss'. Refer to "1.4.3.17 Date and Time Format Conversion" for more information.
- It is not possible to obtain fragments for days of the week.
- Only 1 item can be obtained at any one time.

1.4.3.19 Notes when Editing Items

- When there is no data present, the conversion results will be treated as a NULL.
- When NULL data is included in the operation items, the operations results will be NULL. However, it is also possible to select whether NULL in a parameter will be treated as a 0.
- When specifying the format, the content of the pre-conversion data must match the data format exactly. If it does not match the specified data format, the conversion results will be treated as a NULL.

1.4.4 Edit Records

1.4.4.1 Filtering

Pre-filtered data is categorized according to the filtering conditions.

The following types of conditions may be filtered:

Table 1.7 Types of conditions that may be filtered

Search attribute	Search condition	Type of pattern search	Pattern search condition (*1)	Content
String search	Partial string match (Pattern search)	String search (Search full-width/ half-width character strings)	Match string	Extracts records that include the specified keyword.
			Prefix match	Extracts records that begin with the specified keyword.

Search attribute	Search condition	Type of pattern search	Pattern search condition (*1)	Content
			Suffix match	Extracts records that end with the specified keyword.
			Arbitrary character	Extracts records in which the 2 specified keywords are present before and after an arbitrary character.
			Character interval	Extracts records in which a character(s) within the specified number of characters is present between the 2 specified keywords. The number of characters that may be specified ranges from 0 to 1024.
			Partial character	Extracts records that include one of the specified characters between the 2 specified keywords.
			Character range	Extracts records within a specified range that contain a specified sub-string in the keyword.
			Numeric range	Extracts records in which a portion of characters in the specified keyword include one of the numeric value strings within the specified range.
		Word search (searching English words)	Word match	Extracts records that include the specified keyword. A record with consecutive keywords can be extracted by specifying keywords consecutively.
			Word interval	Records can be extracted that contain a phrase between two specified keywords, where the phrase comprises the specified number of words (or less) and includes arbitrary words.
	Perfect string match	-	-	
	Character code comparison	-	-	
Numeric search	Numeric match	-	-	
	Size of numeric value	-	-	
Date and time search	Date match	-	-	
	Date size	-	-	

Note: Multiple conditions can be combined together and extracted by joining the pattern search conditions of logical operators '&' (logical conjunction), '|'(logical disjunction) and '~'(negation). It is also possible to give an order of priority by specifying a period '.'.

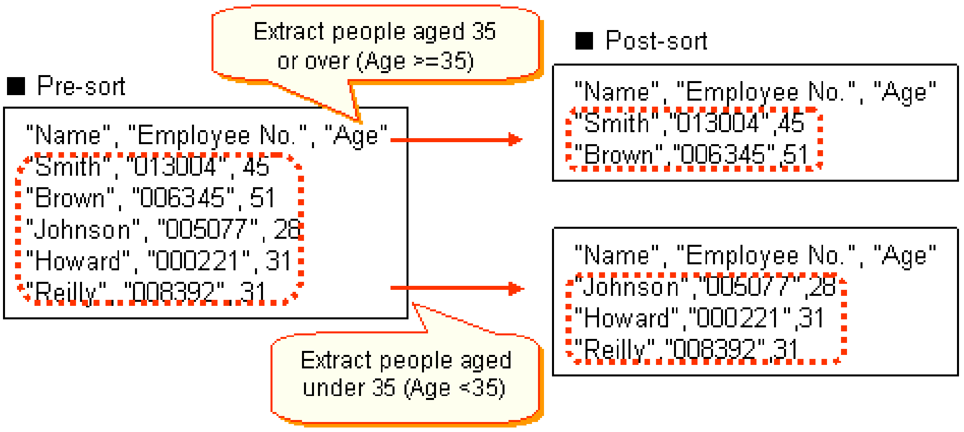
Point

How to change the appended enclosing characters

It is not possible to add or remove enclosing characters to/from data when filtering. To add or delete enclosing characters to/from filtered data, implement a conversion process (add or delete enclosing characters) for the target data.

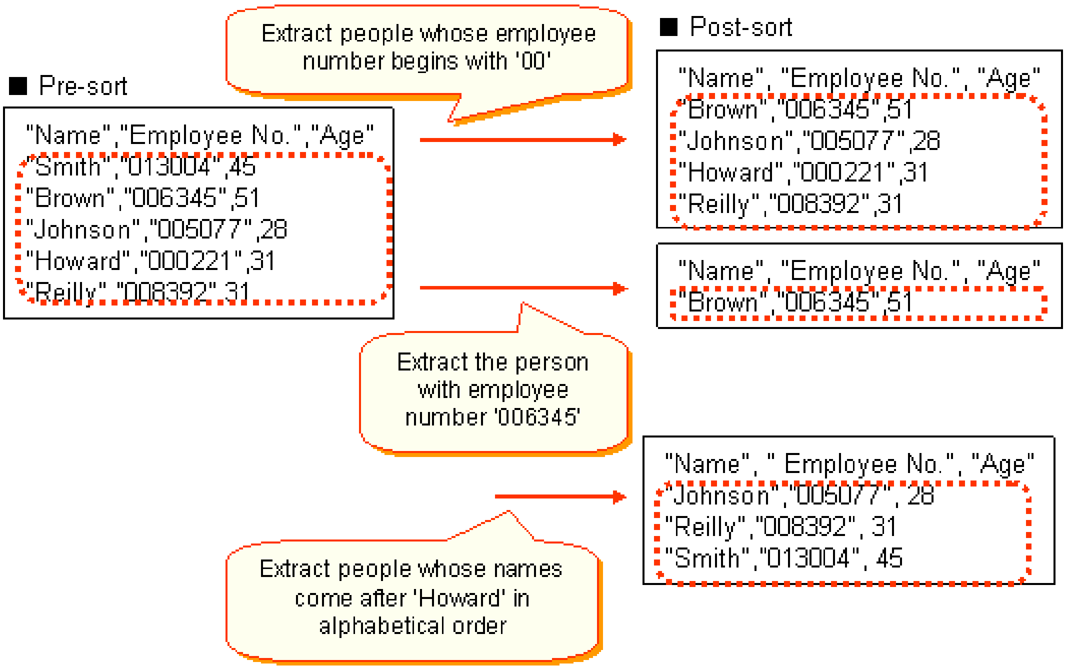
[Example 1] Numeric search

The example below illustrates a size comparison using a numeric keyword and extracts records that satisfies the conditions.



[Example 2] String search

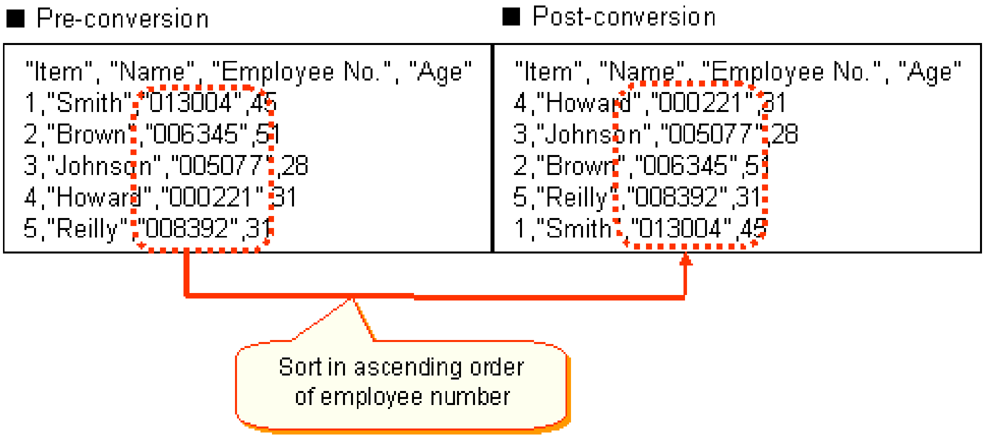
The example below illustrates the extraction of records that match a specified string keyword perfectly or partially, or that are larger or smaller than the specified keyword character code (i.e., come before or after the specified keyword character in terms of alphabetical order).


Extract people whose employee number begins with '00'
■ Pre-sort
"Name","Employee No.","Age"
"Smith","013004",45
"Brown","006345",51
"Johnson","005077",28
"Howard","000221",31
"Reilly","008392",31
■ Post-sort
"Name", "Employee No.", "Age"
"Brown","006345",51
"Johnson","005077",28
"Howard","000221",31
"Reilly","008392",31
"Name", "Employee No.", "Age"
"Brown","006345",51
Extract the person with employee number '006345'
"Name", " Employee No.", "Age"
"Johnson","005077", 28
"Reilly","008392", 31
"Smith","013004", 45
Extract people whose names come after 'Howard' in alphabetical order

1.4.4.2 Sort

Arbitrary items can be specified in the sort key and then sorted in ascending or descending order.

The sorting method differs according to the attributes of the item specified in the sort key.



■ Pre-conversion

```
"Item", "Name", "Employee No.", "Age"
1,"Smith","013004",45
2,"Brown","006345",51
3,"Johnson","005077",28
4,"Howard","000221",31
5,"Reilly","008392",31
```

■ Post-conversion

```
"Item", "Name", "Employee No.", "Age"
4,"Howard","000221",31
3,"Johnson","005077",28
2,"Brown","006345",51
5,"Reilly","008392",31
1,"Smith","013004",45
```

Sort in ascending order of employee number

1.4.4.3 Joining (Join, Union)

Multiple data items can be merged using the join condition. Note that it is possible to join multiple items of master data at once. A maximum of 128 items of master data can be joined.

The following types of joins can be performed:

Table 1.8 Types of joins

Feature name	Merging type	Merging condition	Feature description
Join (Horizontal join)	INNER JOIN	Perfect match (*1)	Items will be merged when the merge condition is true. Records will be deleted when the merge condition is false.
		Partial match (*2)	
	OUTER JOIN	Perfect match (*1)	Items will be merged when the merge condition is true. When the merge condition is false, records will be output but the joining master item will return a NULL value.
		Partial match (*2)	
Union (Vertical join)	UNION	Perfect match	Records will be merged. Duplicate records will be deleted.
	UNION ALL	-	Records will be merged. Duplicate records will be output.

*1: A perfect match occurs when the join keys match with each other.

*2: A partial match occurs when the join key of the master data is included in the join key of the transaction data. When there are multiple items of data that satisfy a partial match, conditions will only be returned as true for the number of items in the output data and will be merged.

Point

Maximum number of merge conditions that may be set for join

The maximum number of merge conditions that may be set for each individual master data item to be joined is as described below.

- When the merge condition is a perfect match:

  Up to a maximum of 8 join conditions may be specified for each individual master data item.

- When the merge condition is a partial match

  Up to a maximum of 1 join condition may be specified for each individual master data item.

Each merge type is described below.

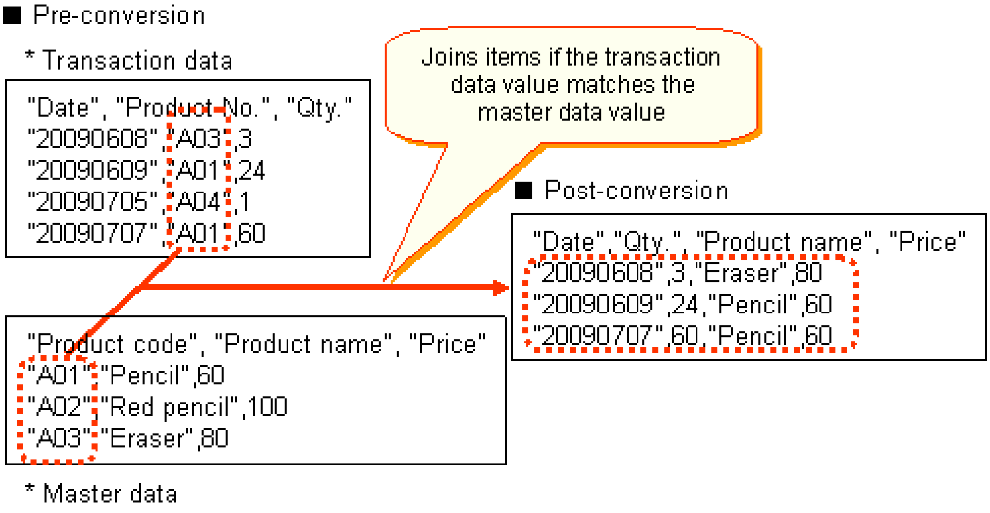
INNER JOIN

Items will be joined when master data strings and transaction data strings are a perfect match or a partial match.

No records will be output for INNER JOIN merging types when the transaction data is not present in the master data.

[Example]

The example below illustrates items being joined when the transaction data 'Product No' and Master Data 'Product Code' are entered as keys. (Perfect match example)



■ Pre-conversion
* Transaction data
Joins items if the transaction data value matches the master data value
■ Post-conversion
* Master data

```
"Date", "Product No.", "Qty."
"20090608","A03",3
"20090609","A01",24
"20090705","A04",1
"20090707","A01",60
```

```
"Product code", "Product name", "Price"
"A01","Pencil",60
"A02","Red pencil",100
"A03","Eraser",80
```

```
"Date","Qty.", "Product name", "Price"
"20090608",3,"Eraser",80
"20090609",24,"Pencil",60
"20090707",60,"Pencil",60
```

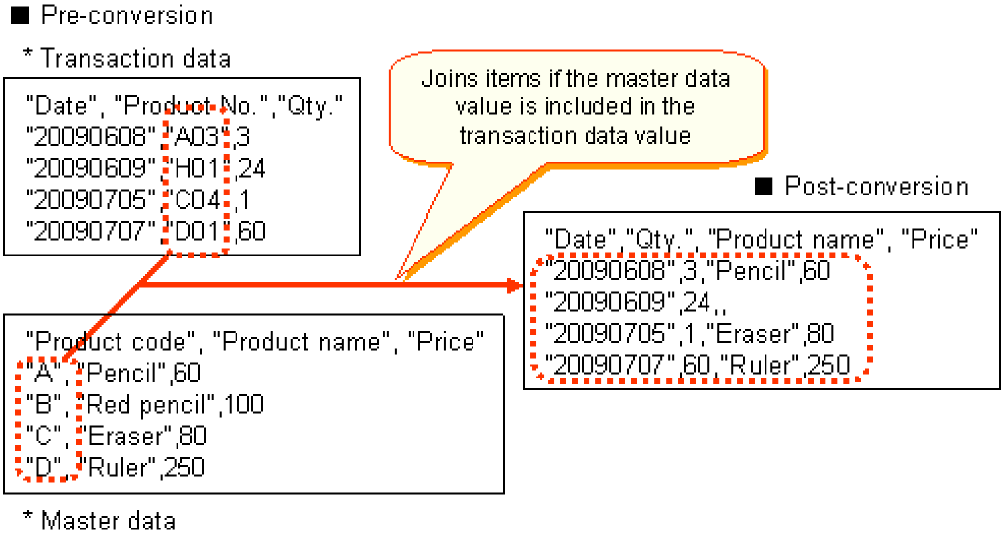
OUTER JOIN

Items will be joined when master data strings and transaction data strings are a perfect match or a partial match.

A NULL value will be returned for the master data for INNER JOIN merge types when the transaction data is not present in the master data.

[Example]

The example below illustrates items being joined when the transaction data 'Product No' and Master Data 'Product Code' are specified as keys. (Partial match example)



■ Pre-conversion
* Transaction data
Joins items if the master data value is included in the transaction data value
■ Post-conversion
* Master data

```
"Date", "Product No.","Qty."
"20090608","A03",3
"20090609","H01",24
"20090705","C04",1
"20090707","D01",60
```

```
"Product code", "Product name", "Price"
"A","Pencil",60
"B","Red pencil",100
"C","Eraser",80
"D","Ruler",250
```

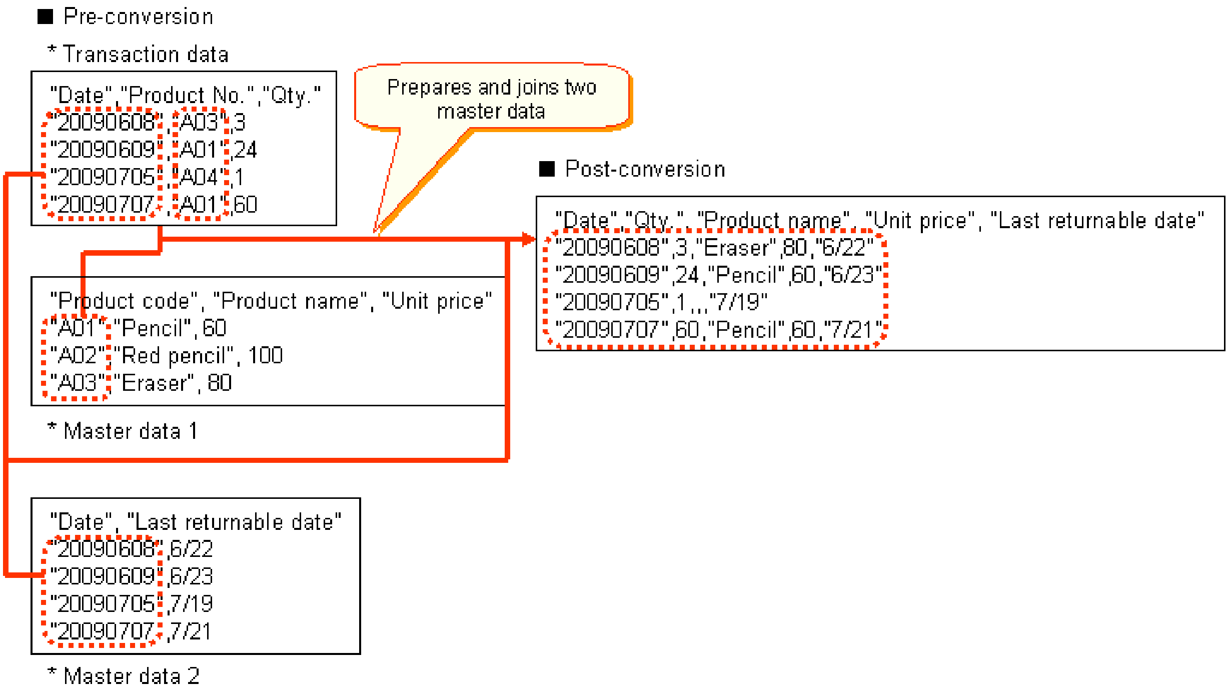
```
"Date","Qty.", "Product name", "Price"
"20090608",3,"Pencil",60
"20090609",24,,
"20090705",1,"Eraser",80
"20090707",60,"Ruler",250
```

Supplementary information on joins: When preparing multiple master data items

Three or more joins can be performed by preparing two or more master data items. Note that in this case, at least one join condition is required for each master data item.

[Example]

The example below illustrates items being joined when either the transaction data 'Product No' and master data 'Product Code', or the transaction data 'Date' and master data 'Date' are specified as keys.



```
"Date","Product No.","Qty."
"20090608","A03",3
"20090609","A01",24
"20090705","A04",1
"20090707","A01",60
```

```
"Product code", "Product name", "Unit price"
"A01","Pencil", 60
"A02","Red pencil", 100
"A03","Eraser", 80
```

```
"Date", "Last returnable date"
"20090608",6/22
"20090609",6/23
"20090705",7/19
"20090707",7/21
```

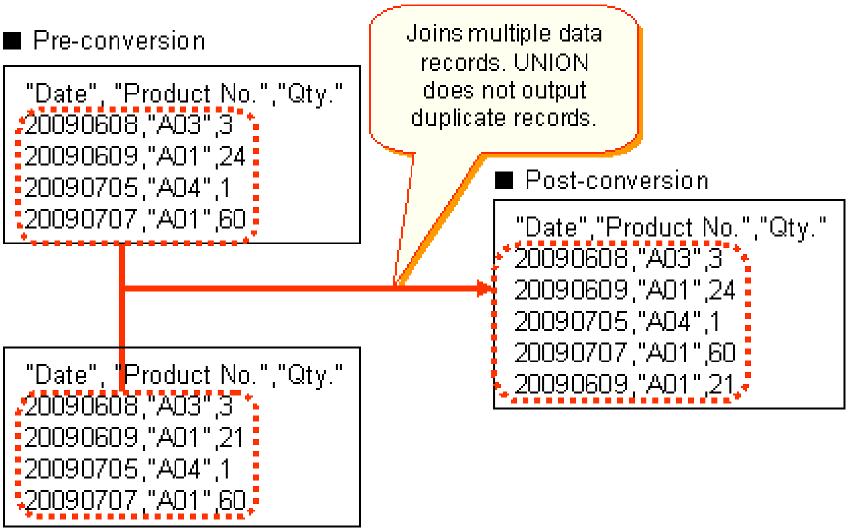
```
"Date","Qty.","Product name","Unit price", "Last returnable date"
"20090608",3,"Eraser",80,"6/22"
"20090609",24,"Pencil",60,"6/23"
"20090705",1,,,"7/19"
"20090707",60,"Pencil",60,"7/21"
```

UNION

This merge type joins multiple transaction data records. Duplicate records are not output for UNION merging.

[Example]

The example below illustrates 2 transaction data records being joined.


■ Pre-conversion
Joins multiple data records. UNION does not output duplicate records.
■ Post-conversion

```
"Date", "Product No.","Qty."
20090608,"A03",3
20090609,"A01",24
20090705,"A04",1
20090707,"A01",60
```

```
"Date", "Product No.","Qty."
20090608,"A03",3
20090609,"A01",21
20090705,"A04",1
20090707,"A01",60
```

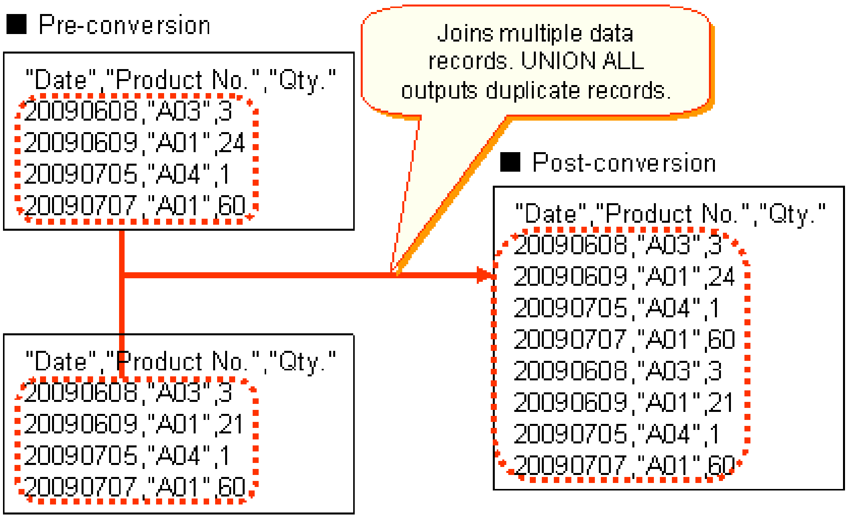
```
"Date","Product No.","Qty."
20090608,"A03",3
20090609,"A01",24
20090705,"A04",1
20090707,"A01",60
20090609,"A01",21
```

UNION ALL

This merge type joins multiple transaction data records. Duplicate records are output for UNION ALL merging.

[Example]

The example below illustrates 2 transaction data records being joined.



Joins multiple data records. UNION ALL outputs duplicate records.

```
"Date","Product No.","Qty."
20090608,"A03",3
20090609,"A01",24
20090705,"A04",1
20090707,"A01",60
```

```
"Date","Product No.","Qty."
20090608,"A03",3
20090609,"A01",21
20090705,"A04",1
20090707,"A01",60
```

```
"Date","Product No.","Qty."
20090608,"A03",3
20090609,"A01",24
20090705,"A04",1
20090707,"A01",60
20090608,"A03",3
20090609,"A01",21
20090705,"A04",1
20090707,"A01",60
```

1.4.4.4 Aggregate

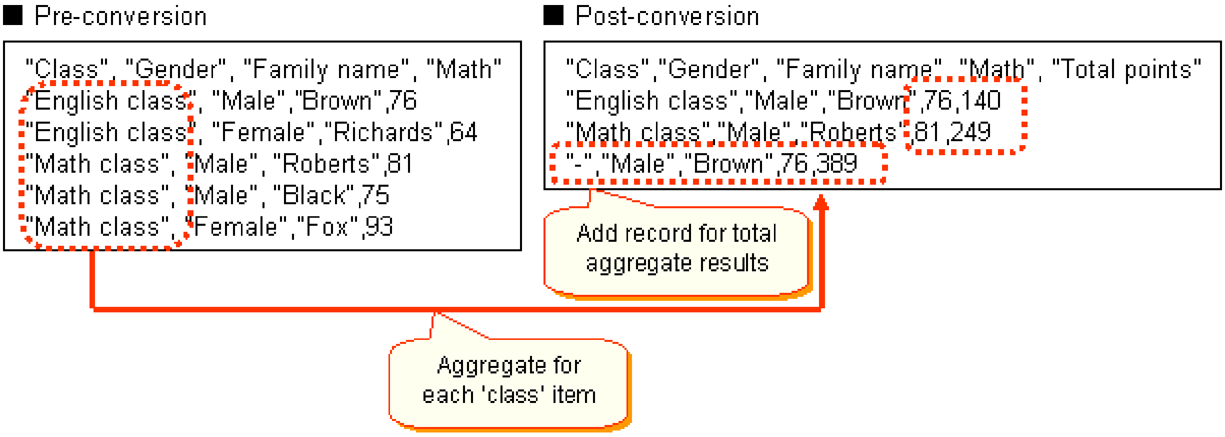
This feature groups data and outputs the total value of specified items. It is also possible to group multiple items and output the total value.

The aggregate feature can also be used in cases such as when outputting to a form.

[Example 1]

The following example illustrates aggregating the value of 'Math' for each 'Class' item.

If items which are not aggregate targets are output at this time, the value of the first record in each group will be displayed.



```
"Class", "Gender", "Family name", "Math"
"English class", "Male","Brown",76
"English class", "Female","Richards",64
"Math class", "Male", "Roberts",81
"Math class", "Male", "Black",75
"Math class", "Female","Fox",93
```

```
"Class","Gender", "Family name", "Math", "Total points"
"English class","Male","Brown",76,140
"Math class","Male","Roberts",81,249
"-","Male","Brown",76,389
```

Add record for total aggregate results

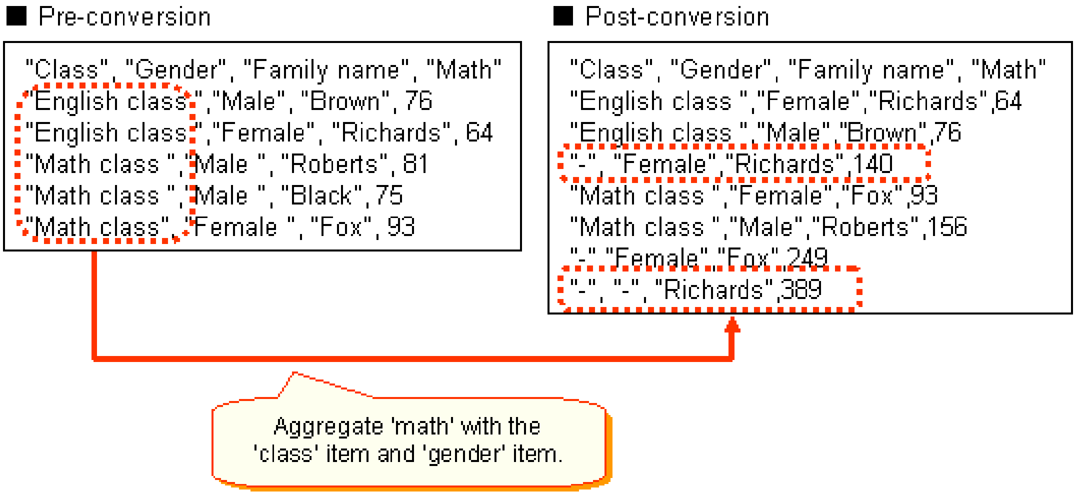
Aggregate for each 'class' item

[Example 2]

The example below illustrates specification of multiple units to be aggregated.

Data is grouped according to the first specified key, and then grouped further according to the next specified key.

In the following examples, it is grouped by "Gender" items after it is grouped by "Class" items, displays a by sex subtotal of each class, and the mathematics total point that outputs statistics line in addition is displayed.



■ Pre-conversion

```
"Class", "Gender", "Family name", "Math"
"English class","Male", "Brown", 76
"English class","Female", "Richards", 64
"Math class ","Male ", "Roberts", 81
"Math class ","Male ", "Black", 75
"Math class", "Female ", "Fox", 93
```

■ Post-conversion

```
"Class", "Gender", "Family name", "Math"
"English class ","Female","Richards",64
"English class ","Male","Brown",76
"-", "Female","Richards",140
"Math class ","Female","Fox",93
"Math class ","Male","Roberts",156
"-","Female","Fox",249
"-", "-", "Richards",389
```

Aggregate 'math' with the 'class' item and 'gender' item.

Point

- The aggregation results for the upper group specified in the group key can be output arbitrarily.
- The records for all aggregation results can be output arbitrarily.
- The upper group and all aggregation results can be output with one record added.

1.5 Execute Remote Jobs

1.5.1 Execute Remote Jobs

The 'Execute remote jobs' feature starts up jobs on a remote system. Jobs are started by executing the utility command.

Products integrated with 'Execute remote jobs'

To start a job on a remote system, it is imperative that the integrated product(s) on the target remote system support the 'Execute remote jobs' function.

Information Integrator can request jobs to be started on remote systems for the integrated products listed below. Note that it is not possible for the local system to receive job execution results for remote system integrated products with the exception of Linkexpress.

- Linkexpress
- Linkexpress File Transfer
- DTS
- HICS

The integrated products at the remote system that can request jobs to be started on Information Integrator are listed below. Note that it is not possible to send notification of job execution results to the local system for remote system integrated products with the exception of Linkexpress.

- Linkexpress
- Linkexpress File Transfer
- DTS
- HICS

Remarks: Refer to the manual of the relevant integrated product to obtain information on communication protocol(s) for integrated products that support the 'Execute remote jobs' function.

Notes on 'Execute remote jobs'

- 'Execute remote jobs' cannot be used when the remote job is a dialog-type job.
- 'Execute remote jobs' cannot be used when using SAN or HULFT as the communication protocol.

The notes for each remote system OS are described below.

- Windows
    - 'Execute remote jobs' cannot be used when the remote job fits any of the following descriptions:
        - A job with an application window open (such as notepad)
        - A job that does not operate normally when the AT command is executed
        - A job that does not operate normally when the schtasks command is executed
        - A shell script created with Windows PowerShell
    - It is not possible to access resources such as network drives or printers from programs that have been executed using 'Execute remote jobs'.
    - When specifying the time to receive job results as the time that this command should be restored, and starting up a remote job accordingly, those jobs started by this command will be required to send notification of job results to Information Integrator. The command will not be restored unless results notification is sent. Furthermore, if notification of any value other than '0' is received from the job as a job result, the command will return a '1' as the returned value. In this case, the user should determine the job execution results from the message output by the command.
    - Select one of the following options to use as the timing to restore the 'Execute remote jobs' command.
        - When job results are received for the remote job
        - When the remote job is started

      Ensure that notification of job results is sent to Information Integrator from the remote system when restoring 'Execute remote jobs' at the time of receiving the remote job results. Refer to the Linkexpress manual for information on how to send job results notification to Information Integrator from the remote system.

      Note that the timing for restoring 'Execute remote jobs' should be specified in the jobrsp operand within the extension operand specification file.
    - Users with the required privileges for job execution under Windows specifications must belong to the Administrators group. If the user does not belong to the Administrators group, a TR10325, TR20325 or TR30325 message will be output and the job may fail to start. Furthermore, the Administrator group must be set to the Domain Admin group if a domain has been created. Note that the job will only operate according to the specified user privileges at the time of job execution. However, the started job will operate on the systems administrator privileges if started on Windows Vista or Windows Server 2008.
      Note also that the remote user name should be specified in the prtsecur operand within the extension operand specification file.
    - Environment variables are carried over by the systems environment variables. If a job requires its own unique environment variables, such as 'path' environment variables, change the environment variables within the job.
- When the remote system is UNIX
    - 'Execute remote jobs' cannot be used when the remote job fits any of the following descriptions:
        - Commands required by the control terminal (such as passwd)
        - Full screen-type commands (such as vi)
    - Select one of the following options to use as the time to restore the 'Execute remote jobs' command:
        - When job results are received for the remote job
        - When the remote job is started

      Ensure that notification of job results is sent to Information Integrator from the remote system when restoring 'Execute remote jobs' at the time of receiving the remote job results. Refer to the Linkexpress manual for information on how to send job results notification to Information Integrator from the remote system.

Note that the timing for restoring 'Execute remote jobs' should be specified in the jobrsp operand within the extension operand specification file.

- The environment that started up the Information Integrator Server (such as an RC script, console or terminal) carries over any environment variables, umask values, standard output and/or standard error output. The user's shell environment (such as .profile or .chsrc) specified in 'Execute remote jobs' is not carried over.
  Set any environment variables required to execute a job, and/or any umask values required to access files from the job, in the execution environment of the RC shell, console or terminal that starts up the Information Integrator Server, or alternatively, set the aforementioned environment variables within the job.
  Standard output and/or standard error output from the job is output to the console or terminal that started up the Information Integrator Server. If the terminal is exited after starting up the Information Integrator Server, the output may return an EIO error. Accordingly, please consider redirecting any standard output and/or standard error output from the job to a temporary file.
  Note that if the console or terminal locks or hangs, it will not be possible to complete the job because the standards output and/or standard error processing will be in a waiting state.

- Any jobs started on UNIX will be started in accordance with the privileges of the specified user at the time of job startup. However, no support is provided for secondary user groups.

- When the remote system is Global Server or PRIMEFORCE

  'Execute remote jobs' will be restored when the remote job is started. The timing for this command to be restored should be specified in the jobrsp operand within the extension operand specification file. Ensure that 'no' is specified to make the command restore when the job is started.

1.5.2 Notify Job Results

'Notify job results' refers to the process of notification of job execution results being sent by the remote job (*1) to the Information Integrator Server. Note that FTP+ or HTTP or HTTPS must be used when 'Notify job results' is performed using 'Execute remote jobs'.

When performing 'Execute remote jobs' for any products other than Linkexpress integrated products, settings must be specified to ensure that 'Notify job results' is not executed. For example, set to 'Do not notify job results' when starting a job on Global Server or PRIMEFORCE.

*1: When requesting 'Execute remote jobs', job results will be notified when the 'Notify job results' value is omitted.

Specify whether or not to notify job results in the extension operand specification file of the ifiexejob command.

'Notify job results'

Set the data source definition as described below when using 'Notify job results'.

- Set 'FTP+' or 'HTTP' or 'HTTPS' in 'Protocol'.
- One multiple each is required for 'Concurrency-initiator' and 'Concurrency-responder' on systems that request job execution.
- One multiple each is required for 'Concurrency-initiator' and 'Concurrency-responder' on systems executing jobs.

'Notify job results' must be issued from the started jobs when set to 'Notify job results'. 'Notify job results' prepares the command and the user program interface.

In Information Integrator: the ifirspjob command

In Linkexpress: the lxrspjob command

Specify the job results waiting time to account for times when, for some reason, it may not possible for 'Notify job results' to send notifications to the Information Integration Server, such as when 'Notify job results' has not been issued. The job results waiting time can be specified when requesting the startup system. If 'Notify job results' has not been issued and the job results waiting time has also not been specified, timer monitoring will not be performed, resulting in the job start request entering an unresponsive state. If no results are notified by 'Notify job results' within 30 minutes of the job being started, a TRx3014 message will be output. No message will be output if the job results waiting time has been specified.

'Do not notify job results'

Set the data source definition as described below when not using 'Notify job results'.

- Specify 'FTP', 'FTP+', 'HTTP', 'HTTPS', or 'HICS' as the protocol.
- Systems requesting jobs to be started require one multiple of 'Concurrency-initiator'.

- Systems executing jobs require one multiple of 'Concurrency-responder'.

Information

Specify 'Do not notify job results' when starting a job using 'Execute remote jobs' on Global Server or PRIMEFORCE.

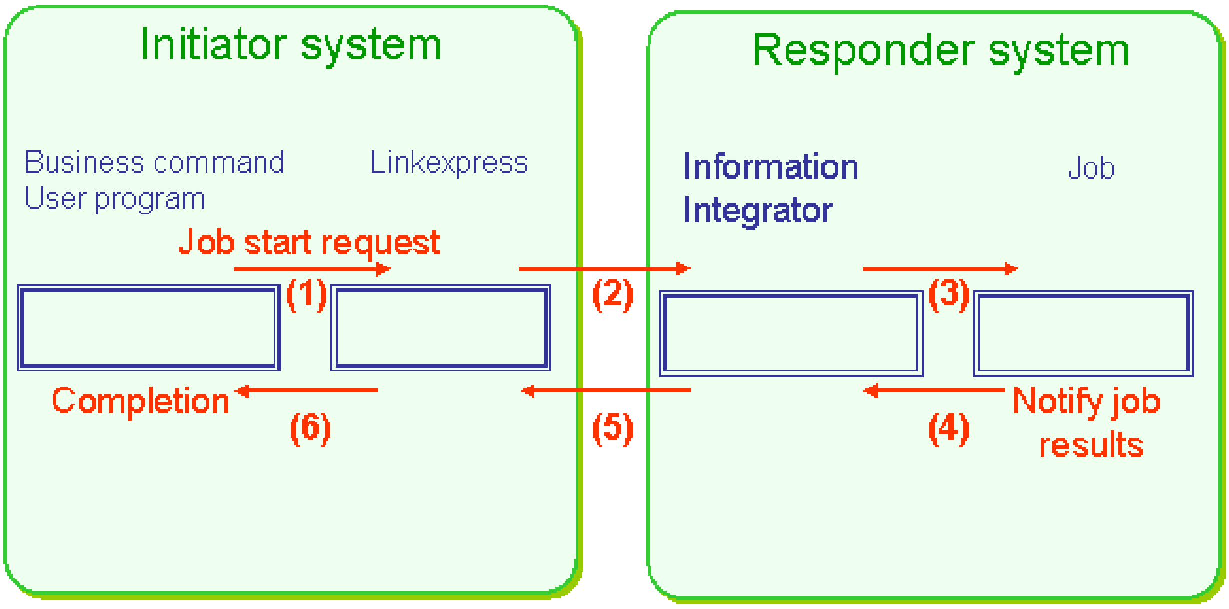
1.5.3 Startup Interface for Jobs

Notification of execution results for jobs started from the remote system will be sent for some jobs and not sent for others.

- Jobs for which notification of execution results is sent

  Jobs that perform 'Notify job results' send notification of results to Information Integrator when execution is complete. The call sequence for jobs is illustrated below.

Figure 1.19 Process Flow of jobs that send notification of execution results



1. A job startup request is sent to Linkexpress from the user program, file transfer command or business.
2. A job execution request is sent from Linkexpress at the initiator to Information Integrator at the responder system.
3. Information Integrator at the responder system starts the job.
4. The job sends notification of the job results to Information Integrator at the responder system. (*1)
5. Information Integrator at the responder system sends notification of the job execution results to Linkexpress at the initiator.
6. Linkexpress sends notification of the job execution results to the user program, file transfer command or business. (*2)

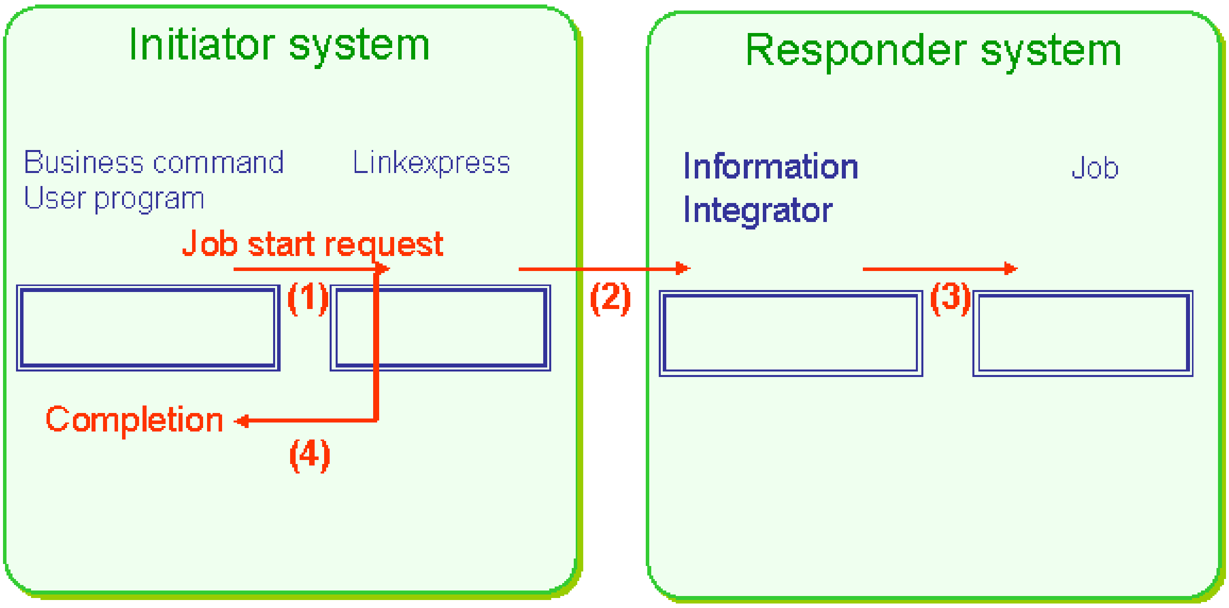
*1: The notification indicated in point 6 will not happen if the job results notification indicated at Point 4 does not occur, or if the process ends in an error.

*2: Timer monitoring can be specified for points 1 through 6 using the jobtime operand in the user program, file transfer command, event definition or network definition.

- Jobs for which 'Notification of execution results' is not sent

  Jobs for which 'Notification of job results' is not sent will recognize the completion of job start to be the execution results. The call sequence for jobs is illustrated below. Note that these types of jobs can also be called from Linkexpress File Transfer, DTS or HICS.

  Figure 1.20 Process Flow of jobs that do not send notification of execution results



1. A job startup request is sent to Linkexpress from the user program, file transfer command or business.
2. A job execution request is sent from Linkexpress at the initiator to Information Integrator at the responder system.
3. Information Integrator at the responder system starts the job.
4. After the job indicated in point 3 is started, Linkexpress sends notification to the user program, file transfer command or business that the job start is complete.

1.5.4 Operating Environment for 'Execute Remote Jobs'

The operating environment for 'Execute remote jobs' is as described below.

ID environment

Jobs started by Information Integrator operate according to the privileges of the specified user at the time the remote job is started.

If 'User authentication-logon' in the data source definition is set to 'NO', the environment will operate on the privileges of the system that started Information Integrator.

Note that the environment will operate according to system administrator privileges when starting jobs on Windows Server 2008, regardless of the aforementioned specification.

Environment variables

When starting jobs on UNIX, the environment variables of jobs started by Information Integrator will be carried over by the environment variables at the time Information Integrator was started up. When starting jobs on Windows, the system environment variables will be carried over.
If a job requires its own unique environment variables, such as 'path' environment variables, set the environment variables within the job.

When Information Integrator is started automatically, the environment variables will be as described below.

- Windows: The 'SYSTEM' environment variables will be carried over.
- UNIX: The environment variables set in Information Integrator's start shell will be carried over.

Cautionary notes about operating environments

The following notes apply to jobs started on UNIX:
Environment variables, umask values, standard output and/or standard error output will carry over the environment that started up Information Integrator (such as an RC script, console or terminal). The shell environment (such as .profile or .chsrc) of the user specified in 'Job execution' is not carried over.
Set any environment variables required to execute a job, and/or any umask values required to access files from the job, in the execution environment of the RC shell, console or terminal that starts up Information Integrator, or alternatively, set the aforementioned environment variables within the job.
Standard output and/or standard error output from the job is output to the console or terminal that started up Information Integrator. If the terminal is exited after starting up Information Integrator, the output may return an EIO error. Accordingly, please consider redirecting any standard output and/or standard error output from the job to a temporary file.
Note that if the console or terminal locks or hangs, it will not be possible to complete the job because the standards output and/or standard error processing will be in a waiting state.

1.5.5 Job Start Parameters

Job start parameters started by Information Integrator will be notified in the format described below. Of the parameters for start, the user-specified job parameters will be common except when the initiator server is Information Integrator or Linkexpress.

Job start format

The job start format will be as described below when performing 'Notify job results'. Parameters will be notified in the format described below when 'Execute remote jobs' is specified by 'Notify job results' from the initiator.

```
jobName -r remoteSystemName -i jobIDNumber [user-specified job parameters]
```

The job start format will be as described below when performing 'Execute remote jobs' with 'Do not notify job results'.

```
jobName [user-specified job parameters]
```

remoteSystemName:

Parameters used by Information Integrator. Notification is sent to jobs started by Information Integrator. The parameter type used is '-r'.

jobIDNumber:

Parameters used by Information Integrator. Notification is sent to jobs started by Information Integrator. The parameter type used is '-i'.

user-specified job parameters:

Parameters used by the user as required. User-specific parameters specified by the initiator are appended behind the aforementioned parameters.

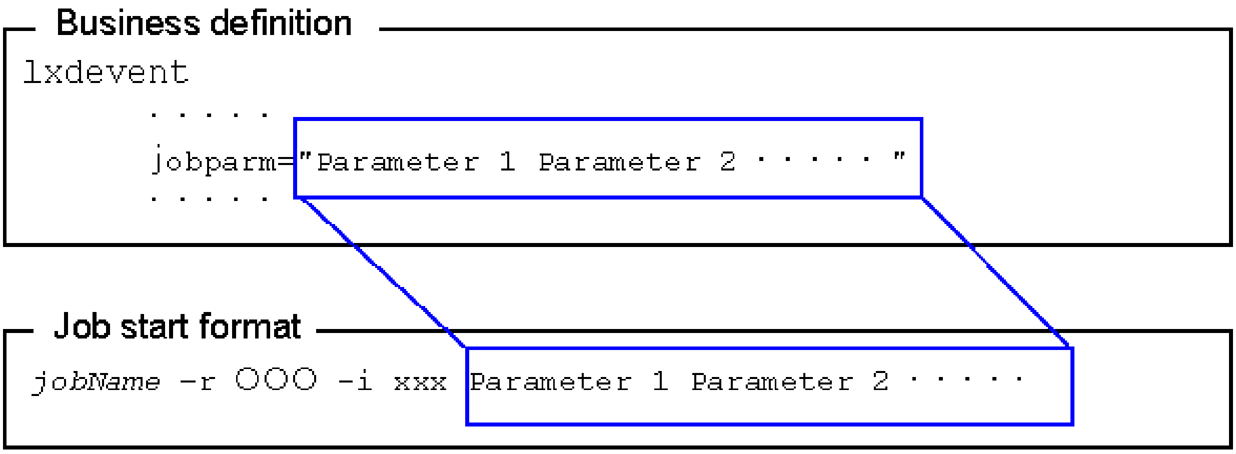
Notes on user-specified job parameters

The rules listed below apply to user-specified job parameters. These rules are common when performing 'Execute remote jobs' with servers other than Information Integrator or Linkexpress.

- Notification of user-specified job parameters cannot be sent when the remote system for 'Execute remote jobs' is HICS
- If spaces are used in the user-specified job parameters, the job will be recognized as containing multiple parameters.
- Use alphanumeric characters in user-specified job parameters.

When the initiator is Linkexpress

Enter user-specified job parameters in the business definition using the jobparm operand. The jobparam operand is also used when executing the file transfer command.



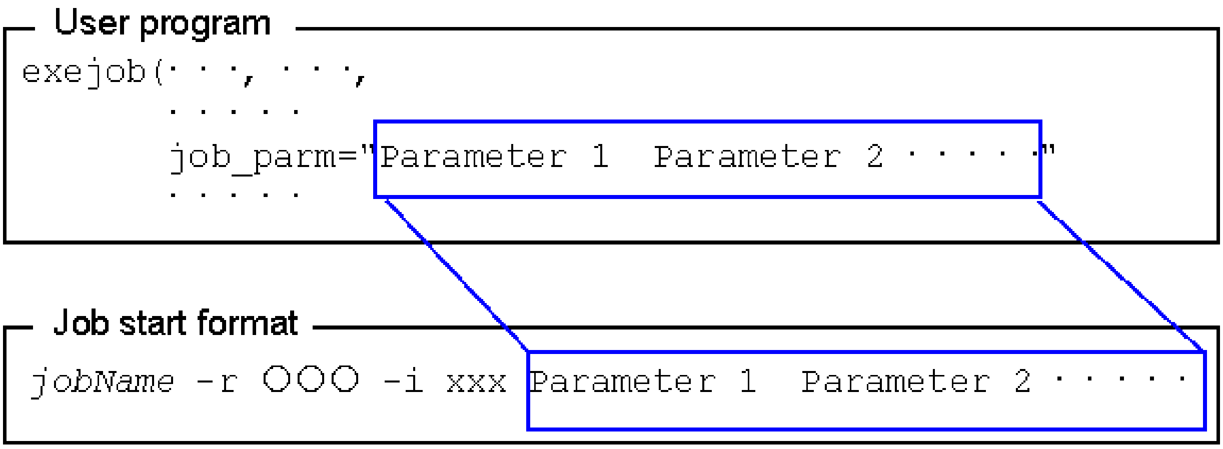
Business definition

```
lxdevent
        . . . . . .
        jobparm="Parameter 1 Parameter 2 · · · · · · "
        . . . . . .
```

Job start format

```
jobName -r OOO -i xxx Parameter 1 Parameter 2 · · · · · ·
```

Enter the user-specified job parameters for the user program in job_parm of EXEJOB or in JobParameter of ExecuteJob.



User program

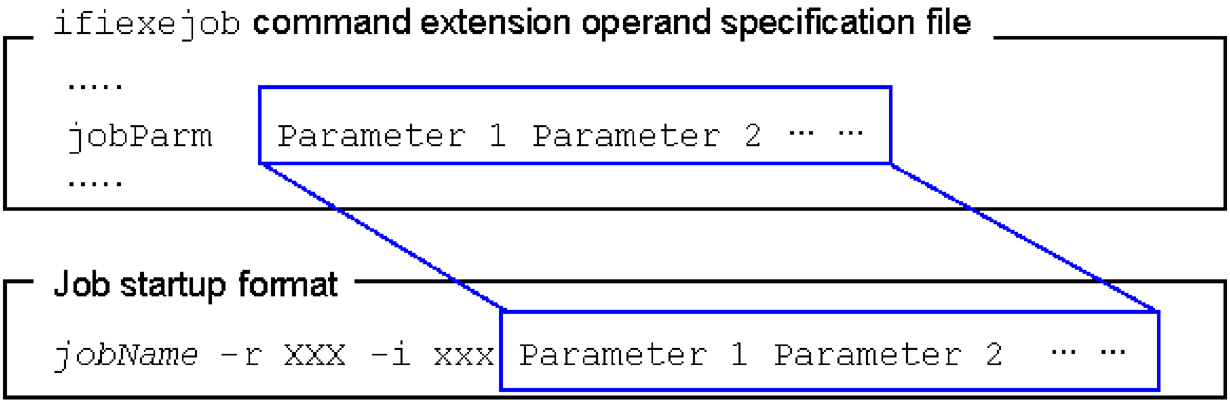
```
exejob(· · ·, · · ·,
        . . . . . .
        job_parm="Parameter 1  Parameter 2 · · · · ·"
        . . . . . .
```

Job start format

```
jobName -r OOO -i xxx Parameter 1  Parameter 2 · · · · · ·
```

When the initiator is Information Integrator

Enter the user-specified job parameters in jobparm of the ifiexejob command's extension operand specification file.



ifiexejob command extension operand specification file

```
.....
jobParm    Parameter 1 Parameter 2 … …
.....
```

Job startup format

```
jobName -r XXX -i xxx Parameter 1 Parameter 2  … …
```

For further information on how to enter user-specified job parameters from non-Linkexpress integrated products, such as Linkexpress File Transfer or DTS, refer to the manual of the relevant integrated product.

Note

It is not possible to specify job execution parameters when the initiator is Global Server/PRIMEFORCE's Linkexpress File transfer or DTS.

1.5.6 Notification of Information from Job to Information Integrator

Jobs started from Information Integrator with 'Notify job results' must notify Information of the job results using the output interface described below. Note that there is no requirement for remote jobs started with 'Do not notify job results' to send notification to Information Integrator.

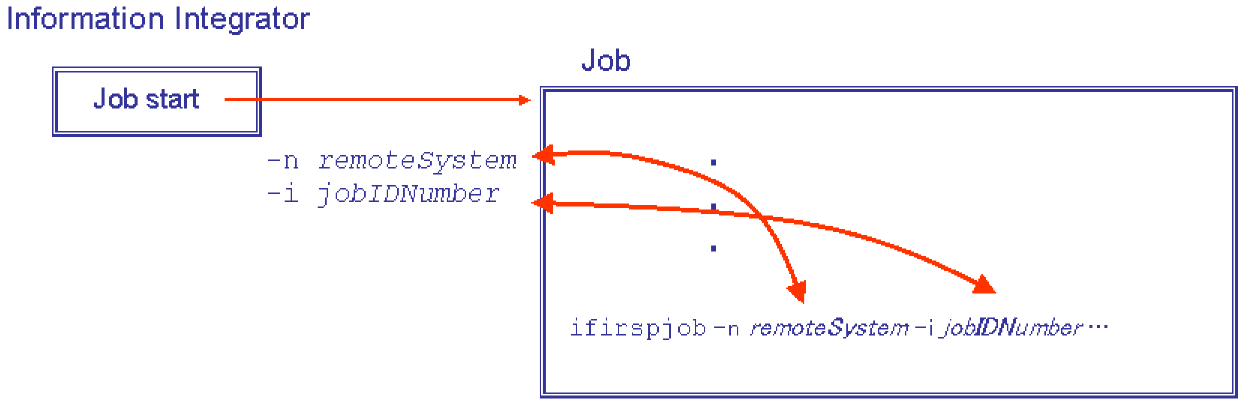
Notify job results

'Notify job results' is an interface for notifying the results of jobs started by Information Integrator. Specification of 'Notify job results' enables Information Integrator to recognize when a job is complete. Either 'FTP+' or 'HTTP' or 'HTTPS' must be specified in the data source definition 'Protocol' for results of started jobs to be notified.

- Linkexpress: lxrspjob command
- Information Integrator: ifirspjob command

The settings value requested in 'Notify job results' contains the remote system name and job ID number. These settings values are notified in the input interface when a job is started.

Figure 1.21 Settings values requested by 'Notify job results'



remoteSystem: Notification is sent from Information Integrator as the -n operand.

jobIDNumber: Notification is sent from Information Integrator as the -i operand.

1.5.7 Job Samples

Please refer to the information below on job samples. When using a sample, copy it into an arbitrary directory and then edit.

- Windows

```
InformationIntegratorServerInstallDirectory\sample\RSPJOB.BAT
```

- UNIX

```
/opt/FJSVifisv/sample/rspjob.sh
```

When using the above samples, remove the comment hashtag (#) character from the PATH environment row. Also, add a path if required.

Point

The remote system name and job ID number can each be referred to by the second parameter ($2) and the fourth parameter ($4), respectively.

1.6 Process Control Features

1.6.1 Process Flow Execution

Information Integrator provides the following three execution methods:

- The user or application executes the process.
- Execute processes periodically using the schedule feature.
- Execute processes when an event occurs.

  This refers to event monitor execution for processes executed as the result of some event occurring, such as when data arrives from the collection source. As event occurrences are automatically detected and the Process Flow can then be executed, this feature can be integrated with business systems that perform serial data processing.

Each of the following specifications is possible:

Table 1.9 Types of schedule features

Category	Type	Description
Schedule execution	Daily	Executed daily at a certain time (hours:minutes)
	Weekly	Executed weekly on a certain day and time (hours:minutes)
	Monthly	Executed monthly on a certain date and time (hours:minutes)
	Fixed time interval repeat	Executed repeatedly at the specified interval 1 to 999 (minutes)
Event monitor execution	Event monitor execution	Automatically executes the Process Flow when an event occurs, such as data arriving from the collection source. Refer to "1.6.5 Event Monitor" for more information on this topic.

Point

The schedule execution time span can be set in advance for fixed time interval repeat and collection monitor execution. For example, it is possible to perform settings in Information Integrator so that processes executed repeatedly at 60 minute intervals and processes executed for event monitoring are each executed during a time span that begins at 9:00 AM and concludes at 5:00 PM.

Furthermore, it is possible to pause or restart a schedule following the start of process operations for all schedule features, including daily and weekly.

1.6.2 Automatic Retry

File transfers between systems can end in error for a variety of reasons. Transfer errors caused by temporary communication errors on the network can be resolved by resending files. However, it can be time-consuming for users to investigate causes or to perform resend operations.

The automatic retry feature automatically determines the cause of the transfer error and can automatically resend files without any special operation in the case of temporary communication errors on the network. This helps to alleviate the burden on the user and shorten the time required to resend files.

Automatic retry retries each requested file transfer in accordance with the specified retry interval and number of retry attempts. If an error occurs with the file transfer and the process abnormally ends, despite attempting to retry for the specified number of times, notification of the final error information will be sent to the user.

1.6.3 Exclusive Control

When it is not possible to execute multiple processes due to the relationship between each execution process and the resources used by applications for user plug-ins (for example, to prevent duplicate extraction of identical data when performing 'Extract from DB' or when the specifications of a product to be used as a plug-in do not permit duplicate execution) the process can be made exclusive within Information Integrator.

This alleviates the risk of data being duplicated by simultaneous execution or excessive errors (excessive concurrency) occurring.

The unit of control varies according to the process. The table below indicates the relationship between processes and the unit of control.

Table 1.10 The relationship between process types and exclusive control

Process types	Feature	Unit of control	Assumed risk	Remarks
Data Collection	File transfer	Remote system name + DataBox	Data duplication	Invalid when operation type is the responder
	Extract from DB	DBMS type + database name + schema name + table name	Data duplication Data lock	Deliver data also exclusive
	BLOB collection	End point URL + container name + BLOB name	Data duplication	Deliver data also exclusive
Data delivery	File transfer	Remote system name + DataBox	Data duplication	Invalid when operation type is the responder
	Store in DB	DBMS type + database name + schema name + table name	Data duplication Data lock	Data collection is also exclusive
	BLOB delivery	End point URL + container name + BLOB name	Data duplication	Data collection is also exclusive
Data conversion	Conversion	Data conversion ID	CPU monopolized by the same conversion process	Control as substitute for concurrency restrictions
Filtering	Filtering	Filtering ID	CPU monopolized by the same filtering process	Control as substitute for concurrency restrictions
Plug-in	Plug-in	Plug-in ID	Data duplication	

There is no exclusive control feature when using the extract object feature at data collection and also using the store object feature at deliver data.

1.6.4 Sequence Control

It is possible to prevent multiple processing of the same process and/or alleviate the system load when newly executing an identical Process Flow during execution of an existing Process Flow.

Perform sequence control to allow the first Process Flow to complete processing before starting the next Process Flow.

1.6.5 Event Monitor

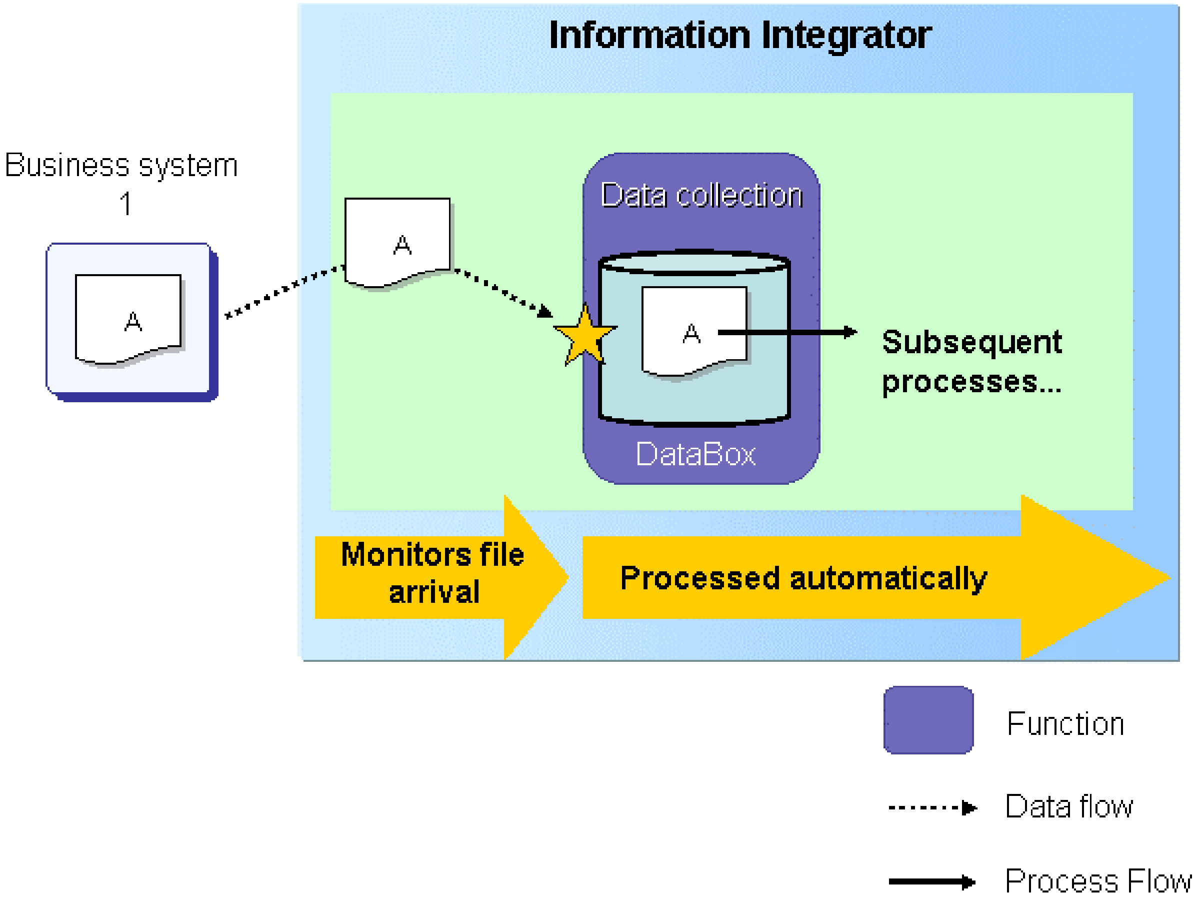
'Event monitor' refers to the automatic execution of processes as the result of some event occurring, such as when data arrives from the collection source.

'Event monitor' will automatically execute a Process Flow when one of the following events occurs:

- When receiving files as the responder

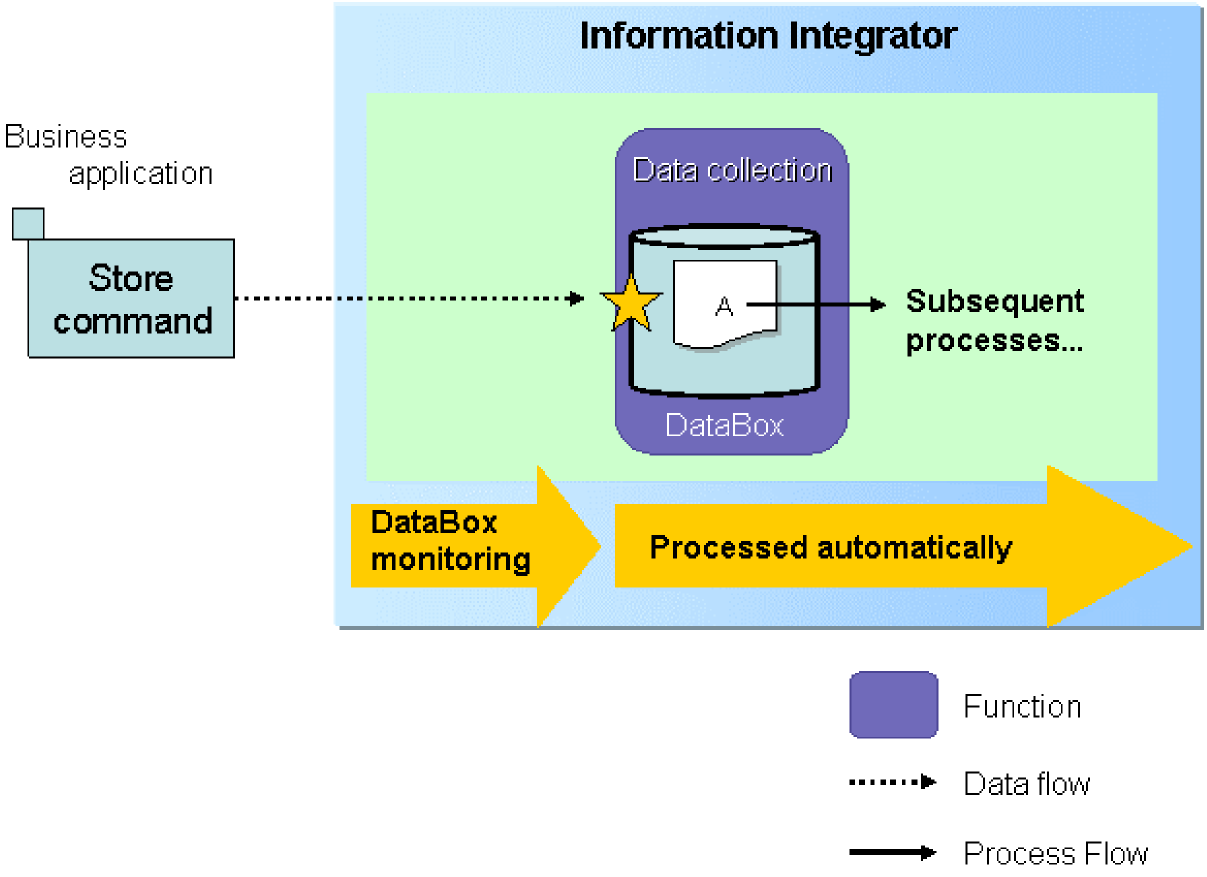
  When receiving files as the responder, the Process Flow will execute automatically when file transfer completes.

  This occurs regardless of the transfer protocol used (such as FTP+ or HULFT).



- When storing data in the DataBox

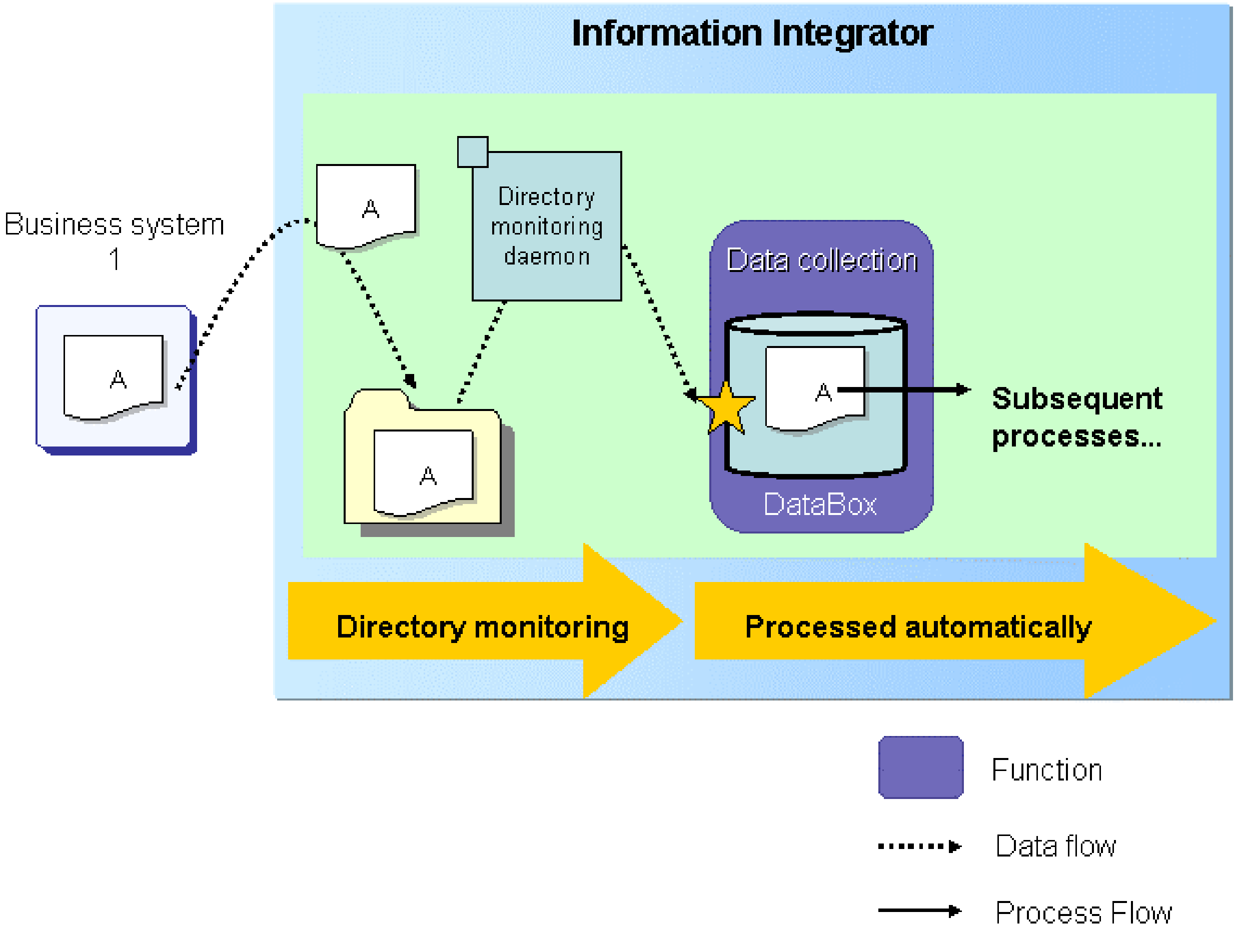
  When using the Information Integrator operation command (the ifirstor command), the Process Flow will execute automatically when data is stored in the DataBox.



- When generating a file in the monitored directory

  Specify the directory to be monitored and then use the OS command or business application to generate files to be executed in the Process Flow in that directory. The Process Flow will execute automatically when the files are generated in the directory to be monitored.

Specify the directory to be monitored in the operating environment file. Refer to the *Setup Guide* for more information.



Refer to "1.6.5.1 When Monitoring a Directory" for more information on operations under these conditions.

1.6.5.1 When Monitoring a Directory

This feature determines whether updating of files for execution in the Process Flow has been completed by another application. When the updating is complete, the Process Flow will execute automatically.

Some time will be required from start of write operations until completion when a large volume of data is involved. Accordingly, if the date of updating is prior to the current time, and there is no exclusivity on the file, the write operations will be deemed to be complete.

Files for processing that have been generated in the monitored directory will be automatically deleted when the Process Flow is executed.

Usage is described below.

Note

In general, only those files to be executed in the Process Flow should be generated in the monitored directory.

Of those files that are generated in the monitored directory, files to be executed in the Process Flow at the start time of Process Flow execution will be searched. This is because it is time-consuming to perform this search.

Note that although it is possible to determine whether file update is complete by using the aforementioned procedure, files should only be generated using one of the methods described below. In particular, one of the following methods must be used when generating files containing large volumes of data in the monitored directory:

- Create files once in a separate directory and then place them in the monitored directory.
- Make the directory exclusive whilst writing the files.

Specifying files for Process Flow execution

Of those files that were generated in the monitored directory, specify in advance the names of the files for Process Flow execution in the collection file names of the data collection definition.

- Perfect match

  When processing only those files whose file name matches perfectly.

- Wild card specification

  When processing files whose name matches partially, such as a prefix match or suffix match.

Refer to *III Studio Help* for more information.

Handling files exempt from Process Flow execution

It is possible to specify in advance how files not targeted for execution in the Process Flow will be handled, from those files that were generated in the monitored directory.

Non-targeted files will be handled as described below.

- Non-targeted files are automatically deleted.
- Non-targeted files are backed up to a separate directory.

These are specified in the operating environment file. Refer to the *Setup Guide* for more information.

Note

Avoid continued placement of files not required for Process Flow execution in the monitored directory.

Point

When backing up non-targeted files to a separate directory

After being backed up, the new file name will comprise the source file name (the file name when initially generated in the monitored directory) together with the date and a serial number. This differentiates files when there are multiple instances of the same file name.

1.6.5.2 When Monitoring Multiple Events

The Process Flow can be executed automatically when monitoring multiple events if one of the following occurs:

- Files arrive for all events

  Wait until all files to be executed in the Process Flow arrive.

- One of the files arrives

  The Process Flow is executed automatically when monitoring multiple events and the files arrive for one of those events. There is no need to wait for all files.

Note that it is also possible to use 'Action when not found' in the data collection definition to customize the point at which execution takes place when monitoring multiple events. Refer to "3.6.3 Designing the Collection Method" for more information.

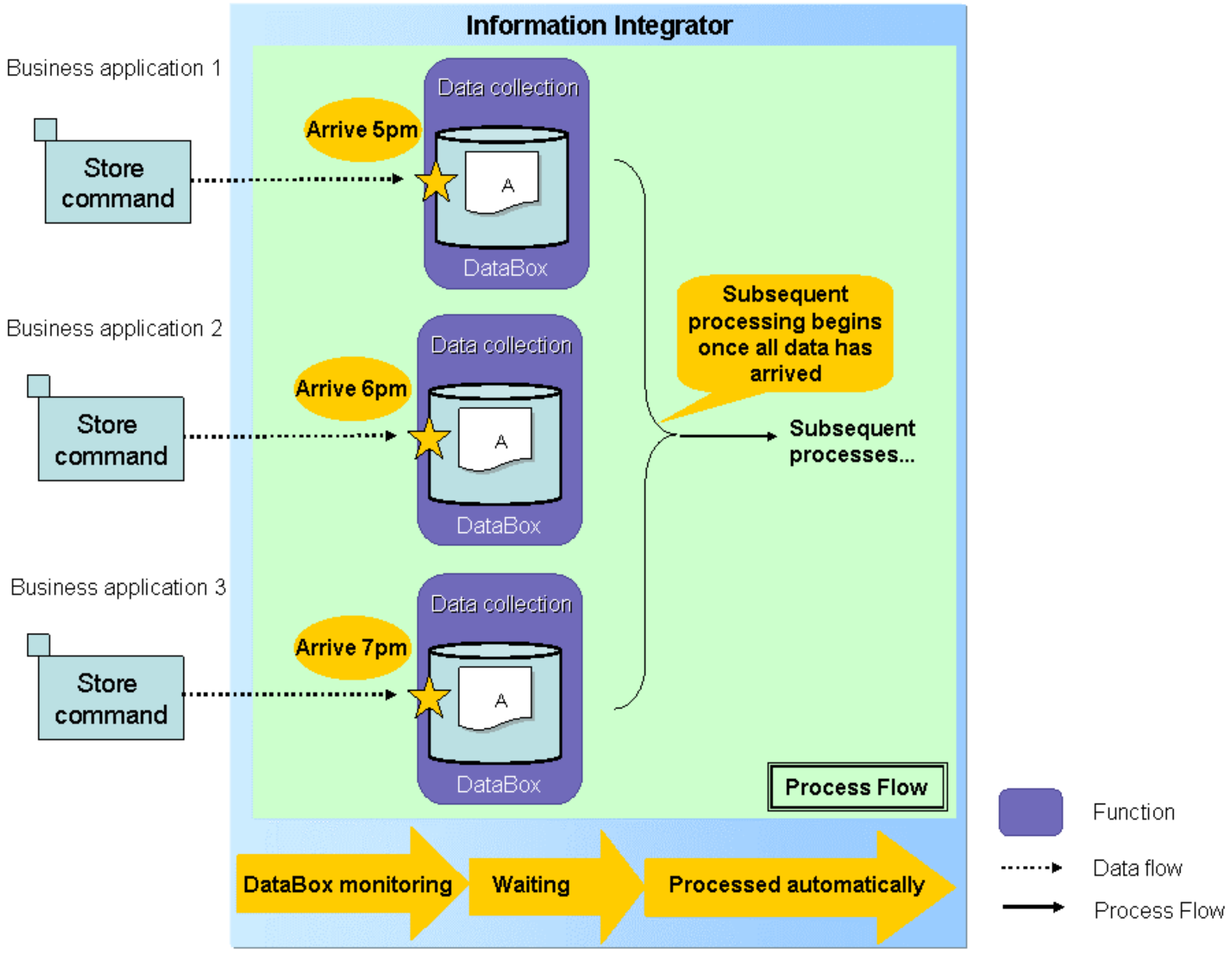
Some examples are illustrated below.

Point

While the example below is an event form that stores all events in the DataBox, events can become mixed when receiving files as the responder or generating files in the monitored directory.

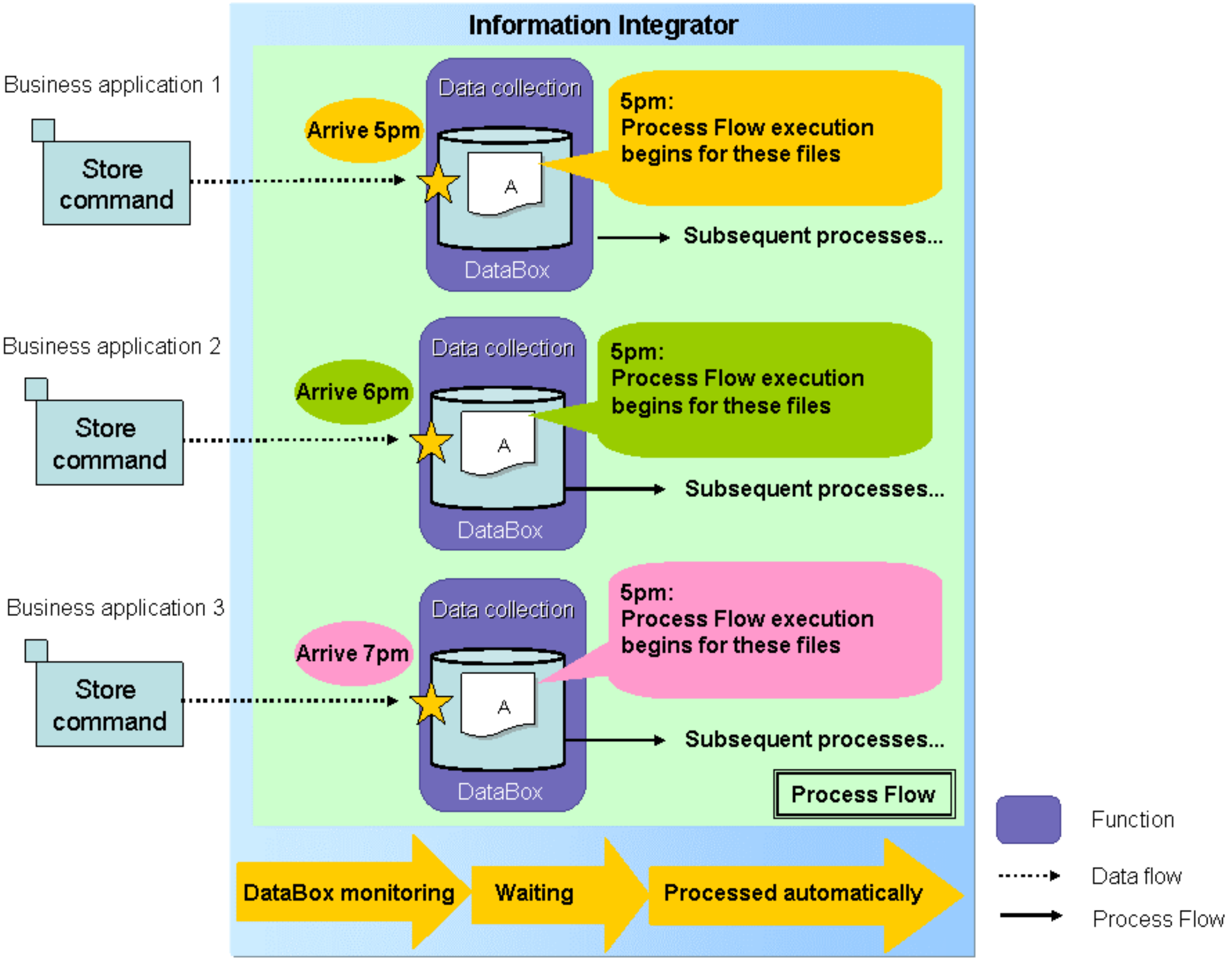
- Example: Waiting until files arrive for all events.

  In the example below, Information Integrator waits for all files transferred from business application 1 through 3 to arrive.



- Example: The Process Flow executes automatically when one of the files arrives.

  When data is stored in the DataBox once daily at a fixed time, the Process Flow will execute for the files stored in the DataBox three times, at 17:00, 18:00, and 19:00.



1.6.5.3 Pausing or Setting a Pause Period for Event Monitoring

It is possible to set a pause period for event monitoring ahead of time, in case of any events that may occur outside of business hours. Similarly, it is also possible to pause event monitoring in case of any events that may occur during recovery operations after switching cluster operations.

- Setting a pause period for event monitoring

  When daily business hours are clearly defined, specify the event monitor time span ahead of time. Specify the time span in 'sch_workingtime' in the operating environment file. Refer to the *Setup Guide* for more information.

- Pausing 'Event monitor'

  Pause the Process Flow schedule to temporarily stop 'Event monitor'.

  To commence again, resume the Process Flow schedule.

  Schedule pauses and resumes are performed using the 'Operation Management Client' or the operations command. Refer to the *Operations Guide* for more information.

Note that processing will continue without change even if a pause time is entered into during process execution or the execution of the Process Flow is paused.

Additionally, if an event occurs during a pause time or paused process, the Process Flow will not execute automatically. Execute the Process Flow after entering into the 'Event monitor' time span or after restarting the schedule. Use the 'Operation Management Client' or ifiexeprc command to execute the Process Flow.

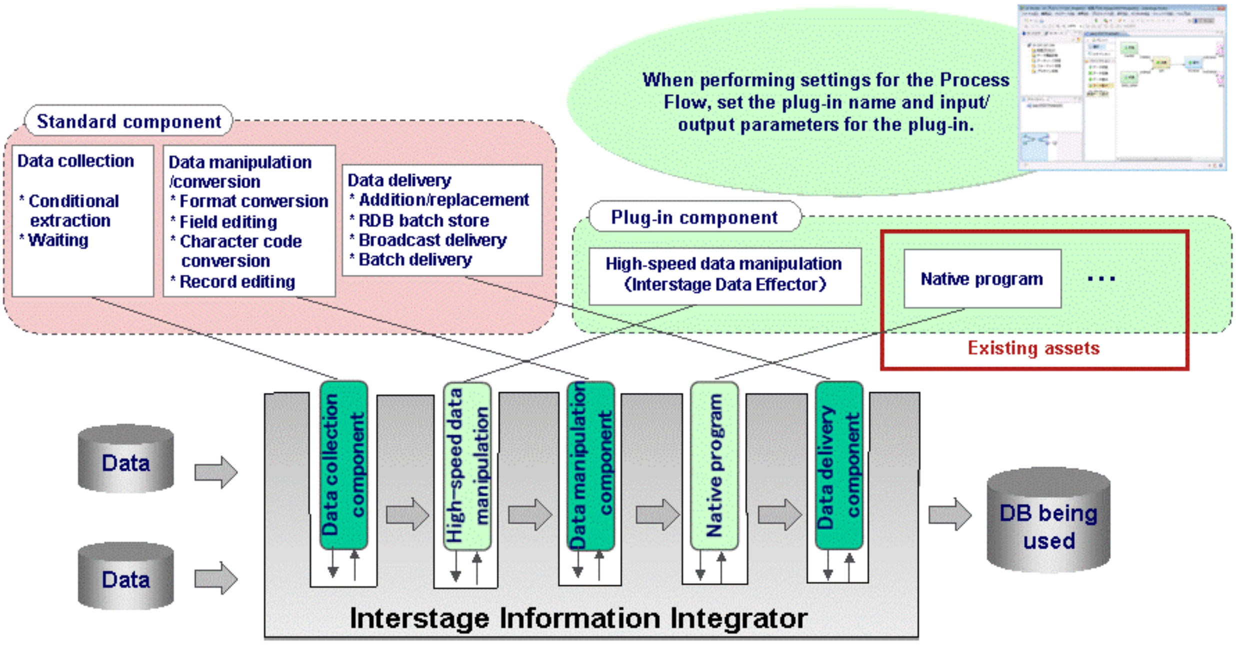
1.7 Plug-ins

Conversion applications that were in use on existing systems and various kinds of extension tools can be embedded in Information Integrator using a plug-in feature.

The plug-in features comprise one feature (hereafter known as the process plug-in feature) that enables plug-ins to be embedded for various processes such as collection, conversion and delivery, and another feature (hereafter known as the conversion plug-in feature) that embeds, as plug-ins, any conversion applications required at the time of conversion processing by item.

Each of the plug-in features is described below.

1.7.1 Process Plug-in Illustration


When performing settings for the Process Flow, set the plug-in name and input/ output parameters for the plug-in.
Standard component
Data collection
* Conditional extraction
* Waiting
Data manipulation /conversion
* Format conversion
* Field editing
* Character code conversion
* Record editing
Data delivery
* Addition/replacement
* RDB batch store
* Broadcast delivery
* Batch delivery
Plug-in component
High-speed data manipulation (Interstage Data Effector)
Native program
...
Existing assets
Data
Data
Data collection component
High-speed data manipulation
Data manipulation component
Native program
Data delivery component
DB being used
Interstage Information Integrator

The applications that may be embedded are listed below.

Table 1.11 Applications that can be used by plug-ins

Platform	Applications that can be used
Windows	exe/bat format (executable formats). Binary programs and scripts (batch file format) that can be executed from the command prompt.
UNIX	Binary programs and various scripts that can be executed from the console.

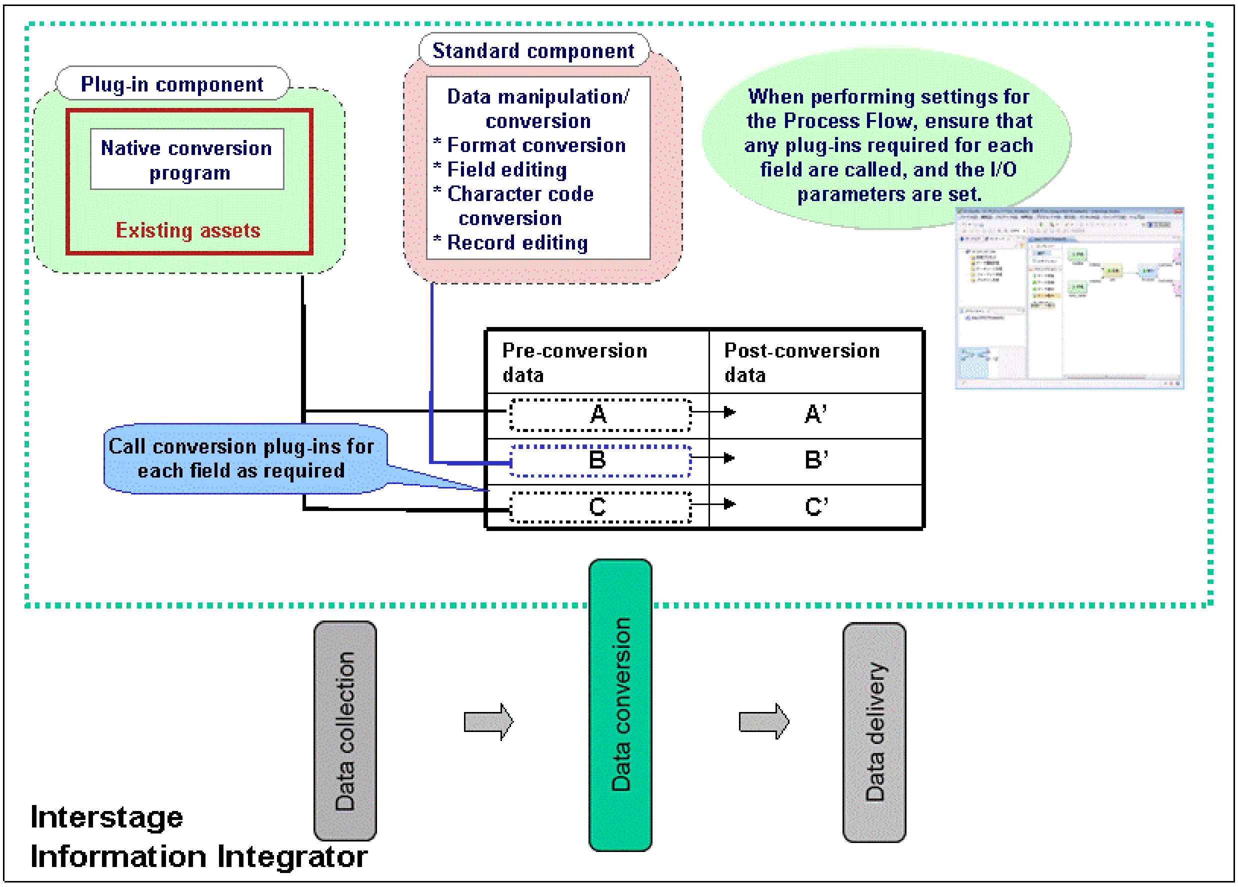
Note

The following programs cannot be executed:

- Windows
    - Programs (such as notepad) that use windows or commands
    - Programs or commands that do not operate normally when the AT command is executed
    - Programs or commands that do not operate normally when the schtasks command is executed
    - Shell scripts created with Windows PowerShell

- UNIX
    - Commands required by the control terminal (such as passwd)
    - Full screen-type commands (such as vi)

1.7.2 Conversion Plug-in Illustration


Plug-in component
Native conversion program
Existing assets
Standard component
Data manipulation/ conversion
* Format conversion
* Field editing
* Character code conversion
* Record editing
When performing settings for the Process Flow, ensure that any plug-ins required for each field are called, and the I/O parameters are set.
Pre-conversion data
Post-conversion data
A
A'
B
B'
C
C'
Call conversion plug-ins for each field as required
Data collection
Data conversion
Data delivery
Interstage Information Integrator

The applications that can be embedded are listed below.

Applications that can be used by plug-ins

Applications that can be used are subroutines (such as functions or APIs) created in C language or COBOL language and provided in a format included in a mappable dynamic library within the Information Integrator address space.

Note

Data formats when handling data for date attributes and date time attributes

Following execution of the conversion plug-in, the format will be as indicated below.

- Date attributes: 'YYYY-MM-DD'
- Date and time attributes: 'YYYY-MM-DD hh:mm:ss'

Programs that cannot be executed

The following programs cannot be executed:

- It is not possible to return processing results for multiple items when calling an application a single time. Processing results can only be returned for a single item per single call of an application.
  However, it is possible to send notification to the application of multi-item data per single call of that application.

- It is not possible to use applications frequently used in object-oriented language that require execution procedures to be performed in advance, such as getter/setter. Specify an application that will be processed completely solely by performing individual calls.

- It is not possible to specify functions belonging to the standard library in the conversion plug-in.

Data formats that can be integrated with the conversion plug-in

Data in CSV file format or binary format can be integrated with the conversion plug-in.

1.8 Management Features

1.8.1 Definition Creation and Definition Management

In Information Integrator, Process Flow design information is used without change as system definition information in the execution environment.

Definition Creation

Definitions are created for design information for the various types of parameters required for Process Flow processing procedures, and to execute the Process Flow.

These procedures are performed using III Studio.

- III Studio

  'III Studio' is the name given to the tool that designs whilst visually confirming each definition.

  III Studio is a plug-in that operates on the Interstage Studio GUI-based integrated development environment (IDE). III Studio is characterized by the unified operations of its IDE.

Point

The 'Metadata collection feature' enhances the efficiency of definition creation

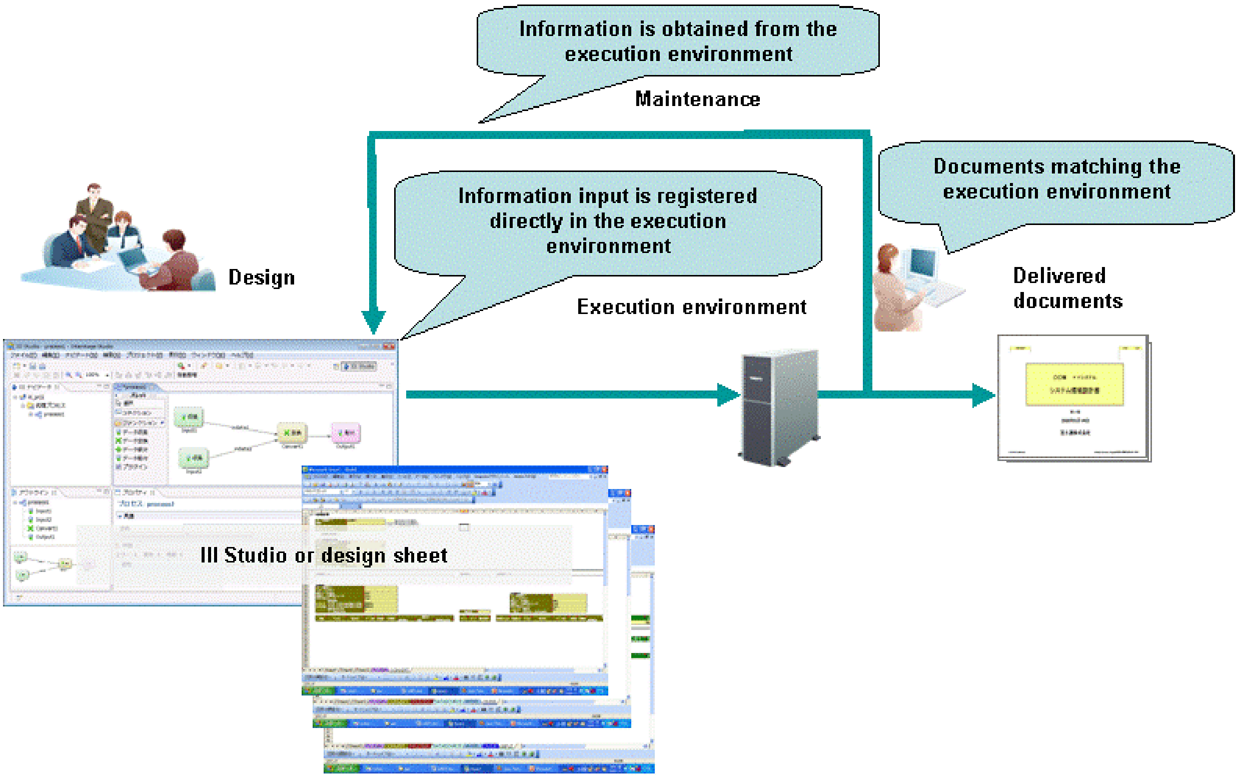
Metadata refers to information on data formats or processing methods in collection sources or delivery destinations. In Information Integrator, it is possible to take metadata relating to data structures (such as item names or data attributes) and incorporate it into the design.

This feature eliminates the need to create metadata from the beginning when inputting information to III Studio, thus greatly improving the efficiency of definition creation.

Refer to "1.8.1.1 Metadata Collection Feature" for more information.

Information designed in III Studio is registered in the execution environment prior to use. A series of system development phases from design through to implementation use III Studio as the starting point.

Figure 1.22 Illustration of definition creation in Information Integrator


Information is obtained from the execution environment
Maintenance
Design
Information input is registered directly in the execution environment
Execution environment
Documents matching the execution environment
Delivered documents
III Studio or design sheet

Definition management

The series of definition information, which can be referenced or from which definitions registered in the execution environment can be retrieved, is managed in III Studio.

Definitions can be referenced or reviewed using one of the following:

- III Studio

1.8.1.1 Metadata Collection Feature

This feature imports metadata from the collection source or delivery destination into Information Integrator.

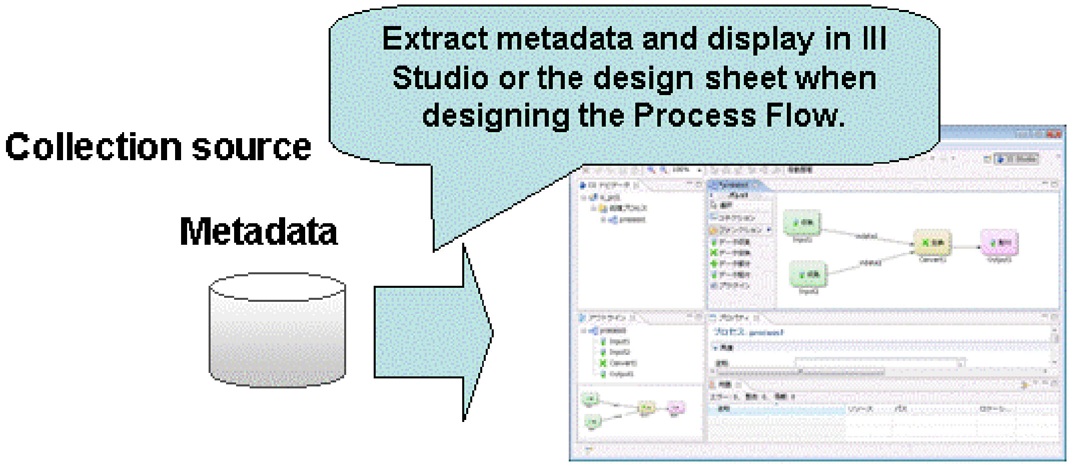
If data conversion or editing is being performed in Information Integrator, information relating to the structure of the data, both pre- and post-conversion, should be input in advance. When performing this definition, this feature will import metadata from the data being converted.

It is also possible to collect metadata relating to the data structures of the following types of data:

- CSV files
- COPY LIB files

  Files specifying definitions (COBOL COPY phrases) relating to COBOL data items.
- Symfoware Server
- Oracle
- SQL Server
- SQL Azure
- Salesforce objects

Metadata that may be collected comply with the conditions of the data that can be collected, such as data formats and data types.

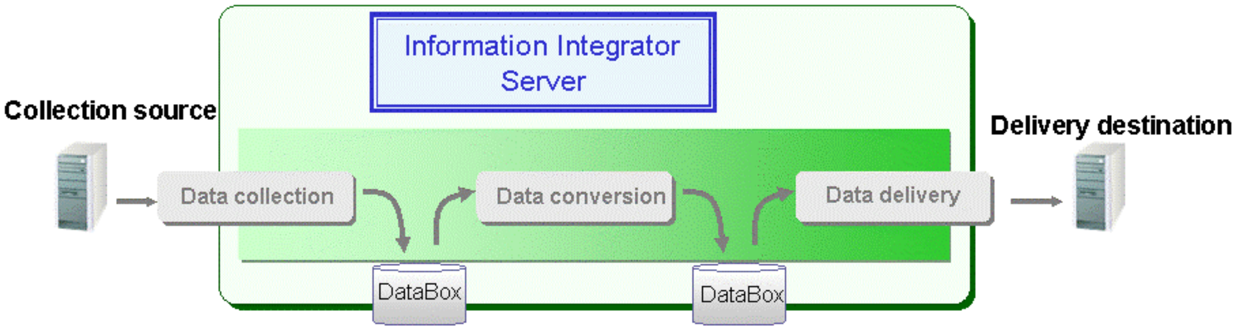

Extract metadata and display in III Studio or the design sheet when designing the Process Flow.
Collection source
Metadata

1.8.2 Data Management

Data collected from collection source system and data distributed to delivery destination system are maintained in the Information Integrator Server in Information Integrator. The area for retaining this data is called the 'DataBox'.

Multiple DataBoxes are created in the system. Data operations (such as storage and extractions) can be performed using the data in the DataBoxes.

Figure 1.23 Data management illustration



Types of DataBoxes

The following types of DataBoxes are available:

- DataBox
  - When used to store serial data

    A user can specify a DataBox in data collection and data delivery units. A DataBox can be used for such purposes as queue control between business applications and the system.

    A DataBox can be specified in the basic processing of a process definition, a data collection definition, a data filter definition, and a data delivery definition.

    The methods of performing operations on a DataBox are to use commands for data operations such as data storage, extraction, and deletion, and to use the Operation Management Client for data reference.

    Refer to the *Command Reference* and *Operation Management Client Help* for detailed information on these topics.

  - When used as a spool

    A user can specify a DataBox in data collection. A DataBox can be used if master data is managed by being accumulated during data collection.

    A DataBox can be handled as a spool by using commands for data operations such as data storage, extraction, and deletion.

Point

Prompts for deleting the data stored in the DataBox

The following two scenarios are the prompts for deleting the data stored in the DataBox:

- Deletion prompted by the elapse of "exctterm" key in the operating environment file

  On the basis of "resettime" key in the operating environment file, the data stored in the DataBox will be deleted automatically after the storage life of DataBox store information has elapsed. Note that if the Information Integrator Server was stopped, the data will be deleted next time the Information Integrator Server is started. Refer to the *Setup Guide* for detailed information on the operating environment file.

- Deletion due to the deletion request of a user

  The data stored in the DataBox can be deleted by executing the ifirdelt (Delete data) command. Refer to the *Command Reference* for details.

- Temporary DataBox

  When data is to be processed in the Process Flow, a temporary DataBox is used to temporarily save the process result data of functions such as data conversion and filtering so that data delivery between the functions can be performed.

  A temporary DataBox can be used to perform recovery operation and confirm the data flow if an error has occurred in a function.

  A temporary DataBox cannot be used to extract data once it has been extracted, but it can be used for data reference.

Point

Prompts for deleting the data stored in the temporary DataBox

The following two scenarios are the prompts for deleting the data stored in the temporary DataBox:

- Deletion prompted by the completion of a process

  If 'proc' was specified in "tmpdboxdel" key in the operating environment file and if "tmpdboxdelterm" key was specified, the data stored in the temporary DataBox will be deleted automatically at the completion of the process.

- Deletion after one day has elapsed since the completion of a process

  If "tmpdboxdel" key was omitted in the operating environment file or if 'fix' was specified, the data stored in the temporary DataBox will be deleted automatically after one day has elapsed since the completion of the process, on the basis of "resettime" key in the operating environment file. Note that if the Information Integrator Server was stopped, the data will be deleted next time the Information Integrator Server is started.

Refer to the *Setup Guide* for detailed information on the operating environment file.

Management

Data is stored sequentially in the 'DataBox' as generations. The system automatically allocates a unique data number to each generation of data. A user can identify the data on the basis of this number.

For each generation of data, the following structure is used to manage the data management information and the data:

- Management information

  The following information is used for management:

  - Information that signifies the data (such as data number, data size, and storage date and time)

  - Information on processes (functions) that have performed data operations

  - Data status

- Data

  Data is managed in files using units of data under "stk_path" key in the operating environment file.

Handled data

Data can be managed in a DataBox regardless of whether it has a CSV, binary, or other data format.

In addition, no limits are placed on the data size or on the upper limit of the number of generations per DataBox.

Note

Data with different structures or formats cannot be stored in a single DataBox.

Note that data with different structures and formats can be stored in a single DataBox in the following case:

- If Mixed format was specified as the data format in the data structure definition corresponding to the DataBox

Storage status of a DataBox

The following table shows the storage statuses of a DataBox. The storage status of a DataBox will transition through each of these statuses in the order shown in the table, from the top down.

If an error is detected in a storage process or extraction process, the DataBox will return to the status it had before execution. Data being processed by a function will have a storing or extracting status until the function ends successfully or with errors.

Table 1.12 Storage status

Status	Description
Storing	The storage process is in progress. Data undergoing data collection will also have this status.
Stored	The storage process has completed. Extraction is now possible.
Extracting	The extraction process is in progress. Data undergoing data delivery will also have this status.
Extracted	The extraction process has completed.

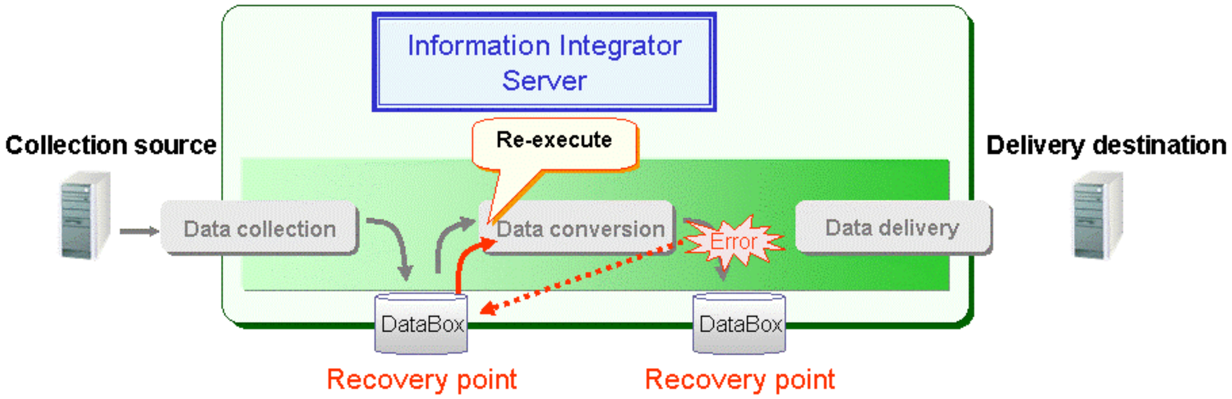
Recovery point

If an error is detected during execution of the Process Flow, it will be necessary to recover the business by reexecuting the Process Flow, but there is a checkpoint to enable reexecution to be started partway through the processing. This checkpoint is known as the 'Recovery point'.

The recovery point refers the point at which processing results at the time of completion of data collection, and also the processing results prior to data delivery are retained.

If reexecution is requested, recovery can be performed efficiently by using the recovery point immediately prior to the process in which the error occurred (the point at which data collection was completed, or the point immediately prior to delivery) to skip those processes already completed up until that point.

Figure 1.24 Recovery point illustration



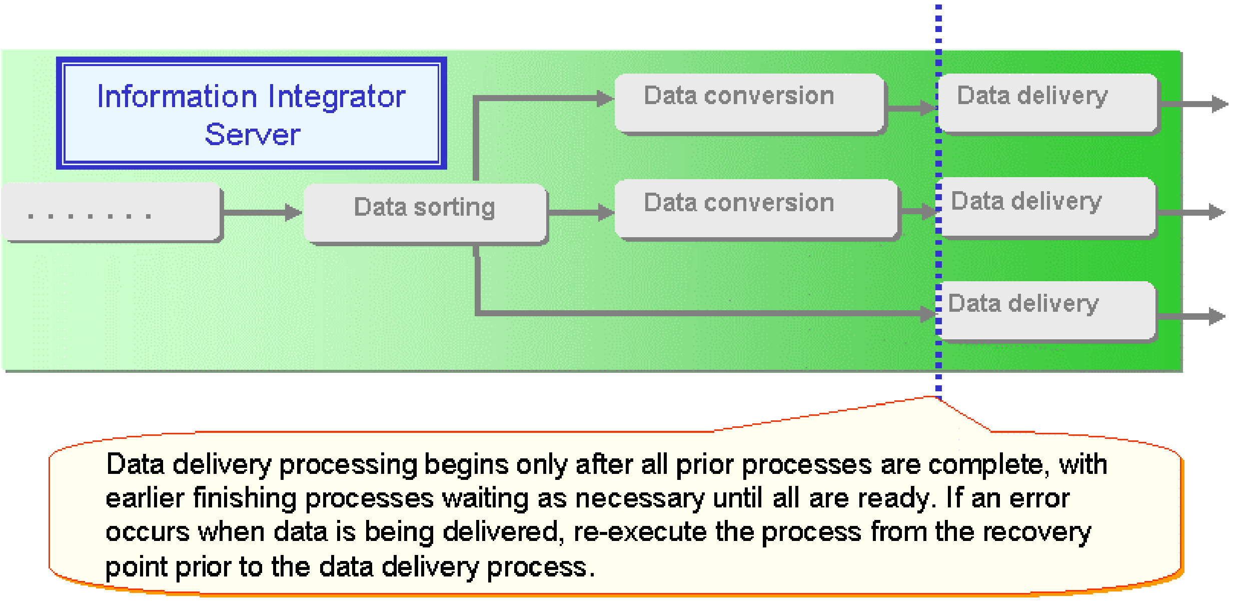
Point

A note on recovery for Process Flows comprising multiple data delivery processes

As illustrated below, Process Flows comprising multiple data delivery processes will wait until all pre-data delivery processes are complete before starting each of its data delivery processes.

Accordingly, even if an error occurs in any of the data delivery processes, it will still be possible to perform recovery from the recovery point immediately prior to the process in which the error occurred (the point immediately prior to data delivery).

Figure 1.25 Waiting for the delivery process



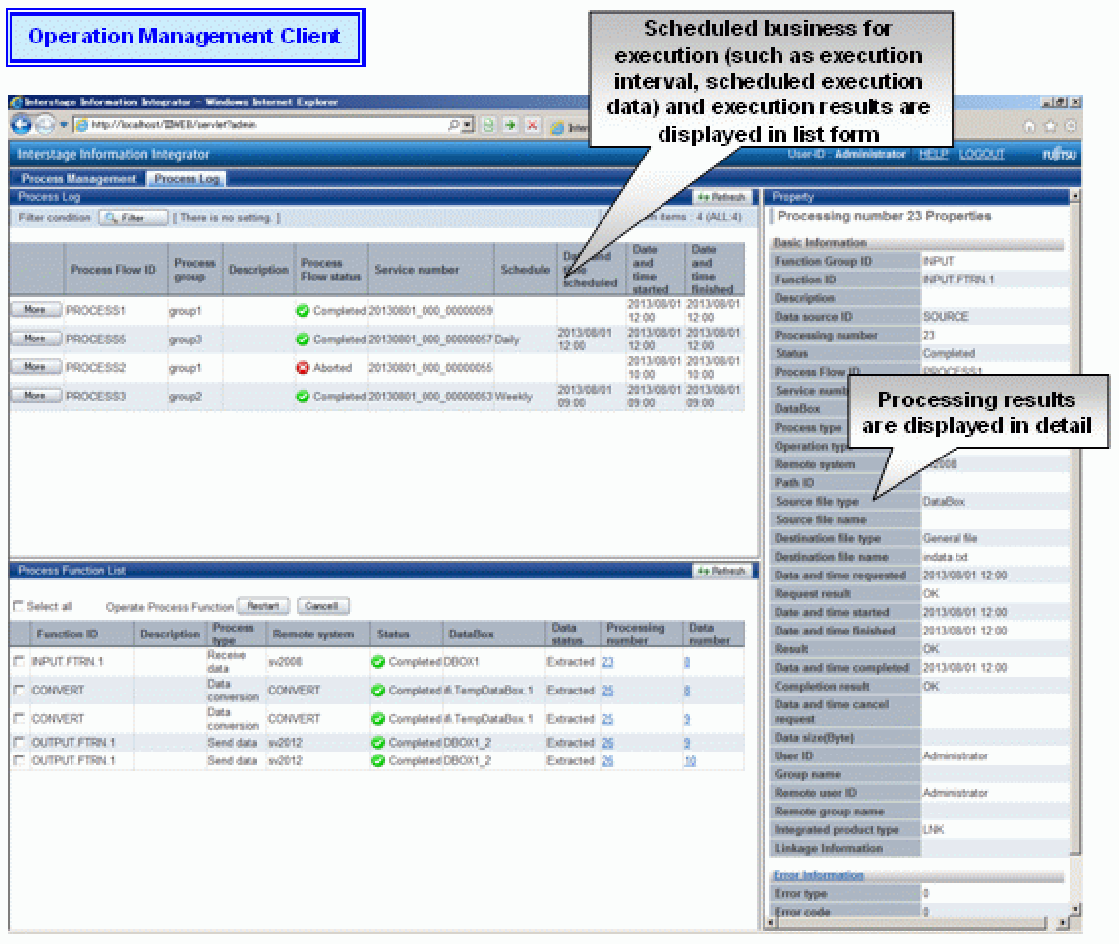
Data delivery processing begins only after all prior processes are complete, with earlier finishing processes waiting as necessary until all are ready. If an error occurs when data is being delivered, re-execute the process from the recovery point prior to the data delivery process.

1.8.3 Operation Management

This feature manages the operating state of the Process Flow registered in the execution environment. The tool to manage the operating state of the Process Flow is known as the 'Operation Management Client'.

The Operation Management Client displays the forecast vs. actual status of the Process Flow in list form. Furthermore, this list provides easy confirmation of the processing results and resulting data for the Process Flow at the time of the error, making it easy for the user to deal with the error.

Figure 1.26 Operation management illustration



Operation Management Client

Scheduled business for execution (such as execution interval, scheduled execution data) and execution results are displayed in list form

Processing results are displayed in detail

Interstage Information Integrator - Windows Internet Explorer
http://localhost/IIWEB/servlet?admin
Interstage Information Integrator
User-ID : Administrator
HELP
LOGOUT
FUJITSU
Process Management
Process Log

Process Log

Filter condition [Filter] [There is no setting.]

items : 4 (ALL:4)

	Process Flow ID	Process group	Description	Process Flow status	Service number	Schedule	Date and time scheduled	Date and time started	Date and time finished
More	PROCESS1	group1		Completed	20130801_000_00000059			2013/08/01 12:00	2013/08/01 12:00
More	PROCESS6	group3		Completed	20130801_000_00000057	Daily	2013/08/01 12:00	2013/08/01 12:00	2013/08/01 12:00
More	PROCESS2	group1		Aborted	20130801_000_00000055			2013/08/01 10:00	2013/08/01 10:00
More	PROCESS3	group2		Completed	20130801_000_00000053	Weekly	2013/08/01 09:00	2013/08/01 09:00	2013/08/01 09:00

Process Function List

☐ Select all   Operate Process Function [Restart] [Cancel]

	Function ID	Description	Process type	Remote system	Status	DataBox	Data status	Processing number	Data number
☐	INPUT.FTRN.1		Receive data	sv2008	Completed	DBOX1	Extracted	23	8
☐	CONVERT		Data conversion	CONVERT	Completed	ifi.TempDataBox.1	Extracted	25	8
☐	CONVERT		Data conversion	CONVERT	Completed	ifi.TempDataBox.1	Extracted	25	9
☐	OUTPUT.FTRN.1		Send data	sv2012	Completed	DBOX1_2	Extracted	26	9
☐	OUTPUT.FTRN.1		Send data	sv2012	Completed	DBOX1_2	Extracted	26	10

Property

Processing number 23 Properties

Basic Information

Function Group ID	INPUT
Function ID	INPUT.FTRN.1
Description	
Data source ID	SOURCE
Processing number	23
Status	Completed
Process Flow ID	PROCESS1
Service numb	
DataBox	
Process type	
Operation typ	
Remote system	2008
Path ID	
Source file type	DataBox
Source file name	
Destination file type	General file
Destination file name	indata.txt
Data and time requested	2013/08/01 12:00
Request result	OK
Date and time started	2013/08/01 12:00
Date and time finished	2013/08/01 12:00
Result	OK
Data and time completed	2013/08/01 12:00
Completion result	OK
Data and time cancel request	
Data size(Byte)	
User ID	Administrator
Group name	
Remote user ID	Administrator
Remote group name	
Integrated product type	LINK
Linkage Information	

Error Information

Error type	0
Error code	0

1.9 III Agent Feature

The III Agent is a feature that enables data at the collection source to be converted at the collection source, or data located on the delivery destination system to be converted on the delivery destination system. For example, data that has been converted at the collection source is delivered in file format to the Information Integrator Server, or data is converted on the delivery destination system after being sent in file format from the Information Integrator Server. The load on the Information Integrator Server can be balanced by cutting out part of this Server's conversion functions.

Furthermore, it is also possible to retain operability as it is possible to centrally manage the system structure, operations, and even monitoring on the Information Integrator Server.

Refer to "List of Features that may be used with the III Agent" for more information on the differences between features of the III Agent and the Information Integrator Server.

Refer to the descriptions of the various features of the Information Integrator Server for information on the series of features and operations of the system structure, operations and monitoring.

1.9.1 Centralized Management of the Information Integrator Server

The special characteristics for centralized management on the Information Integrator Server for system structure, operations and through to monitoring are as described below.

Definition creation and management methods

Definitions for Process Flow design information are created and managed on the Information Integrator Server. The definition used by the III Agent is created on the Information Integrator Server and delivered to the III Agent for use.

Refer to the *Setup Guide* for detailed information on how the Process Flow definition is delivered from the Information Integrator Server to the III Agent.

Point

Making and the management of the definitions can be individually done on each III Agent. In this case, the method of making and management of the definitions are the same as the Information Integrator Server.

Operation Management

The Process Flow status of the III Agent can be referenced from the Operation Management Client of the III Agent. In addition, the Process Flow results of the III Agent can be referenced from the Information Integrator Server's Operation Management Client.

Refer to the *Operation Guide* for detailed information on the Operation Management Client.

Note

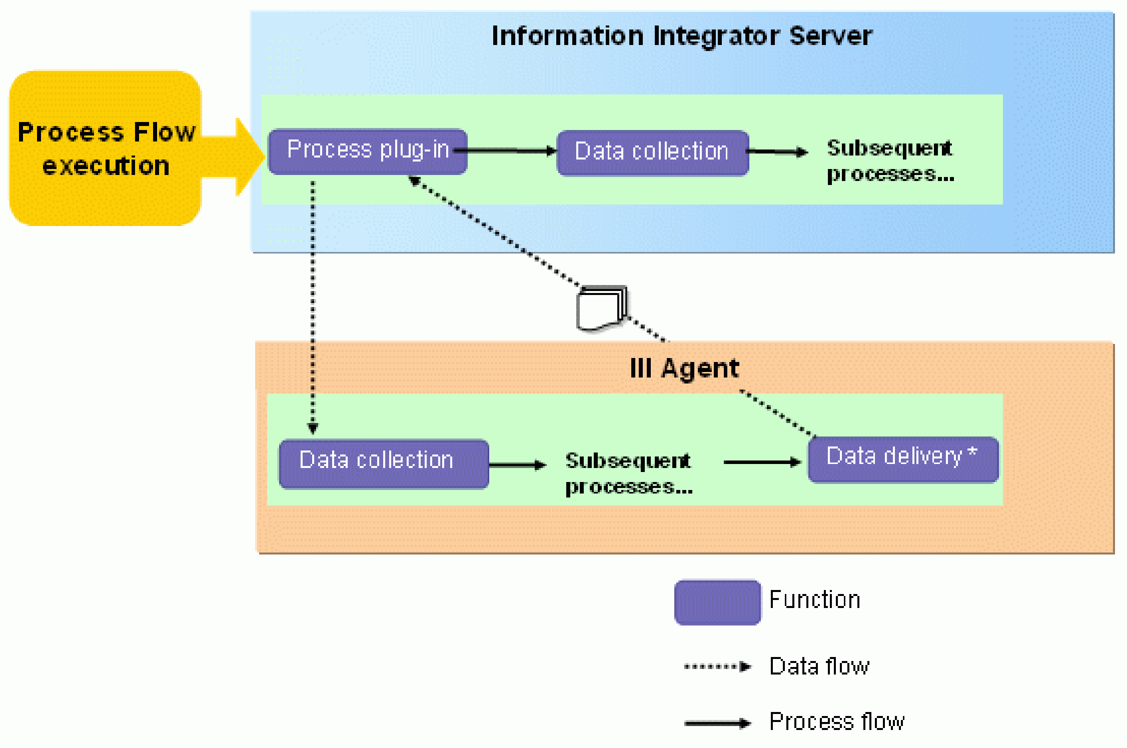
The Process Flow results of the III Agent can be referenced from the Information Integrator Server's Operation Management Client only if the process plug-in feature has been used to integrate with the Process Flow of each III Agent.

1.9.2 How to Operate the Process Flow

The Process Flow on the III Agent can be integrated with the Process Flow on the Information Integrator Server by executing a Process Flow on one of the servers in accordance with the usage format.

Some examples are illustrated below.

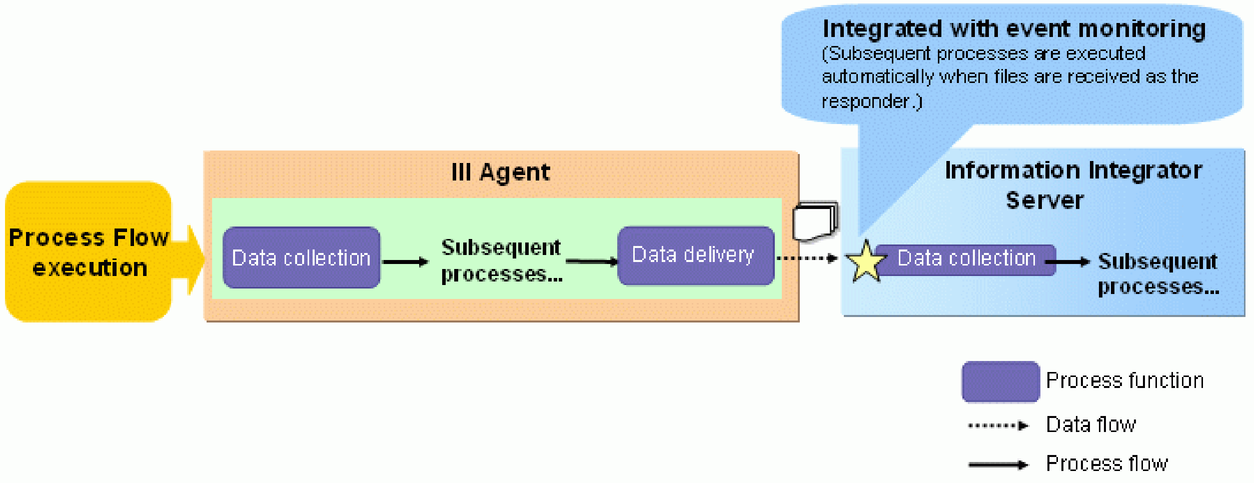
Figure 1.27 Example 1: Executing the Process Flow on the Information Integrator Server



*1: Data must be stored in the DataBox for III Agent data delivery.

When executing the Process Flow on the Information Integrator Server, use the process plug-in feature to integrate the Process Flow of each server. Refer to "3.10.1 Design Procedure for the Process Plug-in Feature" for detailed information on the process plug-in feature.

Figure 1.28 Example 2: Executing the Process Flow on the III Agent



When executing the Process Flow on the III Agent, use the execution of the Process Flow on the Information Integrator Server as a catalyst to operate the Server using event monitor. Then, when files delivered from the III Agent arrive at the Information Integrator Server, this action will act as a catalyst to begin execution of the Process Flow on the Information Integrator Server-side.

The event monitor used here utilizes the feature that is automatically executed by subsequent processes when files are received remotely. Refer to "1.6.5 Event Monitor" for more information.

Point

Format of files transferred between the III Agent and the Information Integrator Server

Design the process such that the 'FTP+' protocol is transferred between the III Agent and the Information Integrator Server. At this time, the file transfer concurrency design must take into account the communication of internal information.

1.10 III Transfer Accelerator

III Transfer Accelerator is software of data transfer accelerator. It speeds up data communications of applications that use TCP/IP between remote places (for instance, the data transfer tool and the user application, etc.). It uses original high-speed transfer protocols based on UDP and the communication bandwidth control technology that use the telecommunication bandwidth to its maximum. As a result, the reliability of the communication is secured achieving the cancellation of communication delay by long distance communication and the fast transfer.

It is possible to introduce III Transfer Accelerator into an existing System easily by the following features.

- High-speed data transfer is achieved only with software

  It is not necessary to introduce the hardware such as WAN acceleration device, and to reinforce the network line. High-speed data transfer is achieved only with software by using an existing network line.

- The adjustment of applications is unnecessary

  To use III Transfer Accelerator, it is not necessary to adjust the applications that use TCP/IP. In the data source, High-speed data transfer can be achieved only by modifying the configuration of destination of data transfer of the application.

- Making transmission data a file is unnecessary

  A past data transfer product transmits data reading transmission data made a file beforehand. By contrast in III Transfer Accelerator, the immediate data is received from the application, and III Transfer Accelerator is transmitted to the other party side system as it is. Therefore, you do not have to make a file beforehand about transmission data.

Refer to *Transfer Accelerator Setup Guide* for details of III Transfer Accelerator.

1.11 Using SSL Communication

This section explains how to operate the environment configuration and certificate management for performing SSL communication using the Information Integrator Server.

1.11.1 Certificate Management

In SSL communication, information called a 'certificate' is used to perform mutual authentication at connection. For a certificate, a public key is used for user authentication and the electronic signature of the issuing authority of the certificate is added to the certificate.

The authentication method will vary in the following ways according to the version of the SSL protocol:

- If the version of the SSL protocol is 2.0

  The initiator (Client) authenticates the responder (Server).

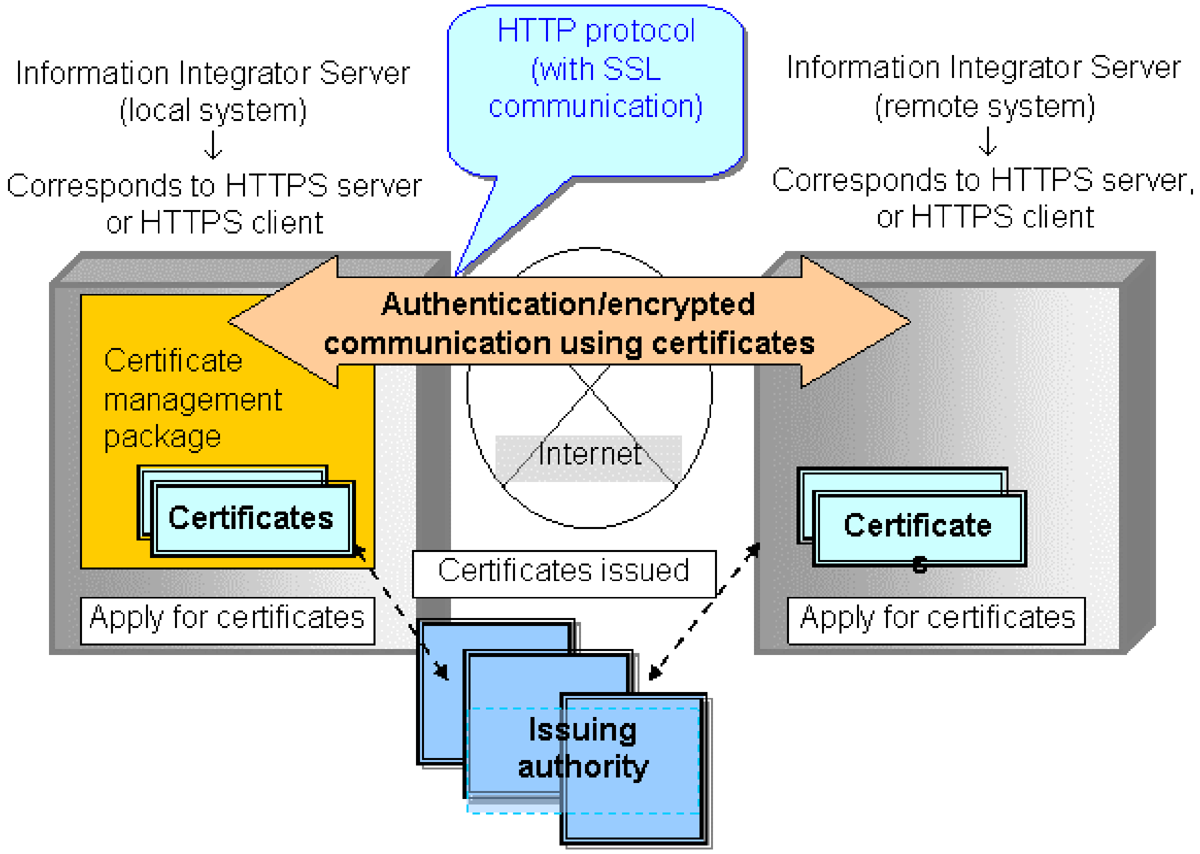
- If the version of the SSL protocol is 3.0

  Mutual authentication is performed.

To authenticate a remote system certificate by including the electronic signature that has been added to the certificate, the certificate of the issuing authority that issued the certificate of the remote system is required.

The relationship between the local system, the remote system, and the issuing authority is explained below.

Figure 1.29 Relationship between the local system, the remote system, and the issuing authority


Information Integrator Server
(local system)
↓
Corresponds to HTTPS server or HTTPS client
HTTP protocol (with SSL communication)
Information Integrator Server
(remote system)
↓
Corresponds to HTTPS server, or HTTPS client
Authentication/encrypted communication using certificates
Certificate management package
Certificates
Internet
Certificates issued
Certificate
s
Apply for certificates
Apply for certificates
Issuing authority

- The issuing authority refers to the system issuing the certificate to be used in SSL communication. The issuing authority is also called the 'Certificate Authority' or 'CA'.

Point

An Information Integrator Server supports certificates issued by the following product only:

- Systemwalker PKI Manager

- Certificates obtained from the issuing authority are registered in and managed by a certificate management package. Remote authentication and encrypted communication is then performed on the basis of the information managed by the certificate management package during SSL communication.

Point

The names of the certificate management packages are as follows:

- UNIX: FJSVsmee package
- Windows: F3FSSMEE package

1.11.2 Scope of Support of SSL Communication

Features that can use SSL communication

In an Information Integrator Server, SSL communication can be used when the HTTP protocol is used from the features listed below. Also define the HTTP protocol used in SSL communication as the HTTPS protocol in the data source definition of the Information Integrator Server.

- III Studio

Note that obtaining a local system certificate is not required for an Information Integration Server that is to communicate as an initiator using SSL 2.0.

Protocol version levels in SSL communication

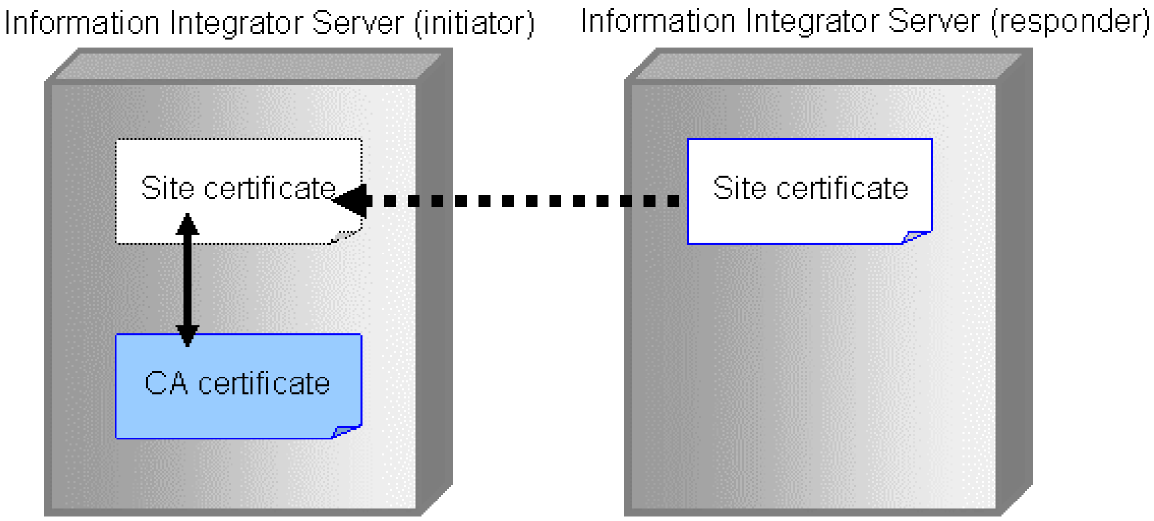
In SSL communication, the protocol version levels are SSL 2.0 and SSL 3.0.

The authentication process for each version level is described below.

- SSL 2.0

  This performs authentication of the responder. The responder must obtain a local system site certificate. The initiator need not obtain a certificate. Note, however, that the initiator must acquire the certificate of the issuing authority that issued a certificate to the responder in order to authenticate the certificate of the responder.

Figure 1.30 Authentication process for SSL 2.0


Information Integrator Server (initiator)
Information Integrator Server (responder)
Site certificate
CA certificate
Site certificate

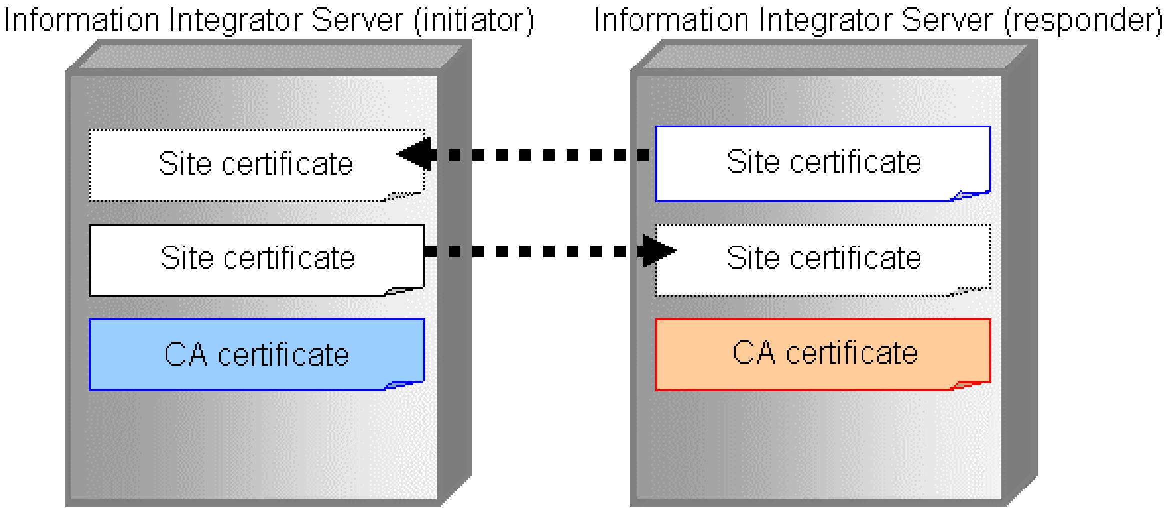
Note

For security reasons, SSL 2.0 is not recommended. Use SSL 3.0.

- SSL 3.0

  This performs mutual authentication of the initiator and the responder. Both systems must obtain a certificate. In addition, each system must acquire the certificate of the issuing authority that issued a certificate to the remote system in order to authenticate the certificate of the remote system.

Figure 1.31 Authentication process for SSL 3.0



Encrypted communication with a remote system in SSL communication

SSL communication using an Information Integration Server can be executed when the remote system is an Information Integration Server or Linkexpress server. Also, for an Information Integrator Server, SSL communication that performs encryption only is not supported. If the encryption of transfer data is to be performed using SSL communication, remote authentication by certificate is required.

Certificates and the encryption keys used in SSL communication are managed using a certificate management package. Refer to the *Setup Guide* for information on the environment configuration of the certificate management packages.

Note that encryption keys to be used in SSL communication must be created by both the initiator and the responder regardless of the protocol version level.

Chapter 2 Designing the Process Flow

In Information Integrator, the Process Flow refers to the series of processes that collect data from various business systems, performing data conversion processes and the delivery of optimized formatted data to the destination system.

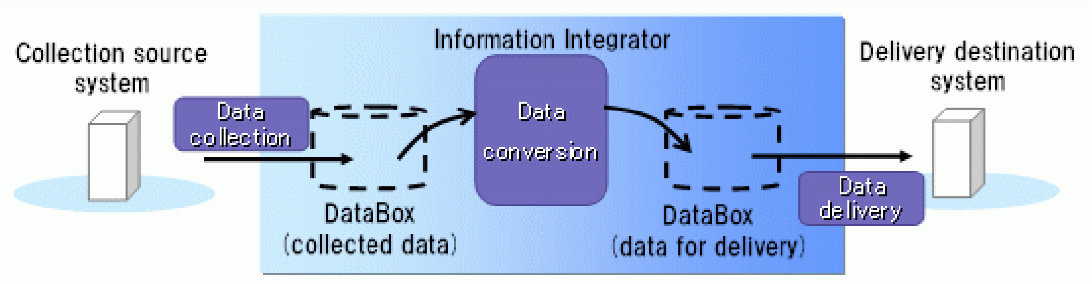
This chapter describes the design principles and procedures of the Process Flow in Information Integrator.

2.1 Overview of Process Flow Design

2.1.1 The Information Integrator Process Flow

The example below depicts the basic Process Flow in Information Integrator.

Figure 2.1 Example of Basic Process Flow



Information Integrator's data management feature collects data and stores it in the DataBox temporarily. Various required conversion processes are then performed on the data stored in the DataBox. The post-conversion data created in the DataBox is then extracted and delivered.

2.1.2 Process Flow Design

The Information Integrator Process Flow is designed as indicated below.

Designing the Process Flow

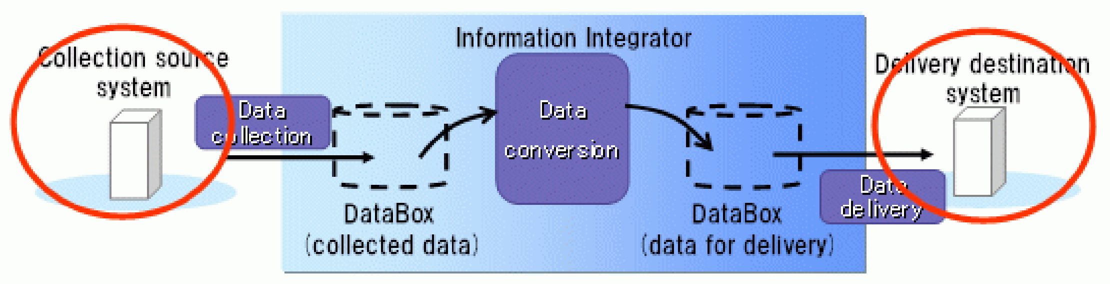
The entire Process Flow can be designed, including the matching and processing procedures for the input/output data for each of the collection, conversion and delivery processes, as well as the processing execution schedule.

Designing the data collection and delivery processes

1. Designing the data collection source and delivery destination

   Design the information (such as system name and IP address) of the systems that Information Integrator will access for data collection and as delivery destinations, as well as the procedures to access those systems (database information, parameters to send files, and authentication information).

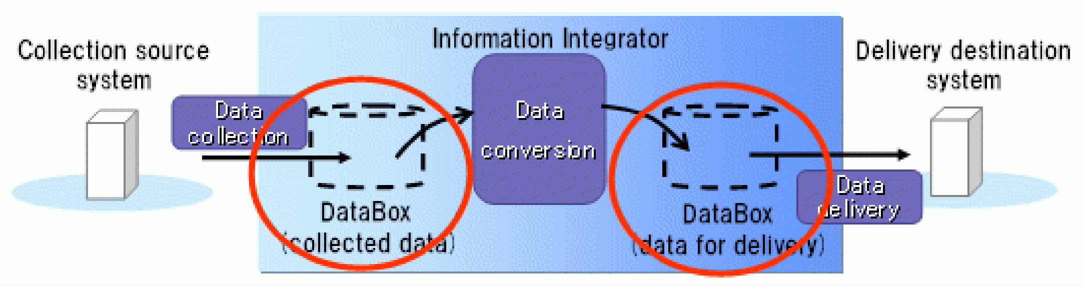
   Figure 2.2 Target sections to be designed for the data collection source and delivery destination



2. Designing the collected data and delivered data

   Design the data to be collected from the collection source, and the data to be delivered to the delivery destination. Data processed in Information Integrator is stored and processed in DataBoxes using Information Integrator's data management feature. The DataBoxes can be designed simply by specifying any unique DataBox ID in the relevant system. When performing conversion processes on the collected data, specify the data structure (such as item names and data attributes) and format attributes (such as file formats and character codes) for each item of data.

   Figure 2.3 Target sections to be designed for collected data or data for delivery

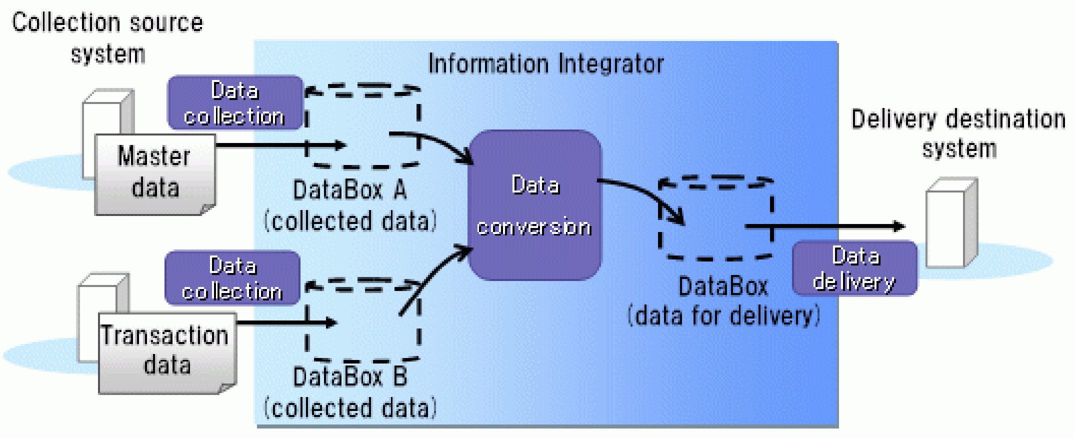
   

   A processing unit for collection processing comprising single or multiple items of data with the same data structure is known as a 'collection group'. Similarly, a processing unit for delivery processing comprising single or multiple items of data with the same data structure is known as a 'distribution group'.

   The DataBoxes storing the collected data or data for delivery are to be prepared for each collection group or distribution group, respectively. (The DataBox can be designed by simply specifying any unique ID in the system on the Information Integrator server).

   For example, to collect and use master data and transaction data with differing data structures, design the collection process by preparing separate DataBoxes for the collected data from the master data and the transaction data.

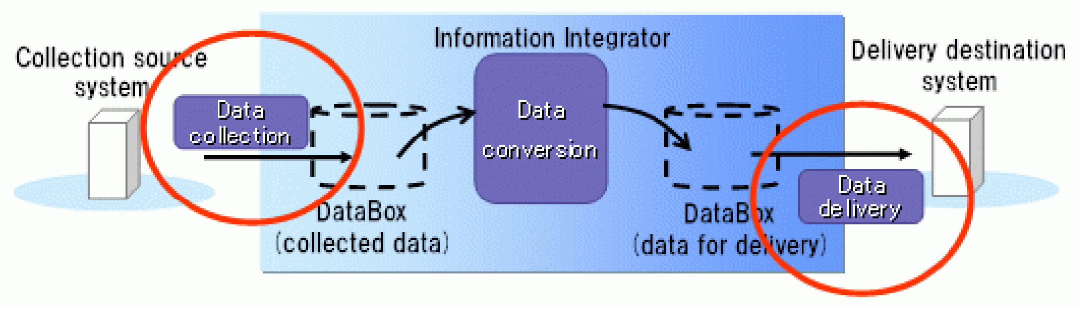
   Figure 2.4 Example of how data with different data structures is handled

3. Designing the collection and delivery methods

   Design the various parameters required for database extraction or the file transfer process in accordance with the format of the data in the collection source. Similarly, design the delivery method in accordance with the format of the data to be delivered.

   Figure 2.5 Target sections to be designed for collection and delivery methods

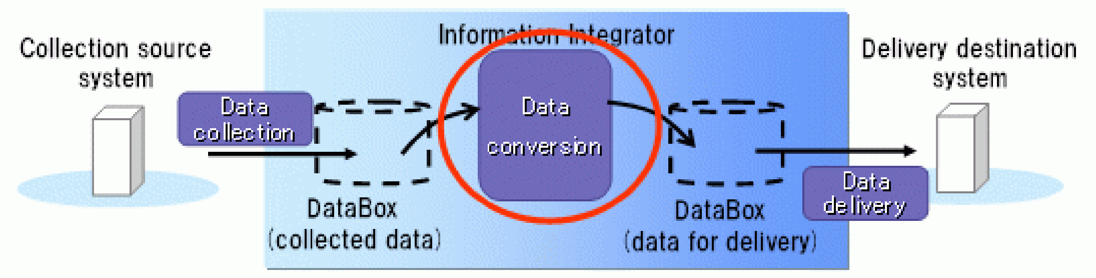
   

Designing the data conversion process

Design the process to convert collected data into data for delivery.

Design the data structure and format attributes of the pre-conversion and post-conversion data, and design the conversion process by mapping the matching of pre- and post-conversion items.

Figure 2.6 Target sections to be designed for the data conversion process



Designing external program calls (plug-ins)

When calling an external program as part of the Process Flow, design the external program information (such as the command names, parameters, and conditional process for the return value) and integrate a plug-in call process in the Process Flow processing order.

2.2 Overview of Process Flow Design Tasks

This section describes the basic approach when designing a Process Flow in Information Integrator. Design tasks are performed using III Studio.

When using III Studio, the processing procedures for the Process Flow can be confirmed visually whilst working on the design.

Register the designed Process Flow in the Integrator server to create an execution environment.

Refer to "Chapter 3 Designing Using III Studio" for detailed information on design tasks using III Studio.

Chapter 3 Designing Using III Studio

This chapter explains the Process Flow design procedures using III Studio.

Process Flow definitions are created using III Studio. Process Flow definitions are configured from the definitions indicated below.

- Design the Process Flow processing procedure

  Process definition

- Design the Process Flow parameters
  - Process Function process definitions included in the Process Flow
    - Data collection definition
    - Data conversion definition
    - Data filter definition
    - Data delivery definition
  - Design the parameters required for each of the above Process Functions
    - Data structure definition
    - Data source definition
    - Format definition
    - Plug-in definition

3.1 Starting and Exiting III Studio

3.1.1 Starting III Studio

The method for starting III Studio is described below.

Note

How to start III Studio for the first time

Assign the '-clean' option when starting Interstage Studio from **Run** or from the **Command prompt**.

- Example of the start command used with Interstage Studio Java EE V9.2.

```
IIIStudioinstallDir\IDE\0902_WB34\eclipse\isstudio.exe -clean
```

III Studio is a plug-in that operates on the Interstage Studio GUI-based integrated development environment (IDE). Select **Interstage Studio** from the **Start** menu to start Interstage Studio.

The initial window of Interstage Studio will be displayed.

Creating a new project

When using III Studio for the first time, create a location in which to store the III Studio definition (hereafter referred to as a 'project').

The method for creating projects is described below.

1. Click **File**, followed by **Create new**, and then click **Project**.

   The **New project** window will be displayed.

2. Select **III Studio,** followed by **III Project,** and then click **Next**.

   The **New III project** window will be displayed.

3. Specify the project name and the destination to which the project will be saved, and then click **Finish**.

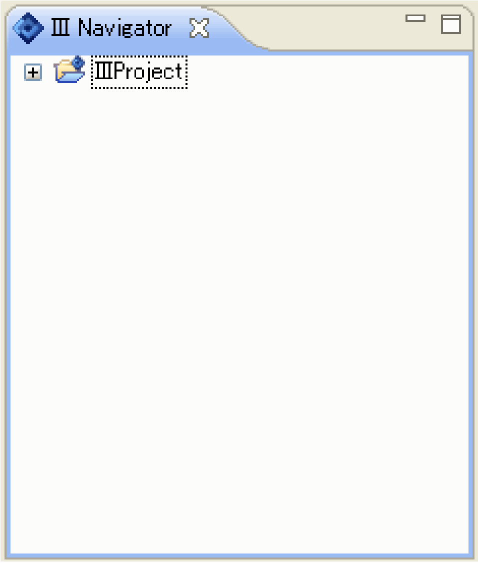
Note

Specifying the project name

- The half-width symbols listed below cannot be used.
    - " (Quotation marks)
    - * (Asterisk)
    - / (Forward slash)
    - : (Colon)
    - < Left angle bracket)
    - > (Right angle bracket)
    - ? (Question mark)
    - \ (Back slash)
    - | (Vertical bar)
- Do not use a period at the end of a project name.

Once the confirmation message displays, click **Yes** to create the new project.

The diagram below illustrates the creation of a project with project name 'iii_prj1'.



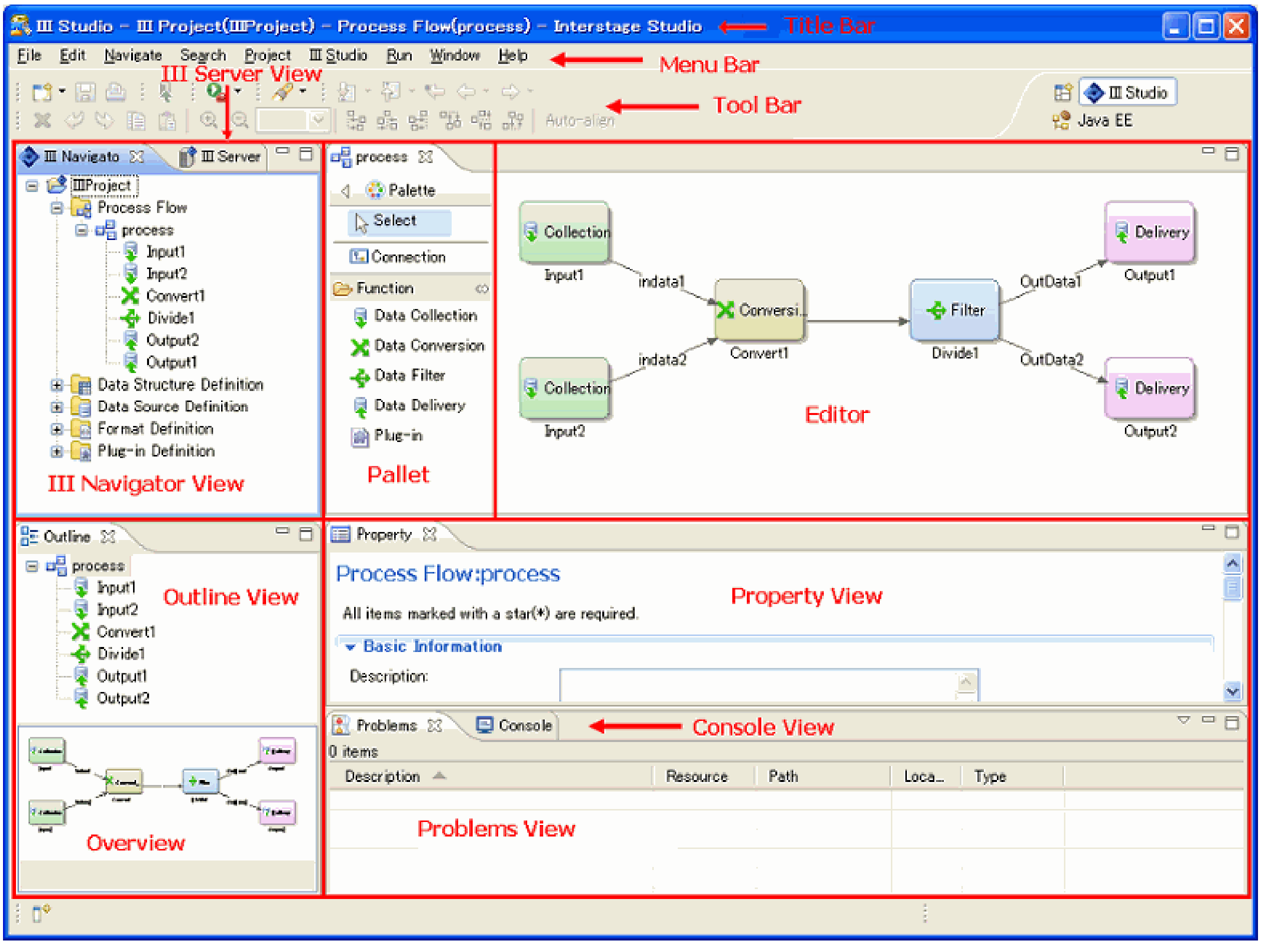
3.1.2 Ending III Studio

Click **File** and then click **End**.

The **III Studio** window will close.

3.2 View Pane Configurations for III Studio

The main view pane configurations used when designing Process Flows in III Studio are described below.



III Studio - III Project(IIIProject) - Process Flow(process) - Interstage Studio

Process Flow:process

All items marked with a star(*) are required.

0 items

- III Navigator view

  Displays the definitions that are currently being created, or that have been temporarily saved, in a tree format.

- III Server view

  Displays the definitions that have been registered in the Information Integrator Server in a tree format.

- Editor

  Designs Process Flow processing sequences and parameters for shared definitions.

- Overview

  Expand the Process Flow in **III Navigator view** to display the newly-created definition ID. Double-click the definition ID to display the content of the definition to be edited in the **Editor**. Refer to the design procedures for each definition, mentioned later in this document, for more information.

- III Property view

  Display and perform settings for each Process Flow and Process Function properties.

- Problems View

  Displays the details of any errors present in the definition content.

- Console view

  Displays registration progress information when registering definition information.

Refer to *III Studio Help* for information on each III Studio item.

3.3 Using III Studio

This section describes how to use III Studio.

3.3.1 Creating a New Definition

The procedures for creating each new Process Flow definition are indicated below.

1. Right-click the project that will store the definition displayed in **III Navigator view**.

   A pop-up menu will be displayed.

2. Select the definition to be created.

   - **New Process**

     Design the Process Flow procedures (the process definition) and the Process Function procedures (definitions for each of the data collection, data conversion, data filtering, and data delivery processes) included in the Process Flow.

   - **New data structure definition / New data source definition / New format definition / New plug-in definition**

     Design the parameters required for the Process Functions (definitions for each of the data collection, data conversion, data filtering, and data delivery processes) included in the Process Flow.

   The dialog box for entering information such as the definition IDs for managing each of the definitions will be displayed.

3. Enter definition IDs in the input fields for each dialog box and click **Finish**.

Point

- Ensure that the following information is entered at this point:

  - Each definition ID
  - The **Data source type** of the **New data source**
  - The **Plug-in type** of the **New plug-in**

  Note that it is also possible to enter and edit other input information at a later stage. Information can be input and edited in **Property view** or the **Editor**.

- The information for each definition is managed using its definition ID (such as the process definition ID or function ID). Definition IDs must be unique on the Information Integrator Server system. In order to prevent the definition IDs from being duplicated, we recommend designing them in advance using a convention to add a prefix or similar

- When using the Information Integrator Server and the III Agent at the same time, assign an ID so that the relationship between Process Flows can be confirmed after the operation. For example, the process definition IDs used in the III Agent could be numbered sequentially with the process definition IDs (the Process Flow used in the Information Integrator Server) with which they are integrated.

Expand the project name displayed in **III Navigator view** to display the created definition ID. Double-click the definition ID to display the edited content in the **Editor**. Refer to the design procedures for each definition, mentioned later in this document, for more information.

3.3.2 Registering Definitions in the Server

1. Connect to the Information Integrator Server. Click **III Studio** and then select **Server Connection**.

   The target server will be displayed in **III Server view**. Expand **III Server view** to display the registered definitions.

2. Right-click the definition to be registered in **III Navigator view**.

   A pop-up menu will be displayed.

3. Click **Register**.

   The definition will be registered in the Information Integrator Server.

4. Disconnect from the Information Integrator Server. Click **III Studio** and then select **Server Disconnection**.

   The definition information will no longer be displayed in **III Server view**.

3.3.3 Revising Definitions Registered in the Server

1. Right-click the definition to be revised in **III Server view**. Note that multiple definitions can also be selected.

   A pop-up menu will be displayed.

2. Click **Retrieve**.

   Select the project in which the definition(s) will be stored to store it in the relevant project under **III Navigator view**. The definition(s) can now be revised.

Point

If editing is stopped for some reason after retrieving the definition, the operation to retrieve the definition will also need to be canceled. Select the definition in **III Server view** and then select **Unlock** from the pop-up menu. Note that this operation can only be performed by the system administrator or by the user that obtained the relevant definition.

3.3.4 Temporarily Saving and Re-editing Definitions

- Saving

  Click **File** and then click **Save**.

  Definition information will be saved on the PC with III Studio installed.

- Re-editing

  Expand the tree menu in **III Navigator view** and double-click the appropriate definition.

  The target definition will be displayed in the **Editor**.

3.3.5 Copying Definitions

1. Right-click the definition ID to be copied in **III Navigator view**.

   The pop-up menu will be displayed.

2. Click **Copy**, and then from the pop-up menu click **Paste**.

   The window for editing definition IDs will be displayed.

3. Enter the definition ID and click **OK**.

   The copied definition ID will now be displayed in **III Navigator view**.

Modifying, editing and registering copied definition IDs enables them to be created without repeating the same definition design.

3.4 Designing the Process Flow

This section describes the design of Process Flow processing procedures.

Design procedures are as follows:

Confirming the properties

Open the Process Flow in the **Editor** and select an area that does not contain any Process Functions or connections. This will display the properties of the Process Flow in **Property view**.

The properties of the Process Function can also be displayed in **Property view** by selecting the Process Function in the Editor.

- How to specify the settings for the entire Process Flow

  This section describes how to specify the settings for the entire Process Flow, including the process group ID and the execution schedule.

  a. Specifying the process group

     Specify the process group in accordance with the user design.

  b. Sequence control

     Specify whether to perform sequence control on identical Process Flows.

  c. The Process Flow execution schedule

     When setting the Process Flow execution schedule, set the schedule type and the start time.

     Do not perform these settings when starting processes using a command from an external application.

     Point

     For processes that execute repeatedly at fixed intervals and processes that execute event (collection) monitoring, the schedule time period can be set in advance. For example, in Information Integrator it is possible to set processes to execute repeatedly at 60 minute intervals and processes to execute event monitoring during a time period of 09:00 to 17:00.

     These settings are performed in the operating environment file. Refer to the *Setup Guide* for information on these settings.

     Example

     Performing Event Monitoring

Setting items	Settings
Schedule type	Event Monitor

     Note that it is not possible to mix monitored and non-monitored data in a single Process Flow. When event monitoring is selected, ensure that **Yes** is defined in the 'collection monitor' of the data collection definition.

     Refer to *III Studio Help* for information on each item.

     Also, refer to "3.6.3 Designing the Collection Method" for information on the data collection definition settings when performing event monitoring.

  d. Recovery point

     Specify whether or not to use a recovery point.

Designing the processing procedure

Design the processing procedure for the Process Flow in the **Editor** is as follows:

1. Double-click the Process Flow to be designed in **III Navigator view**.

   A new tab will be added in the **Editor**, and the editing area of the selected Process Flow will be displayed.

2. Add a Process Function. Select **Process Function** from the **Palette** and drag it to the canvas.

   A new Process Function will be created on the canvas. Modify the definition ID as necessary.

   This operation can be repeated multiple times when creating multiple Process Functions.

3. Specify the processing sequence for the Process Function. Select **Connection** from the **Palette** and sequentially click the Process Functions that correspond to the preceding and subsequent processes that were added to the canvas.

   An arrow will be displayed indicating the processing order between selected Process Functions.

   Note that the data ID will also be displayed above the arrow when there are multiple input-output data items. This operation can be repeated multiple times when specifying the processing sequence for multiple Process Functions.

Refer to "3.4.1 Designing the Delivery of Data between Process Functions" for information on data IDs.

4. Modify the data ID as necessary. Select from the **Palette** and then select the arrow displaying the data ID before clicking on the arrow again.

5. Specify the ID.

   - When the connecting subsequent process is a data conversion definition

     Specify the ID (conversion data ID) for the input data of the data conversion definition in the format 'indataN (N: an integer from 1)'.

   - When the connecting preceding process is a data filter definition

     Specify the ID (filter data ID) for the output data of the data filter definition in the format 'OutDataN (N: integer from 1 to 100)'.

6. Check for errors in the definition content of any Process Flows created in III Studio.

   In the event of errors or warnings, the information will be displayed in **III Problem View**. In this case, take action in accordance with the displayed content.

7. Save the design information. Click **File** and then click **Save**.

   The Process Flow information will be saved.

Note

- The data collection and data delivery processes must be created.
- Specify data collection in the first process of the process definition, and specify data delivery in the last process of the process definition. Note, however, that processes using the process plug-in feature can be specified as the first and last processes.
- When specifying a process that uses the process plug-in feature in the first process, ensure that the corresponding data collection process is specified in the subsequent process. Additionally, if there are multiple subsequent data collection processes present in this event, prepare the same number of plug-in processes as there are data collection processes.

3.4.1 Designing the Delivery of Data between Process Functions

When the input data (pre-conversion data) for data conversion or the output data filtering results in multiple data files, the correlation of data delivered between Process Functions must be designed.

A data ID is automatically set at the time of design for input data in data conversion and for output data in data filtering. ('Conversion data ID' and 'Data filter definition ID').

Data mapping of data in the Process Function process is designed by specifying **Conversion data IDs** and **Filtering data IDs** for the Process Functions before and after the process definition.

Note

- It is not possible to select multiple data filtering processes as preceding processes.
- When subsequent processes of data filtering end in an error, data filtering will wait to be restarted. Use the ififrstr command to restart data filtering. (It is not possible to reexecute from the beginning of the process).

  Refer to the *Command Reference* for information on commands.
- Do not create data filtering as subsequent processes of the data filtering in Process Flows that are included in the 'output to service bus' of Service Integrator.

3.5 Designing the Process Flow Parameters

Design the Process Flow parameters.

Each of the collection, conversion and delivery processes executed within a single Process Flow is referred to as a Process Functions. These Process Functions must be defined.

The Process Function types and each of the Process Functions are described below.

Table 3.1 Process Function types

Process Function types	Processes recognized as single Process Functions
Data collection	In this process, single or multiple data having the same data structure are stored in a DataBox (collection group).
Data conversion	In this process, conversion processes are performed on input data, comprising single or multiple data files, and output as single data files.
Data filtering	In this process, the input data, comprising single data, is filtered into multiple data for each record in accordance with the filter conditions.
Data delivery	In this process, data stored in a DataBox is delivered to one or more data delivery systems (delivery groups).
Plug-in	This process calls a single plug-in (external program).

Each of the Process Functions included in the Process Flow is defined.

3.5.1 Definition Types for Designing the Process Flow Parameters

The definitions for designing Process Flow parameters are listed below.

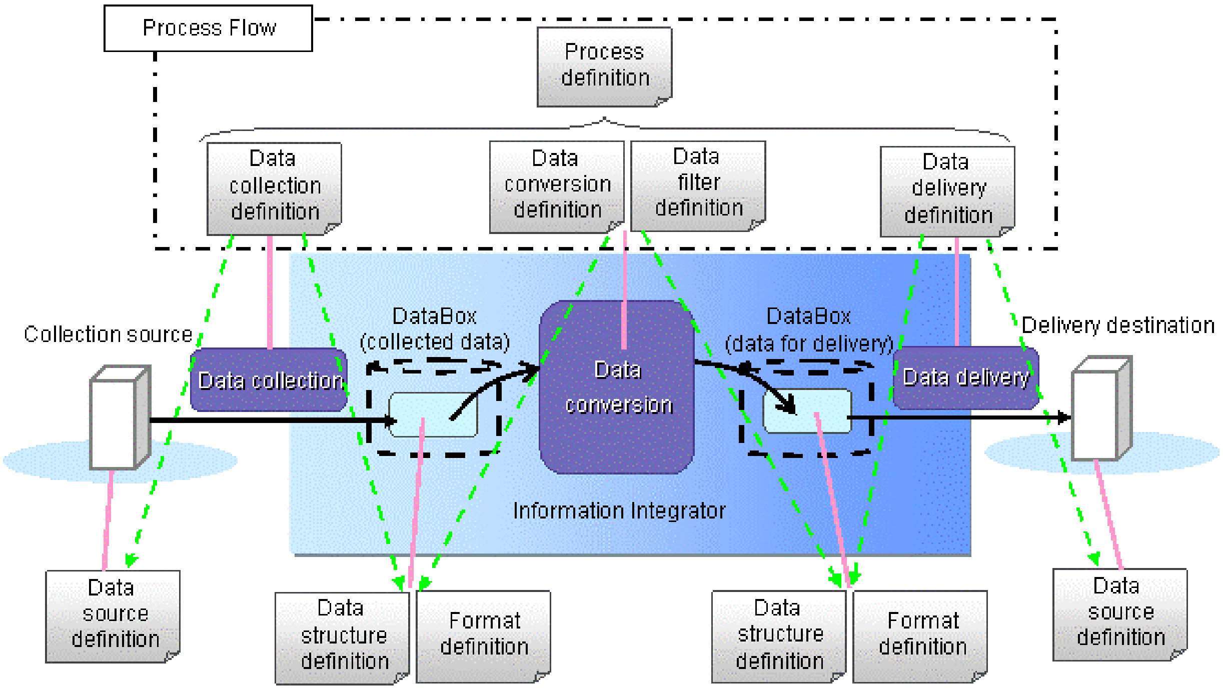
Table 3.2 Definitions for designing Process Flow parameters

Definition name	Definition outline
Data collection definition	Defines the location and collection way of the data collection feature.
Data conversion definition	Defines the content (format conversion, edit item, record editing) of the data conversion. Define the data structure and format of the pre-conversion and post-conversion (input and output) data, and define the mapping of items. Also define editing for the entire record for each item that uses arithmetic expressions, such as aggregate, sort and join.
Data filter definition	Defines the process for filtering data in accordance with the filter conditions, such that each record is filtered into multiple data files.
Data delivery definition	Defines the delivery destination location and delivery procedure for the data delivery process.
Data source definition	Defines the basic information (such as system information and access method) required for performing data collection or delivery processes on the collection source and delivery destination systems. This information is referred to using ID from the data collection definition and data delivery definition.
Format definition	Defines the character code and data format as the pre-conversion and post-conversion data formats when performing data conversion.
Data structure definition	Defines the item name, data attributes and number of columns as the pre-conversion and post-conversion data structure when performing data conversion.
Plug-in definition	Defines the plug-in (external program) information to be called during data processing.

The range of definitions and matching for each definition in the Process Flow is illustrated in the diagram below. (Although the plug-in definition is not shown in this diagram, it is created when the plug-in is called during the Process Flow).

The dotted green arrows indicate the reference relationships between definitions. For example, the data source definition information for the collection source is referred to from the data collection definition.

Figure 3.1 Matching the range to be designed for each definition

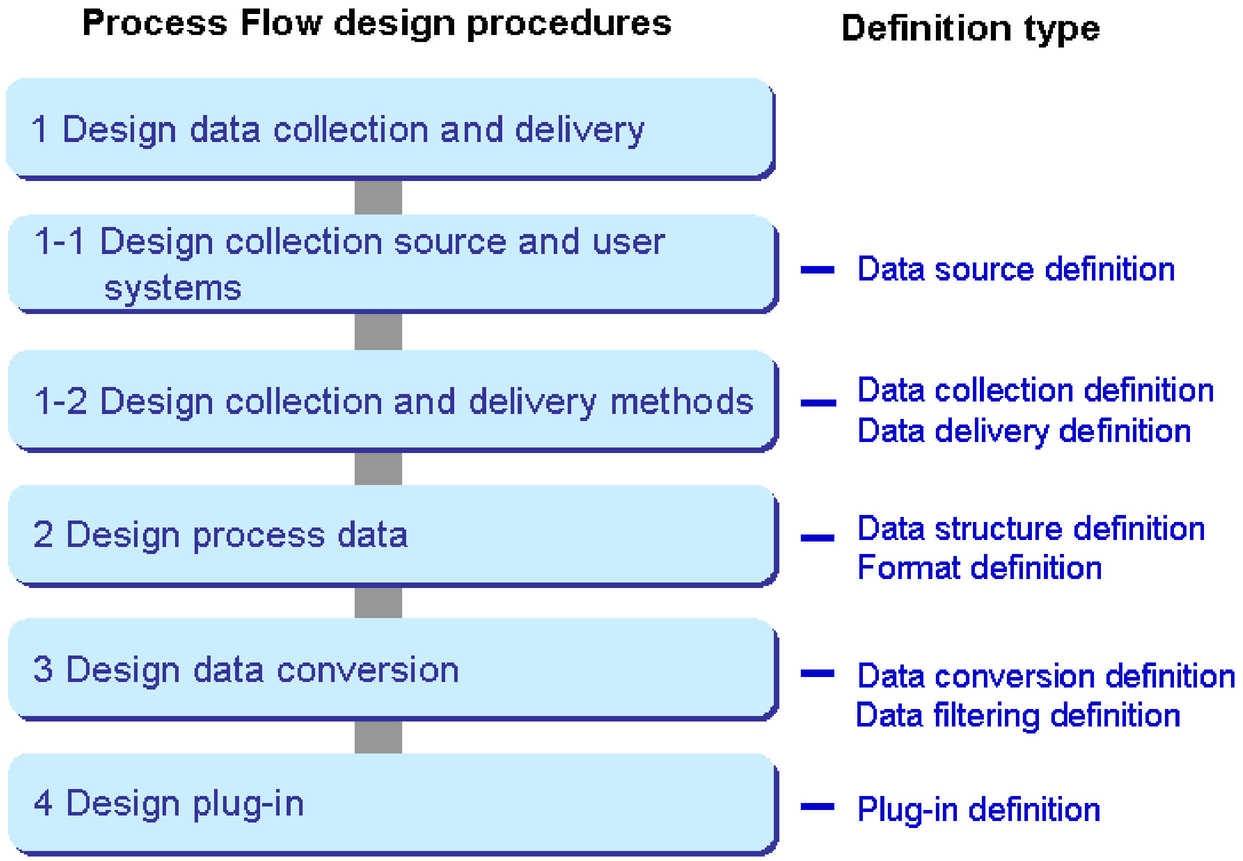


The process definition designs the processing procedure of Process Flows. Refer to "3.4 Designing the Process Flow" for more information.

3.5.2 Design Procedure for Process Flow Parameters

Create each definition and design the Process Flow parameters. The design flow and definitions used when designing each process are illustrated below.

Figure 3.2 Design procedure for Process Flow parameters


Process Flow design procedures
Definition type
1 Design data collection and delivery
1-1 Design collection source and user systems
Data source definition
1-2 Design collection and delivery methods
Data collection definition
Data delivery definition
2 Design process data
Data structure definition
Format definition
3 Design data conversion
Data conversion definition
Data filtering definition
4 Design plug-in
Plug-in definition

Refer to subsequent sections in this document for more information on each design procedure.

3.5.3 Notes for Handling Data with Title Lines

The following notes should be adhered to when handling data with title lines in data conversion or data filtering:

- When performing data conversion or data filtering on data with title lines
    - When collecting data in advance for conversion (during data collection), ensure that the data to be collected is stored individually in DataBoxes.

      Specify **Separate type** in the data collection definition as the method for storing in the DataBox.
    - Data conversion and data filtering can only be performed on data for conversion when its data structure is in 'General' format.

      Specify **General** as the 'data format' of the pre-conversion data in the data structure definition.
    - It is not possible to specify horizontal concatenation joins (INNER JOIN, OUTER JOIN) in data containing a title line in the master data.

      The format ID set in 'indata2' onwards in the pre-conversion data can only be used when 'No' is set in 'Include header' in the format definition.
    - Vertical concatenation unions (UNION, UNION ALL) cannot be performed.
    - It is not possible to use the record extraction feature for pre-conversion data.
- When adding a title line following execution of data conversion or data filtering
    - Data conversion and data filtering can only be performed on data for conversion when its data structure is in 'General' format.

      Specify **General** in the 'data format' of the post-conversion data in the data structure definition.

    - Generate a title line based on the item name. Conversion/output may not execute correctly depending on the post-conversion character code specified.

Refer to the tooltip in each III Studio item or to *III Studio Help* for more information on III Studio.

3.6 Designing the Data Collection Feature

This section describes the design procedure for the data collection feature.

The data collection feature is designed as follows:

1. Design the collection source
2. Design the data for collection
3. Design the collection method

3.6.1 Designing the Collection Source

Design the system information that will be accessed as the data collection source (such as the system name and IP address), as well as the procedures to access the system (such as database information, parameters to receive files, and authentication information).

Creating a data source definition

Expand the data source definition in **III Navigator view** and double-click the data source definition ID to be designed. Each data source definition item will be displayed in the **Editor**. The data source definition can now be created.

Point

The items displayed in the **Editor** will vary according to the data source type. Select the data source type when creating a new data source definition. Refer to "3.3.1 Creating a New Definition" for information.

- When extracting data from a collection source database:

  Enter the parameters relating to the collection source database.

- When collecting data from a collection source using 'receive files':

  Enter the parameters required for the collection source and for 'receive files'.

- When extracting data from a Salesforce collection source:

  Enter the parameters required for the collection source and for 'extract object'.

- When collecting BLOB data from an Azure Storage collection source:

  Enter the parameters required for the collection source and for BLOB collection.

When performing data delivery processes on other Process Flows or using the Information Integrator DataBox store command to execute processes using data stored in the DataBox, enter only the parameters for the 'common' portion of the data collection definition.

Point

When a data source definition has already been registered in the server, specify the definition ID of the registered data source definition in the 'Data source ID' of the data collection definition to use as a reference. There is no need to recreate the same data source definition.

3.6.2 Designing the Data for Collection

Creating DataBox names

Decide on a name for the DataBox that will store the collected data. DataBoxes are necessary to collect and process data using the Information Integrator Server. Enter the DataBox name in **Property view**. Each data collection definition item can be displayed in **Property**

view by selecting the data collection definition to be created in the **Editor**. Click **Common** and then **DataBox** and enter the DataBox name from amongst the items displayed.

A unique value must be set for the DataBox name in the Process Flow. Data can be delivered between multiple processes by setting the same DataBox name in multiple processes.

Designing the data structure and format of the data for collection

The data structure and format of the data to be collected and stored in the DataBox are designed in the data structure definition and format definition, respectively.

Point

It will not be necessary to design the structure or format of data to be collected if the conditions listed below are satisfied.

- Data collection occurs in file format or in service bus integration format
- The collected data is sent directly to the delivery Process Flow without any conversion processing being performed

Format design is not required in the following cases:

- When 'extract object' is used for data collection
- When BLOB collection is used for data collection

3.6.3 Designing the Collection Method

The collection method is designed using the data collection definition. The design procedure is explained below.

Creating a data collection definition

Expand the Process Flow in **III Navigator view** and double-click the Process Flow definition ID to be designed. The processing procedures for the Process Flow will be displayed in the **Editor**. Each data collection definition item can be displayed in **Property view** by selecting the data collection definition to be created in the **Editor**. The data collection definition can now be created.

Example

Performing event monitoring

Setting items	Settings
Operation type	Responder
Monitor collection	Yes
Action when not found	Normal Specify when not waiting for an event.
	Error Specify when waiting for an event.
Collected file name	Specify according to whether a perfect match or wildcard '*'. Only specify for directory monitoring. - When a wildcard has been specified and the name of the file for collection 'totals 230 bytes or more for the name of the directory to be monitored + backup directory name + file name': The file will always be outside the scope for processing, even if the name of the file for processing conforms to the specified value.

Setting items	Settings
	Furthermore, files that will not be executed in the Process Flow will always be deleted, regardless of whether the <*monitorexcept*> parameter of the operating environment file has been specified. Note that when the name of the file for collection 'totals less than 230 bytes for name of the directory to be monitored + backup directory name + file name', files that conform to the specified value of the wildcard specification will be processed. Furthermore, files that will not be executed in the Process Flow will be processed in accordance with the <*monitorexcept*> parameter of the operating environment file.

Note that a separate data collection definition should be prepared for each event monitor target when monitoring multiple events and when the Process Flow is waiting until all executed files have arrived. Only one data source can be specified per data source definition.

Refer to *III Studio Help* for information on each item.

Also, refer to "3.4 Designing the Process Flow" for information on the process definition settings when performing event monitoring.

When specifying 'Collection type' in 'DataBox storage' and multiple events occur simultaneously

Simultaneously execute the same number of Process Flows when there are events. Depending on the timing, data might be collected and processed at the first Process Flow executed, while other Process Flows may be recognized as zero byte data. In this event, specify **Error** in **Action when not found**.

When 'Error' is specified in 'Action when 0 bytes'

Even when a zero byte data file is placed in a directory for monitoring, and that file conforms to files for processing, it will be processed as a non-target file.

Specifying data source definition references

Specification of data source definition references is as follows:

1. In **Property view** of the data collection definition, click **Collection Definitions** followed by **the data source list** and then click **Add**.

   This action will display a window, from which the definition ID of the data source definition to be referenced can be specified, and from which the collection source type can be selected.

   Point

   Collection source type

   Configure the data format of the collection source. The settings values are as follows:

   - Database
   - File transfer
   - Salesforce.com
   - Interstage Information Storage
   - Azure Storage (BLOB)
   - Interstage Service Integrator

2. Enter the definition ID and type.

   Click **Complete** to perform reference settings for the data source definition.

Settings for collected data

Specify the definition IDs listed below in each data collection definition.

- The 'DataBox' in which the collected data will be stored
- The 'data structure definition ID' of the data structure definition of the collected data
- The 'format ID' of the format definition of the collected data

Point

It will not be necessary to specify the 'data structure definition ID' or the 'format ID' if the conditions listed below is satisfied.

- When data collection occurs in file format or in service bus integration format
- When the collected data is sent directly to the delivery Process Flow without any conversion processing being performed

It is not necessary to specify the 'format ID' in the following cases:

- When 'extract object' is used for data collection
- When BLOB collection is used for data collection

Designing the parameters required for collection processing

In **Property view** of the data collection definition, click **Collection Definitions** followed by **the data source list** and then select the data source definition ID for setting the required parameters. The **Data Source Details** will be displayed. Set each of the parameters required for collection processing.

3.7 Designing the Data Delivery Process

The data delivery process is designed as follows:

1. Design the data delivery destination
2. Design the delivery data
3. Design the delivery method

3.7.1 Designing the Data Delivery Destination

Design the system information that will be accessed as data delivery destinations (such as the system name and IP address), as well as the procedures to access these systems (such as database information, parameters to send files, and authentication information).

Creating a data source definition

Expand the data source definition in **III Navigator view** and double-click the data source definition ID that will be designed. Each data source definition item will be displayed in the **Editor**. The data source definition can now be created.

Point

The items displayed in the **Editor** will vary according to the data source type. Select the data source type when creating a new data source definition. Refer to "3.3.1 Creating a New Definition" for more information.

- When storing data in the delivery destination database:

  Enter the parameters relating to the delivery destination database.

- When delivering data to the delivery destination using 'send files':

  Enter the parameters required for the delivery destination information and to 'send files'.

- When storing data in a Salesforce delivery destination:

  Enter the parameters required for the collection source and for object storage.

- When storing data in an Azure Storage delivery destination:

  Enter the parameters required for the delivery destination and for BLOB delivery.

If the only processing required for data output at the previous Process Function process is to store it in the DataBox, enter parameters only in the 'common' portion of the data delivery definition.

Point

When a data source definition has already been registered in the server, specify the definition ID of the registered data source definition in the 'data source ID' of the data delivery definition to use as a reference. There is no need to recreate the same data source definition.

3.7.2 Designing the Delivery Data

Creating DataBox names

Specify the name of the DataBox storing the data to be delivered. DataBoxes are necessary to deliver data using the Information Integrator Server. Each data delivery definition item can be displayed in **Property view** by selecting the data delivery definition to be created in the **Editor**. Click **Common,** followed by **DataBox,** and enter the DataBox name from amongst the items displayed.

A unique value must be set for the DataBox name in the Process Flow. Data can be delivered between multiple processes by setting the same DataBox name in multiple processes.

Designing the data structure and format of delivery data

Design the data structure and format of the data to be delivered. Design the data structure definition and the format definition, respectively.

Point

It will not be necessary to specify the 'data structure definition ID' or the 'Format ID' if the conditions listed below are satisfied.

- When delivering data other than database data or object data
- When the collected data is sent directly to the delivery Process Flow without any conversion processing being performed

It is not necessary to specify the 'format ID' in the following cases:

- When 'store object' is used for data delivery
- When BLOB delivery is used for data delivery

3.7.3 Designing the Delivery Method

The delivery method is designed using the data delivery definition. The design procedure is explained below.

Creating a delivery definition

Expand the Process Flow in **III Navigator view** and double-click the Process Flow definition ID to be designed. The processing procedures for the Process Flow will be displayed in the **Editor**. Each delivery definition item can be displayed in **Property view** by selecting the data delivery definition to be created in the **Editor**. The data delivery definition can now be created.

Specifying data source definition references

Specification of data source definition references is as follows:

1. In **Property view** of the data delivery definition, click **Delivery definition** followed by **the data source list** and then click **Add**.

   This action will display a window, from which the definition ID of the data source definition to be referenced can be specified, and from which the delivery destination type can be selected.

Point

Delivery destination types

Configure the data format of the delivery destination. The setting values are as follows:

- Database
- File transfer
- Salesforce.com
- Azure Storage(BLOB)

2. Enter the definition ID and type.

   Click **Complete** to set the reference settings for the data source definition.

Setting the delivery data

Specify the definition IDs listed below in each data delivery definition.

- The 'DataBox' in which the delivery data is stored
- The 'data structure definition ID' of the data structure definition of the delivery data
- The 'format ID' of the format definition of the delivery data

Point

It will not be necessary to specify the 'data structure definition ID' or the 'format ID' if the conditions listed below is satisfied.

- When delivering data other than database data or object data
- When the collected data is sent straight to the delivery Process Flow without any conversion processing being performed

It is not necessary to specify the 'format ID' in the following cases:

- When 'store object' is used for data delivery
- When BLOB delivery is used for data delivery

In **Property view** of the data delivery definition, click **Delivery definition** followed by **the data source list** and then select the data source definition ID for setting the required parameters. The **Data Source Details** will be displayed. Set each of the parameters required for delivery processing.

3.8 Designing the Structure and Format of Process Data

This section describes the design procedures for the structure and format of process data.

3.8.1 Outline of Process Data Design

Design the data structure and format for the data to be processed.

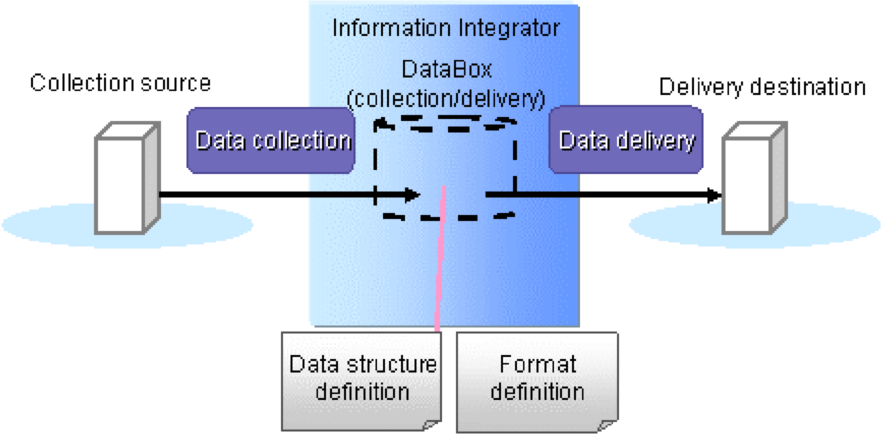
Data that requires design to be performed for the data structure and format will vary depending on the content of the Process Flow. Each Process Flow pattern is explained below.

Process Flows that do not contain data conversion processes

It is not necessary to design the process data for collection data that will be delivered without changes.

When collecting data from a database, or when delivering collected data with a modified data structure or format, design the data structure and format of the data to be delivered.

Figure 3.3 Example design that does not include data conversion processes

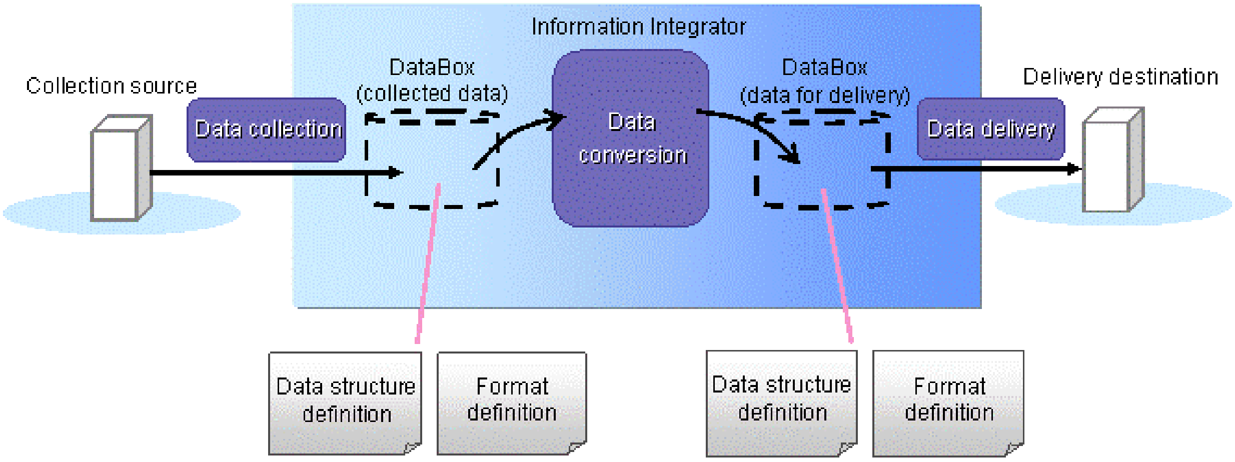


Process Flows that include only one data conversion process

Design the data structure and format of the data for collection and the data for delivery.

The data for collection will become pre-conversion data for data conversion, while the data for delivery will become post-conversion data for data conversion.

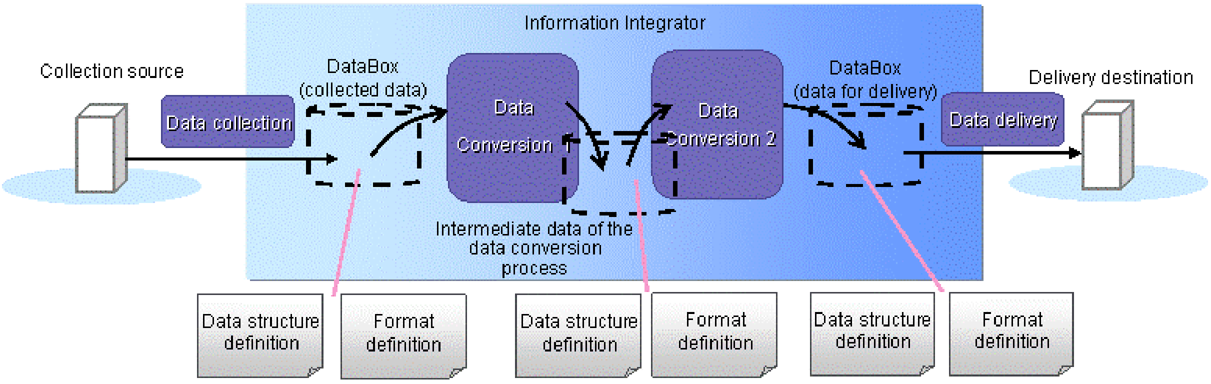
Figure 3.4 Example design that includes only one data conversion process



Process Flows that include multiple data conversion processes (and data filtering processes)

In addition to the data for collection and the data for delivery, design should also be performed for the intermediate results data of the data conversion.

Figure 3.5 Example of multiple data conversion processes (and data filtering processes)



3.8.2 Operation Procedure

The operation procedure is explained below.

3.8.3 Designing the Data Structure

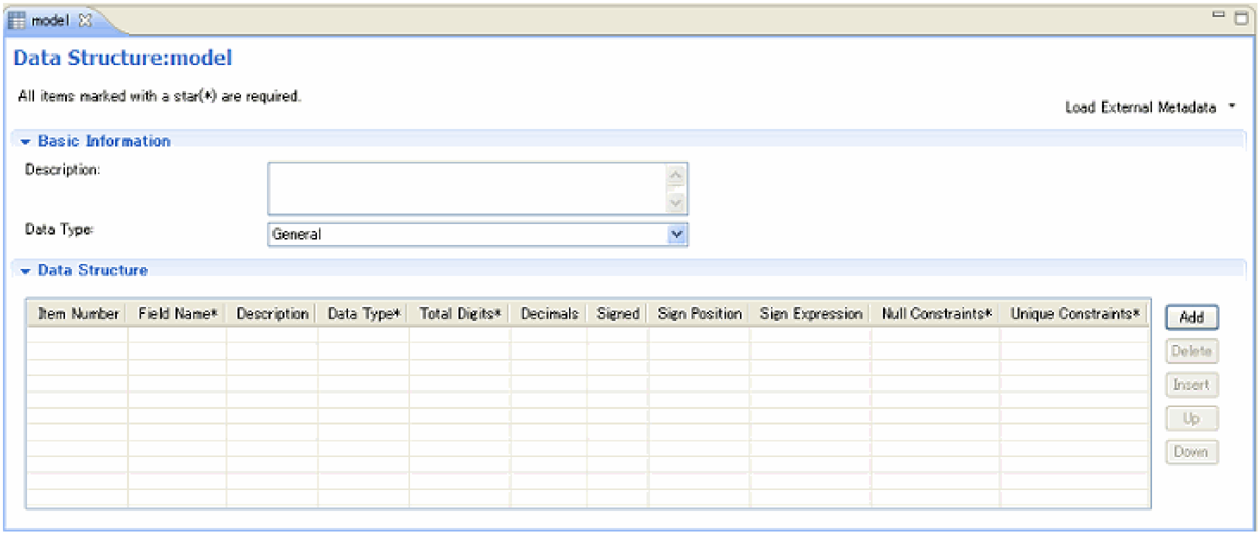
Data structure design involves use of the data structure definition to design the item filtering order and data attributes of data processed in Information Integrator data conversion.

1. Creating a data structure definition

   Expand the data structure definition in **III Navigator view** and double-click the data structure definition ID to be designed. The definition ID name and each data structure definition item will be displayed in the **Editor** tab. The data structure definition can now be created.

2. Designing the Data Structure

   Enter the information for each data structure definition item (such as item name, data attributes, number of columns) displayed in the **Editor**.



Data Structure:model

All items marked with a star(*) are required.

Basic Information

Description:

Data Type: General

Data Structure

Item Number	Field Name*	Description	Data Type*	Total Digits*	Decimals	Signed	Sign Position	Sign Expression	Null Constraints*	Unique Constraints*

Refer to "Design procedures using the metadata collection feature" mentioned later in this document for information on the procedures when using the metadata collection feature to import information from a remote system.

Notes when storing objects

The following notes should be adhered to when storing objects:

- When adding data

  A record management ID is automatically assigned by Salesforce when data is added.

  Accordingly, there is no need to assign Salesforce record management IDs to item names or data in the data structure definition. If record management IDs have been included in the data structure definition item names, the record management IDs defined in the data structure definition will be invalid.

- When updating data

  Assign Salesforce record management IDs to item names or data in the data structure definition.

  If the assigned record management ID matches the Salesforce record management ID, data for that record will be updated. Data will not be updated if the record management ID does not match. In this case, an error message will be output and the process will end in an error.

- When adding or updating data

  Assign a Salesforce external ID to item names and data in the data structure definition.

  If the assigned external ID matches the Salesforce external ID, data for that record will be updated. For records with non-matching external IDs, data will be added instead.

Note

Assigning record management IDs

It is not necessary to assign a record management ID when adding or updating data. If record management IDs have been included in the data structure definition item names, the record management IDs defined in the data structure definition will be invalid.

3.8.4 Procedures when Designing with the Metadata Collection Feature

This section explains the procedures for importing information from a remote system using the metadata collection feature.

CSV files

1. Select **Loading External Metadata** from the information for each data structure definition item displayed in the **Editor**, then select **CSV files**.

   The **Loading External Metadata (CSV)** window for entering the detailed items required for meta collection will be displayed.

2. Enter information in accordance with the selected item. Refer to the tooltip set in each III Studio item or to *III Studio Help* for information on the input content for each item.

   The meta information (item name and attribute information) collected will be displayed in the **Editor**. Confirm the displayed content.

COPY LIB files

The procedure for defining meta collection in files that specify definitions (COBOL COPY phrases) relating to COBOL data items is listed below.

1. Select **Loading External Metadata** from the information for each data structure definition item displayed in the **Editor**, then select **CSV files**.

   The **Loading External Metadata (COBOL COPY Clause)** window for entering the detailed items required for meta collection will be displayed.

2. Enter information in accordance with the selected item. Refer to the tooltip set in each III Studio item or to *III Studio Help* for information on the input content for each item.

   The meta information (item name and attribute information) collected will be displayed in the **Editor**. Confirm the displayed content.

Databases

1. Select **Loading External Metadata** from the information for each data structure definition item displayed in the **Editor**, then select **Database**.

   The **Loading External Metadata (Database)** window for entering the detailed items required for meta collection will be displayed.

2. Enter information in accordance with the selected item. Refer to the tooltip set in each III Studio item or to *III Studio Help* for information on the input content for each item.

   The meta information (item name and attribute information) collected will be displayed in the **Editor**.

   Confirm whether the attributes of the displayed content can be used in the Information Integrator Server.

Objects

1. Select **Loading External Metadata** from the information for each data structure definition item displayed in the **Editor**, then select **salesforce.com**.

   The **Loading External Metadata (salesforce.com)** window for entering the detailed items required for meta collection will be displayed.

2. Enter information in accordance with the selected item. Refer to the tooltip set in each III Studio item or to *III Studio Help* for information on the input content for each item.

   The meta information (item name and attribute information) collected will be displayed in the **Editor**. Confirm the displayed content.

3. Information on meta collection used in 'store object' will be displayed in the data structure definition regardless of whether an item has access permission for 'store object'. It will not be possible to perform 'store object' on any items that do not have access permission, so these items should be deleted. Attempts to 'store object' when an item that does not have access permission is still listed will result in an error.

   Note that the items listed below do not have access permission for 'store object'. Ensure these are deleted.

   - IsDeleted
   - CreatedDate
   - LastModifiedDate
   - SystemModstamp

Point

- When operation results or constants are output as data items after conversion processes have been performed, the data structure for post-conversion data should be designed to include these output items.
- Similar data structure definitions can be created easily by copying the created data structure definition and modifying the Data structure ID to divert it.

3.8.5 Designing the Data Format

When designing the data format, use the format definition to define attributes such as the data format and character code.

1. Creating a format definition

   Expand the format definition in **III Navigator view** and double-click the format definition ID that will be designed. The definition ID name will be displayed in the **Editor** tab, and each format definition item will be displayed. The format definition can now be created.

2. Designing the Data Format

   Enter the information (such as character code, record type and delimiter) for each format definition item displayed in the **Editor**.

Notes for Handling Data with Title Lines

The following notes should be adhered to when handling data with title lines in the data conversion or data filtering. Refer to "3.5.3 Notes for Handling Data with Title Lines" for more information.

3.9 Designing the Data Conversion Process

This section describes the design procedures for data conversion.

The two kinds of conversion processes that can be designed are listed below.

- Designing data conversion processes

  Design a conversion process to execute 'edit item' and 'record editing' on one or multiple input data, and convert into single data. Design using the data conversion definition.

  Data conversion is designed as follows:

  - Design the data structure and format for both pre-conversion and post-conversion data

    The conversion processes are designed by designing the data structure and format for pre-conversion and post-conversion data, respectively.

    Design the data structure and format using the data structure definition and format definition, respectively, and design by specifying each definition ID using the data conversion definition.

    Information Integrator will perform the required conversion processes for pre-conversion and post-conversion with differing data structures or formats. For example, the conversion process from Shift-JIS to UTF8 can be designed by defining 'SJMS' in the 'character code' of the pre-conversion format definition, and 'UTF8' in the 'character code' of the post-conversion format definition.

  - Designing the processes for 'edit item' and 'edit record'

    Design the conversion processes for 'arithmetic expressions', 'combine', 'sort' or 'aggregate' by entering parameters in each field of the data conversion definition. It is also possible to combine multiple data conversion processes in single data conversion definition.

- Data filtering

  Design the process to split single data into multiple data according to the record conditions, and to deliver the same data to multiple systems. Design using the data filter definition.

Point

The data conversion sequence

Format conversions and character code conversions, which affect the format of all files, should be executed at the beginning of data conversion. The 'edit item' and 'edit record' processes should be executed next. 'Aggregate' and 'sort' are executed at the end of data conversion.

Create multiple data conversion definitions (execute multiple conversion processes) when it is not possible to define all conversion processes at once, or when converting multiple collection data into single data after performing each separate conversion process.

This will decide the processing sequence of the multiple conversion Process Functions that were created when the Process Flow processing procedure was designed.

Point

Conversion patterns requiring multiple data conversion definitions

When designing conversion processes such as those indicated below, multiple data conversion definitions will be required.

- When executing 'edit item', 'concatenation' or 'extract record' after 'sort' and/or 'aggregate'.

Notes for handling data with title lines

Notes are provided that should be adhered to when handling data with title lines in data conversion or data filtering. Refer to "3.5.3 Notes for Handling Data with Title Lines" for more information.

3.9.1 Designing Data Conversion Processes

Once the data structure and format design is complete for pre-conversion and post-conversion data, create a data conversion definition and design the data conversion process.

It is necessary to complete the design of data structures for pre-conversion and post-conversion data using the data structure definition because the data conversion design needs to read a data structure definition that has been designed in advance.

3.9.1.1 Creating Data Conversion Definitions

Expand the Process Flow in **III Navigator view** and double-click the Process Flow definition ID to be designed. The processing procedures for the Process Flow will be displayed in the **Editor**. Each conversion definition item can be displayed in **Property view** by selecting the data conversion definition to be created in the **Editor**. The data conversion definition can now be created.

3.9.1.2 Reading Data Structures

Creating Data Structures

Create data structures for the pre-conversion data (the data that will be input in the conversion process) and the post-conversion data (the data that will be output in the conversion process), respectively. For two or more pre-conversion data, add a data structure using the procedure outlined below.

1. Select **Data conversion definition** followed by **Pre-conversion data** and **Data List** and then click **Add**.
2. One data structure will be added to the **Data List**.
3. Repeat steps 1 and 2 and add as many data structures for the input data as required.

Reading data structure definitions

Read the data structure definition that was created in advance for the newly-created input data as follows:

1. Click **Data conversion definition** followed by **Pre-conversion Data** and then **Data List**. Select the input data for which settings will be performed.

   A window to specify the details of the input data will be displayed in **Data Details**.

2. Click **Data Details** followed by **Data structure ID** and specify the **Data structure ID** for the data structure definition to be read.
3. Click the **Read Data structure** button.
   - When the data structure definition that corresponds to the specified **Data structure ID** is in the same project as the data conversion definition, read and display the following from the contents of the data structure definition:
     - 'Item name'
     - 'Data type'
     - 'Total digits'
     - 'Decimals'
   - When the data structure definition that corresponds to the specified 'data structure ID' is not in the same project as the data conversion definition, the following message will be displayed:

     ```
     The Data structure you specified doesn't exist in the project.
     Do you want to obtain from server?
     ```

     Click the **OK** button to read the data structure definition registered in the Information Integrator Server.

(When the server is not connected, use the **Connect Server** dialog box that displays to connect to the Information Integrator Server).

Implement the same operation for post-conversion data.

Note

- It is not possible to perform editing operations on a field that has read a data structure. To modify a data structure, revise the data structure definition and register in the Information Integrator Server before re-reading the data structure.
- Item mapping and arithmetic expressions are deleted when a data structure is re-read, so perform these settings again.

3.9.1.3 Data Format Settings

Define the formats for the pre-conversion and post-conversion data, respectively. Set the 'format ID' format definition that has been pre-designed in 'format ID' of **Data Details**.

3.9.1.4 Design Procedures for Item Mapping

Mapping is performed for each item in the pre-conversion and post-conversion data.

Item numbers are set automatically for each item in the pre-conversion and post-conversion data when a data structure is read.

Mapping for pre-conversion and post-conversion items is defined by entering a pre-conversion data item number in 'Item number (from)' of the post-conversion data setting area, and by arranging 'Item number (from)' and 'Item number (to)' and setting them.

3.9.1.5 Design Procedures for Edit Item Processes

Specify the arithmetic expression for each item.

Specify the operation type and argument in the arithmetic expression field of the item to be edited.

Point

Operations using pre-conversion data items are possible when the data structure definition data format is 'General' and the arithmetic expression operation type in the post-conversion data settings uses 'four arithmetic operations' and 'string concatenation'. The '@n' label is used for operations that use pre-conversion data items.

Specify a label ('@n', where 'n' is an ascending number from one to 1000) in the item to be used in the arithmetic expression, as well as the argument in the arithmetic expression. It is also possible to specify the arithmetic results in the output data without editing the input data, by specifying a return information number ('\n', where 'n' is an ascending number from one to 1000).

Note

Calculations can be performed for horizontal concatenation (INNER JOIN/OUTER JOIN) using each pre-conversion data item. Vertical concatenation (UNION/UNION ALL) cannot be specified.

3.9.1.6 Design Procedures for Sort Processes

Specify the item that will become the sort key for the item output as post-conversion data.

Specify the combination of sort type (ascending or descending) and the sort key priority.

3.9.1.7 Design Procedures for Aggregation Processes

Specify the item that will be the aggregation target and the items that will be grouped at the time of aggregation for the items output as post-conversion data. Specify 'Output/Don't output' for the total grouped item units.

3.9.1.8 Design Procedure for Advanced Settings of Conversion Processes

This section describes the detailed settings relating to data conversion.

The settings listed below can be performed as advanced operational options at the time of executing data conversion.

[Advanced settings example]

- Operation when a conversion process error occurs (continue process, suspend process)
- Alternate characters when a character code conversion error occurs

Setup by entering the parameters in the settings field of the conversion operation definition.

3.9.1.9 Design Procedures for Merging Processes

Setup the merging type, the merging key and the merging conditions, respectively, in the merging specification fields, in accordance with the procedure outlined below.

1. Specify the merging process type in the 'merging type'.
2. A merging specification line should be added, excepting when the merging process type is 'UNION ALL'.

   Click the **Add** button on the table immediately below 'merging type'.
3. Specify the merging key and the merging conditions for the added merging specification line.

   - INNER JOIN/OUTER JOIN

     Specify a merging key for each of the two input data to be merged.

     When performing multiple merges, specify an additional merging specification line(s).

   - UNION

     The item that will be the target whenever determining duplicate lines should be specified in the 'Merging key'.

     Specify the (indata1) item of the first input data in the merging key. (Example: 1-1, 1-2)

     A maximum of eight merging keys can be specified. However, the number of items specified can be as low as -1 when there are eight or fewer input data items.

   - UNION ALL

     As there is no requirement to specify a merging key or merging conditions, the merging specification row is also not specified.

3.9.2 Designing Data Filtering

The process to split data into multiple data in record conditions is designed as the data filtering process. Data filtering is designed in the data filter definition.

3.9.2.1 Creating a Data Filter Definition

Expand the Process Flow in **III Navigator view** and double-click the Process Flow definition ID to be designed. The processing procedures for the Process Flow will be displayed in the **Editor**. Each data filter definition item can be displayed in **Property view** by selecting the data filter definition to be created in the **Editor**. The data filter definition can now be created.

3.9.2.2 Designing the Data Filtering Process

Add as many filter conditions as required in **Filter Settings** in **Property view**, and specify the filter condition(s).

A **Filter data ID** will be automatically assigned to the filtered data, so design the relationship for the delivery of data between Process Functions, both before and after data filtering, when designing the processing sequence of the Process Flow.

Point

- It is possible to specify 'other' in the 'filter conditions'. When 'other' is specified in the 'filter conditions', only data that is not affiliated with any of the previously specified filter conditions will be filtered.

- Up to five filter conditions may include 'other'.

Note

- It is not possible to select multiple data filtering processes as preceding processes.
- Be aware of the following points when specifying 'other' in the 'filter conditions':
    - At least one or more filter conditions must be set when 'other' is specified.
    - Specify 'other' at the end of the 'filter conditions'.

3.10 Designing the Plug-in Process

This section describes the plug-in process design procedure using III Studio.

3.10.1 Design Procedure for the Process Plug-in Feature

The plug-in process (calling an external program) is designed in accordance with the procedure outlined below.

1. Creating a plug-in definition

   Expand the **Plug-in definition** in **III Navigator view** and double-click the plug-in definition ID name to be created. The definition ID name will be displayed in the **Editor** tab, and each plug-in definition item will be displayed.

   Point

   The items displayed in the Editor will vary in accordance with the plug-in type. Select the plug-in type when creating a new plug-in definition. Refer to "3.3.1 Creating a New Definition" for more information.

2. Adding a plug-in process to the Process Flow procedure

   Add a plug-in process when designing the processing order of the Process Flow.

Note

When specifying the plug-in process as the first process, ensure that the corresponding data collection process is specified as the subsequent process. Additionally, if there are multiple subsequent data collection processes present in this event, prepare the same number of plug-in processes as there are data collection processes.

3.10.2 Design Procedure for the Conversion Plug-in Feature

The plug-in process (calling an external program) is designed in accordance with the procedure outlined below.

1. Creating a plug-in definition

   Expand the **Plug-in definition** in **III Navigator view** and double-click the plug-in definition ID name to be created. The definition ID name will be displayed in the **Editor** tab, and each plug-in definition item will be displayed.

   Point

   The items displayed in the **Editor** will vary in accordance with the plug-in type. Select the plug-in type when creating a new plug-in definition. Refer to "3.3.1 Creating a New Definition" for more information.

2. Adding a plug-in process to the conversion definition process

   Add a plug-in process to the processing procedure for each conversion definition item.

Refer to *III Studio Help* for the input content of each item.

3.10.3 Compressing or Decompressing Data

When collecting compressed files from a remote system or when collecting BLOB data, this collected data will need to be decompressed. Similarly, when delivering files to a remote system or when delivering BLOB data, the data to be delivered will need to be compressed.

The design procedures for plug-ins that compress or decompress data are the same as those for the process plug-in feature. Refer to "3.10.1 Design Procedure for the Process Plug-in Feature" for more information.

When compressing or decompressing data, specify either the data compression command or the data decompression command at the time of creating the plug-in definition. The sub-sections below explain the commands.

Refer to *III Studio Help* for information on how to input commands in the plug-in definition.

3.10.3.1 ificompcmd (The Data Compression Command)

Description

Compress the data in zip format.

Command directory

- Windows

  The ificompcmd.cmd command is located in the directory below.

  ```
  InformationIntegratorServerInstallDirectory\bin
  ```

- UNIX

  The ificompcmd.sh command is located in the directory below.

  ```
  /opt/FJSVifisv/bin/
  ```

Command formats

- Windows

```
ificompcmd.cmd inputFilename outputFilename
```

- UNIX

```
ificompcmd.sh inputFilename outputFilename
```

Operands

Input file name

Specify the file name of the input data (pre-compression data) as an absolute path of 255 bytes or less.

Output file name

Specify the file name of the output data (post-compression data) as an absolute path of 255 bytes or less. If the file already exists, overwrite it.

Point

Specifying the plug-in definition

- Specify the command in the plug-in path of the executable plug-in as an absolute path. Specify '$IFI_InputFile' (the Process Function data of the previous process) in input parameter 1, and '$IFI_OutputFile' (the Process Function data of the subsequent process) in input parameter 2.

- The newly-created plug-in definition will be specified in between the data collection Process Function and the data delivery Process Function of the process definition.

Return values

0: Ends normally

8: Ends in an error

12: Ends in an error

3.10.3.2 ifiuncompcmd (The Data Decompression Command)

Description

Decompress data that has been compressed in zip format.

Command directory

- Windows

  The ifiuncompcmd.cmd command is located in the directory below.

  InformationIntegratorServerInstallDirectory\bin

- UNIX

  The ifiuncompcmd.sh command is located in the directory below.

  /opt/FJSVifisv/bin/

Command formats

- In Windows

```
ifiuncompcmd.cmd inputFilename outputFilename
```

- In UNIX

```
ifiuncompcmd.sh inputFilename outputFilename
```

Operands

Input file name

Specify the file name of the input data (pre-decompression data) as an absolute path of 255 bytes or less.

Output file name

Specify the file name of the output data (post-decompression data) as an absolute path of 255 bytes or less. If the file already exists, overwrite it.

Point

Specifying the plug-in definition

- Specify the command in the plug-in path of the executable plug-in as an absolute path. Specify '$IFI_InputFile' (the Process Function data of the previous process) in input parameter 1, and '$IFI_OutputFile' (the Process Function data of the subsequent process) in input parameter 2.

- The newly-created plug-in definition will be specified in between the data collection Process Function and the data delivery Process Function of the process definition.

Return values

0: Ends normally

8: Ends in an error

12: Ends in an error

3.10.4 Integrating with III Agent Process Flows (Optional Feature)

The process plug-in feature is used for integration with each III Agent Process Flow. Refer to "3.10.1 Design Procedure for the Process Plug-in Feature" for information. Specify the commands for integration when using the process plug-in feature to integrate with each III Agent Process Flow. The sub-sections below explain the commands.

Refer to *III Studio Help* for information on how to input commands in the plug-in definition.

3.10.4.1 ificlexeprc (Command for Integration with the Agent Process Flow)

Description

Execute the III Agent Process Flow.

Command directory

- Windows

  The ificlexeprc.bat command is located in the directory below.

  ```
  InformationIntegratorServerInstallDirectory\bin
  ```

- UNIX

  The ificlexeprc.sh command is located in the directory below.

  ```
  /opt/FJSVifisv/bin/
  ```

Command formats

- Windows

```
ificlexeprc.bat IIIAgentSystemToBeIntegrated IIIAgentProcessFlowID
```

- UNIX

```
ificlexeprc.sh IIIAgentSystemToBeIntegrated IIIAgentProcessFlowID
```

Operands

IIIAgentSystemToBeIntegrated

Specify the system name using up to 31 alphanumeric characters.

IIIAgentProcessFlowID

Specify the Process Flow ID located in the III Agent to be integrated in up to 31 alphanumeric characters.

Point

Specifying the plug-in definition

Specify the command in the plug-in path of the executable plug-in as an absolute path. Specify the name of the III Agent system to be integrated in input parameter 1, and the III Agent Process Flow ID in input parameter 2.

Return values

0: Ends normally

8: Error occurs when integrated

12: Input parameter error

14: Input parameter length error

Appendix A Function list that can be various use it

A.1 Function list of each communication protocol that can be used

Here, the function of each data base of each communication protocol that to be extracted and to be stored that can be used is shown by the list.

The following tables show the function of each communication protocol used by the file sending and receiving and the job initiation that can be used.

Table A.1 Function list of each communication protocol that can be used

Function Oita kind	Small function classification	Communication protocol								
		FTP+	FTP	FTPM	HTTP	HTTPS	HICS(FNA)	SAN	HULFT	HISPD
File reception	File reception (bureau initiation)	Y	Y	Y	Y	Y	Y	Y	Y	Y
	File reception (another bureau initiation)	Y	Y	Y	Y	Y	Y	Y	Y	Y
File transmission	File transmission (bureau initiation)-broadcast delivery	Y	Y	Y	Y	Y	Y	Y	Y	Y
	File transmission (bureau initiation)-batch distribution	Y	Y	Y	Y	Y	Y	Y	Y	Y
	Individual file transmission (bureau initiation)-distribution	Y	Y	Y	Y	Y	Y	Y	Y	Y
	File transmission (another bureau initiation)	Y	Y	Y	Y	Y	Y	Y	Y	Y
Job initiation	Other party side job start	Y	N	Y	Y	Y	Y	N	N	N
	Job result notification	Y	N	N	Y	Y	N	N	N	N
Optional feature	Compression and restoration	Y	N	N	N	N	Y	N	N	N
	Transfer rate control	Y	Y	Y	Y	Y	Y	N	N	Y
	File replacement	Y	Y	Y	Y	Y	Y	Y	Y (Note)	Y
	File addition writing	Y	Y	Y	N	N	Y	Y	Y (Note)	N
	Binary forwarding mode	Y	Y	Y	Y	Y	Y	Y	Y (Note)	Y
	Text forwarding mode	Y	Y	Y	N	N	Y	Y	Y (Note)	N
Line control	Multiplicity control	Y	Y	Y	Y	Y	Y	Y	Y (Note)	N
	Connection continuance	Y	N	N	N	N	Y	N	N	N
High reliability	Transmittal confirmation	Y	N	N	N	N	N	Y	N	N

Function Oita kind	Small function classification	Communication protocol								
		FTP+	FTP	FTPM	HTTP	HTTPS	HICS(FNA)	SAN	HULFT	HISPD
Environment	IPv6 address	Y	Y	Y	Y	Y	-	-	Y (Note)	Y
	IPv4 address	Y	Y	Y	Y	Y	-	-	Y	Y

Y: It is possible to use it.

N: It is not possible to use it.

-: Off the subject

Note: It is possible to use it by setting the HULFT side.

Note

- HICS(FNA) and SAN cannot be used with Linux (32 bit system requirements).
- HICS(FNA) cannot be used with Linux (64 bit system requirements).
- Both the IPv6 address and the IPv4 address should be able to be used on the system.
- When HISPD is used for the communication protocol in the IPv6 address environment, it is necessary JDK(Java Development Kit)/ JRE(Java Runtime Environment) 7.0 or later bundled to Interstage Application Server.

A.2 Function list of each data base that can be used

The following tables show the function that can be used by the DB extraction and storing each data base.

DB extraction

Table A.2 Function list of each data base that can be used (At the DB extraction).

	Platform	Data base type	Function	
			Extraction of all	Condition extraction
Extraction from data base of system	Windows	Symfoware Server	B	O
		Oracle	O	O
		SQL Server	B	O
		DB2	O	O
		Other data bases (ODBC connection)	O	O
	Solaris	Symfoware Server	B	O
		Oracle	O	O
		DB2	O	O
		Other data bases (ODBC connection)	O	O
	Linux	Symfoware Server	B	O
		Oracle	O	O
		DB2	O	O
		PostgreSQL	O	O

	Platform	Data base type	Function	
			Extraction of all	Condition extraction
		Other data bases (ODBC connection)	O	O
Extraction from data base of other party side system	Windows	Symfoware Server	O	O
		Oracle	O	O
		SQL Server	O	O
		DB2	O	O
		Other data bases (ODBC connection)	O	O
	Solaris	Symfoware Server	O	O
		Oracle	O	O
		DB2	O	O
		Other data bases (ODBC connection)	O	O
	Linux	Symfoware Server	O	O
		Oracle	O	O
		DB2	O	O
		PostgreSQL	O	O
		Other data bases (ODBC connection)	O	O

B: Unloader or dynamic SQL can be selected as an extraction method. (Even when the unloader is selected, it is likely to become automatic, dynamic SQL.)

O: The extraction method becomes dynamic SQL.

N: It is not possible to use it.

DB storage

Table A.3 Function list of each data base that can be used (At the DB storage).

	Platform	Data base type	Function			
			Data substitution	Data addition	Data update	Data addition/ update
It stores it in the data base of the system.	Windows	Symfoware Server	B	O	O	O
		Oracle	B	B	O	O
		SQL Server	B	B	O	O
		DB2	O	O	O	O
		Other data bases (ODBC connection)	O	O	O	O
	Solaris	Symfoware Server	B	O	O	O
		Oracle	B	B	O	O
		DB2	O	O	O	O
		Other data bases (ODBC connection)	O	O	O	O
	Linux	Symfoware Server	B	O	O	O

	Platform	Data base type	Function			
			Data substitution	Data addition	Data update	Data addition/ update
		Oracle	B	B	O	O
		DB2	O	O	O	O
		PostgreSQL	O	O	O	O
		Other data bases (ODBC connection)	O	O	O	O
It stores it in the data base of the other party side system.	Windows	Symfoware Server	O	O	O	O
		Oracle	B	B	O	O
		SQL Server	O	O	O	O
		DB2	O	O	O	O
		Other data bases (ODBC connection)	O	O	O	O
	Solaris	Symfoware Server	O	O	O	O
		Oracle	O	O	O	O
		DB2	B	B	O	O
		Other data bases (ODBC connection)	O	O	O	O
	Linux	Symfoware Server	O	O	O	O
		Oracle	O	O	O	O
		DB2	B	B	O	O
		PostgreSQL	O	O	O	O
		Other data bases (ODBC connection)	O	O	O	O

B :Loader or dynamic SQL can be selected as a storage method.

O :The storage method becomes dynamic SQL.

A.3 Function list of each object that can be used

The following tables show the function that can be used by extracting and storing each object.

Object extraction

Table A.4 Function list of each object that can be used (At the object extraction).

	Object type	Function	
		Extraction of all	Condition extraction
Other party side system Object extraction	Salesforce	Y	Y

Y: It is possible to extract it.

N: It is not possible to use it.

Object storage

Table A.5 Function list of each object that can be used (At the object storage).

	Object type	Function			
		Data substitution	Data addition	Data update	Data addition/ update
It stores it in the object of the other party side system.	Salesforce	N	Y	Y	Y

Y : It is possible to use it.

N: It is not possible to use it.