



iWay Integration Tools Designer User's Guide

Version 7.0.x and Higher

Active Technologies, EDA, EDA/SQL, FIDEL, FOCUS, Information Builders, the Information Builders logo, iWay, iWay Software, Parlay, PC/FOCUS, RStat, Table Talk, Web390, WebFOCUS, WebFOCUS Active Technologies, and WebFOCUS Magnify are registered trademarks, and DataMigrator and Hyperstage are trademarks of Information Builders, Inc.

Adobe, the Adobe logo, Acrobat, Adobe Reader, Flash, Adobe Flash Builder, Flex, and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2018, by Information Builders, Inc. and iWay Software. All rights reserved. Patent Pending. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Contents

Preface	
Documentation Conventions	
Related Publications	
Customer Support	
Help Us to Serve You Better	9
User Feedback	
Information Builders Consulting and Training	
1. Concepts	13
Introducing iIT Designer	13
About Designer	
Understanding the Process System.	
Integration Projects	
Process Flows.	
The iIT Designer Environment	15
Starting Designer within iWay Integration Tools	
Navigating the User Interface	
The Designer Window	
Toolbar lcons	
Process Flow Toolbar	20
The Palette Toolbar	
Palette Toolbar Selection	
Basic Objects Toolbar	
Control Objects Toolbar	22
Execution Objects Toolbar	
Defining Process Flow Objects	
Start Object	
End Object	
The Control Objects	
Decision Switch Object	
Decision Test Object	
Join Object	
Junction Object.	

Set Object	
Handling Header Information.	
Synch Object.	45
The Execution Objects	
BI Object	
Email Object	
Emitter Object	
ETL Object	61
File Object	68
FTP Object	
HTTP Object	74
Iterator Object	76
Process Object	
Queue Object	83
ReportCaster Object	
Script Object	
Service Object	
SQL Object	
Data Integration Object	
Transform Object	
Web Service Object	
2. Tasks	
Creating a Project	
Working with a Project	
Opening and Closing a Project.	
Refreshing a Project.	
Viewing Project Properties.	122
Importing a Process Flow to a Project	
Managing Process Flows	
Creating a Process Flow	
Adding Objects to a Process Flow	130
Building Relationships Between Objects	

	Saving a Process Flow	134
	Validating a Process Flow	
	Compiling a Process Flow	
	Testing a Process Flow	
	Publishing a Process Flow	
	Exporting a Process Flow	145
	Adding an Adapter to iIT Designer	145
	Creating a new iWay Resource	146
	Selecting the Adapter	
	Connecting to an Enterprise Information System	
	About Targets	
	Creating a Registry	
	Exporting the Adapter Target	159
	Using Variables and XPath	
	Using XPath in Designer	
	The XPath Specification	
	Building XPath Statements	164
	Adding an Object to a Process Flow from an iSM Registry	
	Working with Customizations	169
	Adding a Customization JAR	
	Adding Additional Services and Components.	
3. '	Tutorial	
	An Example Process Flow	
	About This Example	
	Create the Input XML Documents	
	Creating the Process Flow	177
	Save the Process Flow	
	Validate and Test the Process Flow	
4.	Process Flow Object Properties	
	Iterator Object Properties	217
	SQL Object Properties	

Contents

This document describes how to use iWay Integration Tools (iIT) Designer.

Note: This Release 7.0.x content is currently being updated to support iWay Release 8.0.x software. In the meantime, it can serve as a reference for your use of iWay Release 8. If you have any questions, please contact *Customer_Success@ibi.com*.

How This Manual Is Organized

This manual includes the following chapters:

	Chapter/Appendix	Contents
1	Concepts	Describes key concepts related to iWay Integration Tools (iIT) Designer.
2	Tasks	Describes specific tasks related to iWay Integration Tools (iIT) Designer.
3	Tutorial	Provides an example of a process flow that routes a document based on the document type.
4	Process Flow Object Properties	Provides detailed information about process flow objects that have extended properties.

Documentation Conventions

The following table describes the documentation conventions that are used in this manual.

Convention	Description
THIS TYPEFACE or this typeface	Denotes syntax that you must enter exactly as shown.
this typeface	Represents a placeholder (or variable), a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option that you can click or select.
underscore	Indicates a default setting.
Key + Key	Indicates keys that you must press simultaneously.
{}	Indicates two or three choices. Type one of them, not the braces.

Convention	Description
	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis ().
	Indicates that there are (or could be) intervening or additional commands.

Related Publications

Visit our Technical Documentation Library at *http://documentation.informationbuilders.com*. You can also contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have any questions about this product?

Join the Focal Point community. Focal Point is our online developer center and more than a message board. It is an interactive network of more than 3,000 developers from almost every profession and industry, collaborating on solutions and sharing tips and techniques. Access Focal Point at http://forums.informationbuilders.com/eve/forums.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our website, *http://www.informationbuilders.com*. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of *http://www.informationbuilders.com* also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

Call Information Builders Customer Support Services (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code number (*xxxx.xx*) when you call.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Help Us to Serve You Better

To help our consultants answer your questions effectively, be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following tables list the environment information our consultants require.

Platform	
Operating System	
OS Version	
JVM Vendor	
JVM Version	

The following table lists the deployment information our consultants require.

Adapter Deployment	For example, JCA, Business Services Provider, iWay Service Manager
Container	For example, WebSphere
Version	
Enterprise Information System (EIS) - if any	
EIS Release Level	
EIS Service Pack	
EIS Platform	

The following table lists iWay-related information needed by our consultants.

iWay Adapter	
--------------	--

iWay Release Level	
iWay Patch	

The following table lists additional questions to help us serve you better.

Request/Question	Error/Problem Details or Information
Did the problem arise through a service or event?	
Provide usage scenarios or summarize the application that produces the problem.	
When did the problem start?	
Can you reproduce this problem consistently?	
Describe the problem.	
Describe the steps to reproduce the problem.	
Specify the error message(s).	
Any change in the application environment: software configuration, EIS/database configuration, application, and so forth?	
Under what circumstance does the problem <i>not</i> occur?	

The following is a list of error/problem files that might be applicable.

□ Input documents (XML instance, XML schema, non-XML documents)

□ Transformation files

- Error screen shots
- Error output files
- Trace files
- Service Manager package to reproduce problem
- Custom functions and agents in use
- Diagnostic Zip
- Transaction log

For information on tracing, see the *iWay* Service Manager User's Guide.

User Feedback

In an effort to produce effective documentation, the Technical Content Management staff welcomes your opinions regarding this document. Please use the Reader Comments form at the end of this document to communicate your feedback to us or to suggest changes that will support improvements to our documentation. You can also contact us through our website, *http://documentation.informationbuilders.com/connections.asp*.

Thank you, in advance, for your comments.

Information Builders Consulting and Training

Interested in training? Information Builders Education Department offers a wide variety of training courses for this and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our website (*http://education.informationbuilders.com*) or call (800) 969-INFO to speak to an Education Representative.

Chapter

Concepts

This section describes key concepts related to iWay Integration Tools (iIT) Designer.

In this chapter:

- Introducing iIT Designer
- □ The iIT Designer Environment
- Defining Process Flow Objects

Introducing iIT Designer

About Designer

The capability of graphically visualizing a business process is a powerful and necessary component of any application integration offering iIT Designer, a Windows based design-time tool that provides a visual and user-friendly method of creating a business process, also called a process flow.

Through a process flow, you control the sequence in which tasks are performed and the destination of the output from each task in the process flow. In the Designer user interface, you can create a project that holds one or more process flows. Each process flow you create is a single unit of work made up of one or more process objects, such as an Email, Adapter, or Emitter, which are graphically represented by icons and connected by lines that establish a relationship between the objects.

iIT Designer provides a unified iWay toolset for designing and integrating process flows. Through Designer, you can perform the following functions:

- Build process flows.
- Access iWay Explorer to explore metadata.
- Expose web services.
- Launch iWay Transformer.

iIT Designer also offers concurrency control by checking process flows in and out of a repository. This prevents people from simultaneously working on a process flow.

When the design of a process flow is complete, you can use the validation option to confirm that the objects and their relationships meet the design criteria. You can also test the process flow by executing it with an actual input document to see results. When you are satisfied that the process flow is stable, you can publish it, making it available to iWay Service Manager. For details on compiling, validating, testing, and publishing a process flow, see *Managing Process Flows* on page 126

Understanding the Process System

A process system links together a logical set of component objects into an iWay process flow to create an application. These nodes are interconnected in an application by edges along which documents flow. An edge segment stretches from the output terminal of one node, to the input terminal of the next node. The following image shows a single process flow node with an input terminal, into which input documents enter the node, and two output terminals, from which the documents leave the node. The processing within the node determines from which terminal the document leaves the node.



Application flows are published to iWay Service Manager at run time and are stored in the iWay configuration repository.

Integration Projects

An Integration project is the top level organization of resources from which you build process flows in iWay Integration Tools. Integration Projects are used for version management and the sharing and organization of resources. Project resources include process flows, services, transforms, adapters, emitters, and schemas.

Projects can be associated with either a named run-time server configuration or a Registry. Your choice of project association should be determined by where you intend to publish (deploy) your process flow. During the publishing process, all available targets are displayed. It is preferable to have your development environment be the same as your run-time environment. For manipulating projects and developing processes from Designer, the iWay Registry functions much like a configuration. Unlike a configuration, the Registry is a design-time data store and it supports Service Manager's channel architecture. If you are developing a process for use as part of a channel, you must publish it to the Registry for subsequent deployment. For more information about deploying channels, see the *iWay Service Manager User's Guide, Version 7 SM*.

Integration Projects can contain a single process flow or multiple process flows. Process flows are designed within and for a specific project. Therefore, the project must exist before you create a process flow.

Process Flows

An Integration project holds the building blocks for the process flows that you will create under that project. These include services, transforms, adapters, emitters, and schemas.

iIT Designer also provides process flow objects, the executables within a process flow that direct and act upon the input to the process flow. The process flow is graphically represented in the workspace as a box and line diagram. The boxes are the objects you add, and the lines are the relationships you define between those objects. Designer includes executable objects. For example, a transform object converts a document from one format to another. Other objects control the flow of a document. For example, a Decision Test object routes a document based on a true or false condition. As you build a process flow, you will configure each object and the relationships between them to meet your requirements.

Once the process flow is created and saved, you can validate the structure and then test the process flow in a run-time environment. You can then make it available for use by publishing it to a run-time environment.

The iIT Designer Environment

Explains how to navigate in iWay Designer and describes the options and panes available in the interface.

Starting Designer within iWay Integration Tools

To start Designer, double click the *iit.exe* file.

The workspace wizard pane appears as shown in the image below.

🔏 Workspace Launcher	×
Select a workspace	
iWay Integration Tools stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: C:\Program Files (x86)\iIT\eclipse\workspace	Browse
Use this as the default and do not ask again	
	OK Cancel

Click the *Browse* button to select a path for your Workspace, or accept the default path. Click *OK*.

The main iWay Integration Tools window opens, as shown in the following image.



To utilize the functionality of iIT Designer, you must first create an Integration Project. See *Creating a Project* on page 119 for more information. Once the Project is created, a new process flow must be created, as described in *Managing Process Flows* on page 126.

When a new process flow is created, the IIT Designer window provides access to all of the tools you will need to build the process flow. A set of menus and toolbars appear to the right of the window. Beneath them are the Layout and Properties panes. The following sections provide details on the display panes, the menu options, and the toolbar selections.

Once you open a Project and click on a Process Flow, most menu options and toolbar icons become available. Some menu options are only available when certain components of iIT Designer are selected.

Navigating the User Interface

The Graphical User Interface (GUI) of iWay Integration Tools offers a variety of tools and options to assist you in creating projects and process flows. You can customize the GUI to fit your work needs. For example, you can drag toolbars and move, resize, minimize, or hide panes.

Like most applications, Designer provides several ways to perform a task and achieve the same result. Most tasks can be initiated from either a main menu option, a toolbar icon, or by right-clicking an item and selecting an option from the drop-down list. For example, to save a project you can do any one of the following:

- Choose Save from the File menu.
- Click the Save icon on the toolbar.
- Right-click a project folder, and select Save from the drop-down list.

The Designer Window

The Designer window provides the panes, illustrated in the following image, to construct your projects and display project-related information:



- 1. The **Navigator** pane appears on the left. It contains the following tabs: Integration Explorer, Library Manager, and iWay Explorer. This pane provides access to the Designer repositories that contain your projects and process flows, the servers to which you can connect, iWay Explorer, and the available web services.
- 2. The **Workspace** pane appears to the right of the Navigator pane. This is the area in which you build a process flow. The Workspace area contains two tab views, *Design* and *Test Results*. The *Design* view displays the Process Flow as you build it. The *Test Results* view displays test results of the last test run of a process.
- 3. The **Palette** pane appears to the right of the Workspace area when a new Process Flow is created. The Palette pane contains all of the Objects and Relationship elements required to construct a Process Flow.
- 4. The **Layout** pane appears below the Navigator pane. It displays a preview of the process flow as you create it.

5. The **Properties Window** pane appears at the bottom of the workspace. This pane displays the attributes of the selected object in a process. You can change any attribute of the selected object from the Properties Window.

Any pane can be closed or accessed by clicking *Window* in the toolbar and selecting *Show View*, as shown in the following image.

Window Help							
New Window New Editor		0	• • • *	월 ▼ 월 ▼ 🏷 🔶			
Open Perspective	×						
Show View	\mathbf{N}^{*}		Complex Properties				
Customize Perspective	45		Console	Alt+Shift+Q, C			
Save Perspective As		9	Error Log	Alt+Shift+Q, L			
Reset Perspective	•	À	Integration Explorer				
Close Perspective		U	iWay Explorer				
Close All Perspectives		≥ .	Library Manager Outline	Alt+Shift+O_O			
Navigation		• ! <u>/</u>	Problems	All of the good			
Preferences			Other	Alt+Shift+Q, Q			

Toolbar Icons

When you start iIT Designer for the first time, a toolbar appears just above the Navigator Pane and the Palette appears to the right of the Workspace. The Palette contains the following sections:

- Selection elements
- Relationship elements
- Basic objects
- Control objects
- Execution objects

Although the toolbar appears as a single unit, each category is contained between horizontal bars. You can drag the vertical bar to place a toolbar category on a separate line, or space it from the other toolbars within the toolbar area of the window.

Process Flow Toolbar

The Process Flow toolbar contains functions that apply to a Process Flow as a whole. Some functions are not available unless certain project components are selected in iIT Designer.

The icons that make up the Process Flow toolbar are shown below.



The following table describes each icon found on the Standard toolbar.

lcon	Description
5	Compile a process flow.
	Open the Publishing Process tool.
	Launch Property Manager
\Leftrightarrow	Undo or Redo the last action performed.
	Align left, center, or right.
	Align top, middle, or bottom.
100% -	Percentage magnification of Workspace area objects

The Palette Toolbar

The Palette toolbar provides the fundamental options for working with objects in a process flow.

Palette Toolbar Selection

The icons that make up the selection area of the Palette toolbar are shown below.



The following table lists and describes the icons found on the selection area of the Palette toolbar.

lcon	Description
Select	The Select option allows you to click to select a particular object or relationship.
[]] Marquee	The Marquee tool allows you to select an area in the workspace. Click inside the workspace and drag to select the area you want to work in.
•••• Relation	The Relation is used to draw a line that represents the relationship between two objects in a process flow.
⊷⊷ Loop	The Loop option is a special type of Relation (also called a line or edge) that works in conjunction with an Iterator object to repeat certain portions of a process flow over selected portions of one input document.

Basic Objects Toolbar

The Basic Objects toolbar provides fundamental options for working with objects in a process flow. The icons that make up the Basic objects toolbar are shown below.



The following table lists and describes the icons found on the Basic Objects toolbar. See *Defining Process Flow Objects* on page 25 for a detailed description of each object.

lcon	Description
	An End object terminates a process flow. At least one End object is required to terminate a process flow.
Α	Text allows you to include text in the process flow. For example, you can label the relationship line that appears between objects.

Control Objects Toolbar

The Control Objects toolbar provides fundamental options for working with objects in a process flow. The following image shows the icons that make up the Control Objects toolbar. See *Defining Process Flow Objects* on page 25 for a detailed description of each object.



lcon	Description
*	A Decision Switch object evaluates the contents of a document and routes it to the next object(s) based on the values in the document.
1	A Decision Test object evaluates the contents of a document and routes it to the next object(s) based on a true or false result.
	A Join object combines two or more documents into a single document.
	A Junction object combines two or more paths of execution into a single path.

lcon	Description
	A Set object initiates one or more process events that Synch objects are waiting on.
×	A Synch object waits for a condition to be met, such as the expiration of a time limit or the evaluation of an expression as 'true'.

Execution Objects Toolbar

The Execution Objects toolbar is shown in the following image.



The following table provides a brief description of the Execution objects. See *Defining Process Flow Objects* on page 25 for a detailed description of each object.

lcon	Description
B	A BI object allows the process flow to execute a WebFOCUS report.
<u>@</u> .	An Email object allows a process flow to send or receive an Email with one or more attachments.
	An Emitter object adds an address to the payload using one of the supported protocols.
ETL P	An ETL object executes a DataMigrator flow.
	A File Object allows a process flow to perform file operations at execution time, such as read and write.
FTP	An FTP object allows the process flow to send or receive data using FTP.
HTTP	An HTTP object allows a process flow to send or receive data using HTTP.
	An Iterator object is a service object that is used in conjunction with a Loop object to process repeating sections of an input document.
	A Process Lookup object inserts a published process flow into the current process flow as an object.
<u>ا</u> سا	A Queue object sends or receives documents using a supported queue type.
	A ReportCaster object allows a process flow to trigger one or more ReportCaster schedules, which then execute and distribute WebFOCUS reports or the contents of a URL or file.
	A Script object allows you to execute JavaScript on an object in a process flow.
Sec.	A Service object is a unit of work that emits default events and/or one or more custom events.

lcon	Description
SQL	An SQL object executes an SQL request to a JDBC data source.
	A Data Integration object iterates over an SQL result set and batch inserts.
	A Transform object applies a selected transform to the document it receives in the process flow.
	A Web Service object executes a web service.

Defining Process Flow Objects

Process flow objects are the major components of a process flow. IIT Designer provides control objects, which direct the movement of the input document, and execution objects, which perform a specific task with the document. You will configure each object as you add it to the process flow. This section describes each object provided in Designer and explains how to configure these objects to perform the tasks that you require.

Designer provides the following process flow objects to build a process flow:

Start Object	Process Object
End Object	Web Service Object
Join Object	SQL Object
Junction Object	ETL Object
Decision Switch Object	BI Object
Decision Test Object	ReportCaster Object
Set Object	Email Object
Sync Object	FTP Object
Debug Object	File Object

Service Object	HTTP Object
Iterator Object	Emitter Object
Transform Object	Queue Object

Start Object

All process flows must begin with the Start object. Therefore, it is automatically added to the workspace when you create a new process flow. A process flow can contain only one Start object.

The Start object icon is shown below.



To view the properties of a Start object in a process flow, right-click the Start icon in the workspace and select *Properties* from the drop-down list. The Start Object properties are displayed in the bottom of the Workspace.

Note: You can also change the Start object parameters from this dialog box.

The Properties window consists of the General, Schema, and Debug tabs. The General tab is expanded below. It displays the name and a description of the Start object. To view information about the Start object schema or debug setting, select this tab.

Properti	es 🛛	• (Complex Properties	📮 Console	🔛 Problems	1		
Start Obj	ject							
General	Only	one S	start Object can be	defined in any	y one process.			
Schema	Name	e						
Debug	Start							
	Description							
	Start Object						~	
							-	
	📃 Ta	р						

The Schema tab, shown in the following image, allows you to select and view a schema that is available from the current server configuration. This schema maps to the input document expected by the process flow.

🗆 Properti	ies 🖾 😑 Complex Prop	erties 📮 Console 💡	Problems		₹ - 4
Selection fro	om Test_flow				
Start Ob	ject				
General	Schemas available in th	e current server config	juration are s	hown below. To view an existing schema click on the 'View Sc	hema' button
Schema	Schema	Name of the schem	a which des	ribes the input document to this process flow	
Debug				 X	View
	Root Tag Element	Name of the root ta selected)	g element o	the input document to this process flow. (* Required if a Sche	:ma is
					•
	Description	Description for the	selected sche	ma	
					*
	•	k		m	•

Note: Schema is needed if you intend to expose the process flow as a web service.

If you want to designate a schema for the Start object, select a schema from the Schema dropdown list. To view the selected schema in a separate window, click *View*.

The Input Schema Value field reflects the schema selection made in the Schema tab. Select or type a value in the *Root Tag Element* field. An XML root tag is the beginning XML tag and it identifies the XML document. For example, <phonebook>.

End Object

The End object terminates the process flow. All processes must terminate with at least one End object. There can be multiple End objects in a single process flow. The End object icon is shown below.



When you add an End object to the flow, the New End Object dialog box opens, as shown in the following image.



Type a *Name* to identify the End object and, optionally, type a brief *Description*. The Design mode option allows you to include the object in the process flow as a place marker, that is, without defining its parameters. Click *Next* to continue configuring the object.

The End Object Schema dialog box opens, as shown in the following image.

🔬 New End Object				
End Object Schema Select a schema to be assigned to the End Object				
Schemas available in the current server configuration are shown below. To view an existing schema click on				
Schema	Name of the schema which describes the output document to this process flow			
	X View			
Root Tag Element	Name of the root tag element of the output document to this process flow. (* Required if a Schema is selected)			
Terminate	Select if this end object is the completion point			
Description	Description for the selected schema			
?	< Back Next > Finish Cancel			

Note: Schema is needed if you intend to expose the process flow as a web service.

If you want to expose the process flow as a web service, select a schema for the End object from the *Schema* drop-down list. Otherwise, select *None*. If you want to view a schema you selected, click *View*. The schema appears in the browser.

The End object properties are:

- **Root Tag Element.** Enter or select the name of the root tag element of the output document for the current process flow.
- □ **Terminate.** Select false to pass the document to the listener, or select true to suppress transmission of the document to the listener. If your process flow provides all of the output necessary for your application, then you can set Terminate to true in order to bypass the outlets associated with your channel. You can use multiple End objects with different settings in a process flow to direct document flow behavior.

When you have finished entering, or selecting, the property values, click *Finish*. The End object icon appears in the workspace.

The Control Objects

The Control objects for a Designer process flow are:

- Decision Switch object
- Decision Test object
- Join object
- Junction object
- Set object
- Synch object

The following sections describe how to define each Control object as you include it in a process flow.

Decision Switch Object

The Decision Switch object evaluates the contents of a document or a variable, and routes the document accordingly, similar to the switch/case construct of other programming languages. The main difference is that the document can match multiple cases simultaneously when using an XPath switch expression. The Decision Switch object icon is shown below.



When you add a Decision Switch object to a flow, the New Decision Switch Object dialog box opens, as shown in the following image.

New Decision Switch Object				
Name and Description				
Provide object name and description				
A Switch object evaluates the contents of the document and uses the result to determine further routing of the document.				
Name				
Decision Switch				
Description				
Decision Switch Object				
Sack Next > Finish Cancel				

Type a *Name* to identify the Decision Switch object and, optionally, type a brief *Description*. The Design mode option allows you to include the object in the process flow as a place marker, that is, without defining its parameters. Click *Next* to continue configuring the object.

The Switch Expression dialog box opens, as shown in the following image.

🔬 New Deci	sion Switch Object			
Switch Expression				
A switch obtains the value using a variable or function (XPATH(), LDAP(), FILE()) and routes the process down the path(s) equal to				
A switch obtains the value as specified and routes the process down the path(s) equal to the selected value(s). If multiple values are selected, the document goes once down each path for each selected value.				
Expression:	•			
?	< Back Next > Finish Cancel			

The Decision Switch object can direct the document down one or more alternate paths based on the value of the switch expression. In the following example, the document contains a value indicating the protocol(s) through which it should be routed (HTTP, EMAIL, FILE). To accomplish this, an XPath is constructed to capture the switch value from the document. You can also switch on computed conditions by using a variable that was set to one of your case values earlier in the process flow. For illustrative purposes, the following XML document has all switch values represented.

🔏 XPATH Builder				
Load a document instance				
Load a sample of the document that will be used in the flow in order to browse its schema and build an XPATH.				
This wizard will assist in creating a valid XPATH for use with an object. By creating an XPATH and running it againsts the				
sever you can see the results that will be returned to your object property.				
Load Import Reload				
Previously loaded files				
Clear history				
(0mm)				
<pre>(rition="1.0" /> </pre>				
- CROBINGS				
- <pmit_io>PMIT_IO></pmit_io>				
- <contentstre brown="" contents<="" for="" jumped<="" quick="" td=""></contentstre>				
-				
· · · · · · · · · · · · · · · · · · ·				
XML Source: /Test/Flows/switch.xml				
(Park Next) Finish Consel				
< back Next > Finish Cancel				

Click *Next* to build the XPath expression. The following image shows the expression in the XPath Builder.

🔏 XPATH Builder					
Build an XPath Browse the schema and find the elements you wish to use as input to the flow.					
Double click the element, att XML Data Type	ribute or attribute value you wi	ish to use to build the XPath stat XPath Operands			
e metadata e DOCUMENT	count(<node-set>) last() local-name(<node-set: name() namespace-uri(<node- too:ition()</node- </node-set: </node-set>	= != < > >= and			
? < B	ack Next >	Finish Cancel			

When you have finished constructing your XPath expression, click *Next* to continue to the Switch Cases dialog box.

Add the Switch Case values and descriptions to the table. Each switch case creates a named outlet leaving the Decision switch node. When the current switch value is equal to a switch case, the document will be sent down the corresponding edge for further processing. If the XPath expression recovers multiple switch values, the document will be sent down multiple edges simultaneously. See *Building Relationships Between Objects* on page 132 for details on drawing edges that connect to the named outlets of a Decision Switch Case.

Click Finish. The Decision Switch object icon appears in the workspace.



The following image shows the finished process flow containing the Decision Switch example.

Decision Test Object

The Decision Test object evaluates the contents of a document, returns a true or false condition, and routes the document to the next object(s) in the process based on that result. You will define the evaluation and routing criteria through the Decision Test object properties.

The Decision Test object icon is shown below.



When you add a Decision Test object to a flow, the New Test Object dialog box opens, as shown in the following image.

🔬 New Decision Test Object				
Name and Description				
Provide object name and description				
A Test object evaluates the contents of the document and uses the result to determine further routing of the document.				
Name				
Decision Test				
Description				
Decision Test Object				
? < Back Next > Finish Cancel				

Type a *Name* to identify the Decision Test object and, optionally, type a brief *Description*. The Design mode option allows you to include the object in the process flow as a place marker, that is, without defining its parameters. Click *Next* to continue configuring the object.
The Test Operands dialog box opens, as shown in the following image.

🔬 New Decision	Test Object				
Test Operands					
🔇 Provide Oper	rand One				
Test Paramete	rs				
Operands can	be literals, variables, or functions (XPATH(), LDAP(), or FILE())				
Operand One	-				
Operation	Equal To 🔹				
Operand Two	-				
?	< Back Next > Finish Cancel				

Operands in the Test Object can be literals, variables, or XPath expressions. The available comparison operations are: Is Null, Not Null, Equal To, Not Equal To, Greater Than, Less Than, Greater Than or Equal to, Less Than or Equal to.

Select the test parameters and click *Next*. The Value Occurrences dialog box opens, as shown in the following image.



When using an XPath expression, many values can be retrieved from the document. This dialog box allows you to specify how the test comparison should handle multiple values.

Select a test occurrence:

- **Unique.** Only one value can exist in the document.
- **Any.** At least one value must exist in the document.
- □ All. Every value must exist in the document.

Click Finish. The Decision Test object icon appears in the workspace.

The Decision Test object has two pre-defined custom outlets, 'true' and 'false', from which you can draw edges. See *Building Relationships Between Objects* on page 132 for details on drawing edges that connect to the named outlets of a Decision Test case.

Join Object

The Join object combines two or more documents into a single document. The single document then proceeds to the next object in the process flow. For example, if you have two XML documents:

A:

```
<A> ... </A>
and B:
<B> ... </B>
```

The Join object creates the following output:

Caution: Use caution when mixing Join and Junction objects in a process flow. When a Join object is located after a Junction object, results can be unpredictable.

The Join object icon is shown below.



When you add a Join object to the flow, the New Join Object dialog box opens, as shown in the following image.

New Join Object
Name and Description Provide object name and description
A Join object combines two or more documents into a single document which travels on a single thread of execution.
Name Join
Description
Join Object
Sack Next > Finish Cancel

Type a *Name* to identify the Join object and, optionally, type a brief *Description*. The Design mode option allows you to include the object in the process flow as a place marker, that is, without defining its parameters. Click *Next* to continue configuring the object.

The New Join Object - Timeout dialog box opens, as shown in the following image.

🔬 New Join Object					
Timeout					
Enter the timeout in seconds for this join object. The timeout value can be an xpath expression or an integer value.					
Timeout: 0					

Timeout is the amount of time (in seconds) that the Join object will wait to receive the documents it must join. The timeout count begins when the first document enters the Join object. If the Join receives all of the documents before the timeout period ends, the join process continues. If all documents needed to satisfy the join are not received, then a status document is generated and sent along the OnTimeout edge.

Select a Timeout value from the Value drop-down list and click Finish.

The Join object icon appears in the workspace.

Junction Object

A Junction object joins two or more paths of execution into a single path. It does not join documents that travel along the path. For example, if you have five distinct paths that run to a certain point, and then all paths execute the same logic, use the Junction object to consolidate the flow at the point of common execution.

Caution: Use caution when mixing Join and Junction objects in a process flow. When a Join object is located after a Junction object, results can be unpredictable.

The Junction object icon is shown below.



When you add a Junction object to a flow, the New Junction Object dialog box opens, as shown in the following image.

New Junction Object						
Name and Description Provide object name and description						
A Junction object combines two or more paths of execution into a single path. Documents traveling through the juction are not merged and continue to travel on their current thread of execution. Name						
Junction						
Junction Object						
Sack Next > Finish Cancel						

Type a *Name* to identify the Junction object and, optionally, type a brief *Description*. The Design mode option allows you to include the object in the process flow as a place marker. Click *Finish*. The Junction object icon appears in the workspace.

Note: There are no parameters associated with the Junction object.

Set Object

The Set object initiates one or more process events that Synch objects are waiting on. Set and Synch objects are used to suspend and resume the execution paths of a process flow.

The Set object icon is shown below.



When you add a Set object to a flow, the New Set Object dialog box opens, as shown in the following image.

🖌 New Set Object					
Name and Description Provide object name and description					
A Set object fires one or more process events that Synch objects are waiting on. Set and Synch objects are used to suspend and resume execution paths of a process.					
Name					
Set					
Description					
Set Object					

Type a *Name* to identify the Set object and, optionally, type a brief *Description*. The Design mode option allows you to include the object in the process flow as a place marker, that is, without defining its parameters. Click *Next* to continue configuring the object.

The Define Events dialog box opens, as shown in the following image.

🔬 New Set Object						
Define Event	S nore events					
Name	Value	Usage	Description	*		
? < Back Next > Finish Cancel						

Add one or more events that can be used to meet conditions defined in one or more Synch objects. Click the Plus button and type an event name in the *Name* column. Then, double-click and select a value and usage from the *Value* and *Usage* drop-down lists, and type a description of the event in the Description column.

When you finish defining the events, click *Finish*. The Set object icon appears in the workspace.

Handling Header Information

Header information is "metadata" carried along with the actual message payload, but it is not strictly a part of it. Many protocols, such as HTTP, IBM WebSphere MQ, and JMS, provide header information support, although often under names such as "user data" or "side data."

The server makes received header information available in special registers that take the name of the header field. For example, a message received through HTTP will usually carry a header field of "content-type," which can be referenced as sreg("content-type"); these registers carry the type of 'header'.

When messages are emitted, special registers of type header are associated, with the message as appropriate to the protocol. To add a special register of type header, the Set object can be used in a process flow. Set the register type to *'header'*.

The following image shows the Define Events dialog box (accessed during the Set object configuration) defining an event named MyHDR.

New Set Object In the Events Add one or more objects	t 9 e events which can be	used to meet conditions	define	d in one or more Synch		23
Name MyHDR	Value 123	Usage header user header	+	Description Header to be carried		*
?		< Back Nex	t>	Finish	Canc	el

Incoming headers are loaded into header special registers and emitted message headers are drawn from such registers. This provides a convenient method of transferring messages with headers across protocols. The ultimate treatment of header information is the responsibility of the application developer.

Synch Object

The Synch object waits for a condition to be met, such as the expiration of a time limit or the evaluation of an expression as 'true'. The conditions are based on the events created through the Set object.

The Synch object icon is shown below.



When you add a Synch object to a flow, the New Synch Object dialog box opens, as shown in the following image.

🔏 New Synch Object
Name and Description Provide object name and description
A Synch object waits for a condition to be met. A condition is the expiration of a time or the evaluation of an expression as true. Name
Synch
Description
Synch Object
(?) < Back Next > Finish Cancel

Type a *Name* to identify the Synch object and, optionally, type a brief *Description*. The Design mode option allows you to include the object in the process flow as a place marker, that is, without defining its parameters. Click *Next* to continue configuring the object. The Expression Builder dialog box opens, as shown in the following image.

🔏 New Synch Object					
Expression Builder Build an expression using one or more conditions which will suspend execution until the expression is met or a timeout occurs					
Conditions will be joined using an (a) AND (c) OR operation.					
Name	Event Name	Operation	Value		
۰ (ا					

To build a Synch expression, click the *Plus* button. The default, *Condition1*, appears under the Name column, as shown in the following image.

You can change the name and select the Event Name, Operation, and Value for the new condition. The Event Name drop-down list consists of the events defined in the process flow Set objects. Add as many conditions as required. Above the condition parameters, select the relationship between these conditions as an *And* or *OR* operation.

To delete a condition, select the condition, and click the *X* above the list of conditions.

Click Next. The Timeout dialog box opens, as shown in the following image.



Type a timeout value (in seconds). This is the time period for which the Synch object will wait for the expression to be evaluated as 'true'. A value of zero (0) means that the Synch object will wait indefinitely.

Click Finish. The Synch object icon appears in the workspace.

The Execution Objects

The following sections describe how to define each Execution object as you include it in a process flow.

BI Object

The BI object allows the process flow to run a WebFOCUS report.

The BI object icon is shown below.



When you add a BI object to a process flow, the New WebFOCUS Object dialog box opens, as shown in the following image.

New BI Object
Name and Description Provide object name and description
A WebFOCUS object executes a WebFOCUS Report. Name
Description
BI Object
Sack Next > Finish Cancel

Type a Name to identify the BI object and, optionally, type a brief Description.

Note: As a convenience, you can launch Developer Studio and Managed Reporting, two external WebFOCUS tools, by clicking the corresponding buttons in this dialog box. Designer automatically determines the presence and location of these tools. For details on using these tools, see the Developer Studio documentation set and the *WebFOCUS Managed Reporting End User's Manual*.

Once you have named the new WebFOCUS object, click *Next*. The WebFOCUS Connect dialog box opens, as shown in the following image.



Select or type the URL of the target WebFOCUS server and click Next.

Note: If you choose to use a localhost address, be aware that this URL is relative to the iWay Service Manager, not to the machine running Designer.

The next WebFOCUS Connect dialog box opens, as shown in the following image.

👔 New BI Object					
New WebFOCUS Object Connect to a WebFOCUS Ser	ct ver using the following connection information				
Туре					
Application	Reports produced for deployment as a Web application from WebFOCUS Developer Studio.				
Managed Reporting	Provides an integrated, scalable, true thin-client reporting.				
Security					
User ID:	•				
Password:					
Manager Reporting					
User ID:	~				
Password:					
?	ck Next > Finish Cancel				
-					

Select the source of the WebFOCUS report you want to include. Consult the WebFOCUS documentation or your WebFOCUS site administrator to determine where your reports are located. Type your login credentials for WebFOCUS and click *Next*.

If you have chosen Application in the previous dialog box, the WebFOCUS Reports dialog box opens, as shown in the following image.



If you selected Managed Reporting in the previous dialog box, you will see a dialog box similar to the following image.



In either case, navigate the hierarchy and select the desired report, then click *Next*. The Object Properties dialog box opens, as shown in the following image.

🔬 New BI Object				3
Object Proper	ties juired parameters			
URL				
MR User	Managed Report	ing User		
MR Password	Managed Report	ing Password	Ŧ	
	 Password: Expression: 		-	
User ID	Connection User	ID	•	
Password	Connection Pass	word		
	 Password: Expression: 	•••••	Ŧ	
Report Name	WebFOCUS Repo	ort Name		÷
?	< <u>B</u> ack N	ext > Einish	Cancel	

Verify that all the parameters are correct and click Finish.

The BI Object appears in the workspace.

Once a BI Object exists, you can configure it to accept dynamic properties. To do this, open the BI object for editing and select the Properties tab. Replace one or more of the properties with a special register (SREG), an XPath statement, or with any other iWay Functional Language statement. For information on using variables and Xpaths, see *Using Variables and XPath* on page 162.

Email Object

The Email object allows a process to send or receive an email with one or more attachments. The Email object icon is shown below.



When you add an Email object to a process flow, the New Email Object dialog box opens, as shown in the following image.

🔏 New Email Object		
Name and Description		
Provide object name and description		
An Email object allows processes to either send or receive an email with one or more attachements.		
Description		
Email Object		
🗖 Tap		
A Back Next > Finish Cancel		

Type a Name to identify the Email object and, optionally, type a brief Description.

Click Next. The Email Object Type dialog box opens, as shown in the following image.

🔬 New Email Object		
Email Object Type Select the type for the Email Object object you wish to create		
Defined Email Object services in the current configuration are listed Type		
Description		
? < Back Next > Finish Cancel		

Select the Email type from the *Type* drop-down list. The Standard Email Client emits an email with one or more attachments. The Attachment Extractor identifies an attachment extractor to be used on a flow that is deployed to an email listener.

Click *Next*. The Object Properties dialog box opens. The following image shows the Properties dialog box for the Standard Email Attach Agent email type.

🔬 New Email Ob	iject 📃 🗖 🗮 🗙	
Object Properties		
Provide object	properties	
▼ Main		
Attachment Tag	Tag used to identify an attachment (attachment is the default)	
	attachment 👻	
Encode	If true, everything is base64 encoded, if false only non-txt is encoded	
	false 👻	
Target	Target of attachments	
	file 👻	
Output	The target output directory	
Directory	-	
Output File Name	The output file name, which can contain a '*' which gets expanded to a fine time stamp, if empty attachment name is used	
Delete Mail Object	If no other component needs the mail object, it should be deleted	
	true 👻	
?	Back Next > Finish Cancel	

Type or select the property values to define the Email object and click *Finish*. The Email object icon appears in the workspace.

Emitter Object

The Emitter object enables behavior in your process flow equivalent to configuring an emitter through the iWay Console. In other words, the action of the Emitter is delayed until after all steps in your process flow are performed and all post-processing of the document is complete. A consequence of this delayed execution is that if there is an error during processing, and even if an error occurs in the flow after the Emitter, the emit will not occur. The significance of the placement of the Emitter object in your process flow is that it determines the routing information that will be used when the emission ultimately occurs. In contrast, the version of the document that is dispatched by the Emitter is the result of all processing and, therefore, is not necessarily the version at the point in the flow where the Emitter object is placed.



When you add an Emitter object to a process flow, the New Emitter Object dialog box opens, as shown in the following image.

🔬 New Emitter Object
Name and Description
Provide object name and description
An Emitter object adds an address to the payload using one of the supported protocols.
Name
Emitter
Description
Emitter Object
Warning: Emitter object runs at the end of process flow execution. If you like your emitter to execute in order is set, please use Service Object.
Tap

Type a Name to identify the Emitter object and, optionally, type a brief Description.

Click Next. The Emitter Type dialog box opens, as shown in the following image.

New Emitter Object	
Emitter Object Type Specify the type for the new Emitter object	
AQ	
misc AQ AS1 AS2 AS2 [nonblocking] CS3 Email	
Emit a message to Oracle Advanced Queuing	
? < Back Next > Finish Cancel]

Select the type of emitter the object will bind to; a predefined emitter, a local instance based on an existing template, or a protocol definition.

Select a protocol from the Protocol drop-down list. The choices are:

- **AQ.** Emit a message to Oracle Advanced Queuing.
- **AS1.** Emit a message over SMTP using AS1 specifications.
- **AS2.** Emit a message over HTTP using AS2 specifications.
- **AS2** [nonblocking]. Emit a message over HTTP using AS2 specifications.
- **CS3.** iWay Data Server access.
- **Email.** Emit a message to an SMTP host.
- **FTP[S] Client.** Emit a message using FTP to a specific directory on a defined host.
- **File.** Emit a message as a file to a specified directory.

- **HL7LLP.** Emit a message via TCP.
- **HTTP 1.0 [deprecated].** POST a message to a defined host:port/action.
- **HTTP 1.1 [nonblocking] (nhttp).** POST a message to a defined host:port/action.
- □ Internal Queue. Internal queue listener. Internal queues are used to pass messages between channels within a configuration. They facilitate the modularization of message processing.
- Java Message Service (jmsq). Emit a message to a specified Java Message Queue or publish to a Topic.
- **MQ.** Emit a message to a specified MQ Series queue.
- **Passthru.** A dummy emitter to use as a placeholder during development.
- **SFTP.** Emit a message using SFTP to a specific directory on a defined host.
- **Sonic.** Emit a message to be a specified SONIC message queue, or publish to a Topic.
- **TCP.** Emit a message through TCP.
- **Tibrv.** Emit a message to a Tibco Rendezvous queue.
- **Print.** Emit a message using a specified print server using IPP.

For more information on these protocols, see the *iWay Service Manager Protocols Guide*.

Click *Next*. The Object Properties dialog box opens. The properties that appear depend on the Emitter Type you selected. The following image is the dialog box that appears when you select the AQ protocol.

🔬 New Emitter (Dbject
Object Prope	erties
Provide all re	quired parameters.
▼ Main	A
Destination	queue@host , leave host portion blank if using RAC
	v
Host	Host of server connection
SID	SID of server connection
? <	Back Next > Finish Cancel

Note: All fields with titles in red are required parameters.

When you are finished defining the emitter, click *Finish*. The Emitter object icon appears in the workspace.

ETL Object

The Extract Transform Load (ETL) object executes an iWay DataMigrator flow. iWay DataMigrator supports traditional batch ETL and trickle-feed ETL for real-time data updates. Using the ETL object in a process flow enables:

Real-time data warehousing. Utilizing trickle-feed ETL, data can be acquired, cleansed, and updated without waiting for a nightly batch window.

❑ Process-driven business intelligence. This approach may use any of the other data integration mechanisms, but adds two important capabilities: the ability to monitor business processes and detect relevant events in them, and the ability to participate in business processes and contribute information to them.

ETL exposed as web services . Allows ETL jobs to participate in enterprise SOA solutions.

The ETL object icon is shown below.



When you add an ETL object to a process flow, the New ETL Object dialog box opens, as shown in the following image.

New ETL Object
Name and Description Provide object name and description
An ETL object executes a DataMigrator Flow.
Name
ETL
Description
ETL Object
🕅 Тар
Sack Next > Finish Cancel

Type a Name to identify the ETL object and, optionally, type a brief Description (optional).

Click Next. The DataMigrator Connect dialog box opens, as shown in the following image.

New ETL Object
ETL Name and Description Connect to DataMigrator Server using the following connection information.
DataMigrator Connection Host: Port: Server:
DataMigrator Security User ID: Password:
Rext > Finish Cancel

Type, or select from the drop-down list, the DataMigrator host name in the *Host* field and port in the *Port* field. Select the *Server* from the drop-down list. Type the User ID and the *Password* to access DataMigrator, in the fields provided under DataMigrator Security.

Note: A message window opens when you type or select an invalid credential.

Click Next. The DataMigrator Flows dialog box opens, as shown in the following image.

New ETL Object
DataMigrator Flows Select a DataMigrator Flow. Flows are grouped by Application Directories.
Run Time 📃 👻
Design Time Refresh List
 ▲ J Application Directories ➢ foccache ▷ ➢ ibisamp ▷ ➢ baseapp
Seck Next > Finish Cancel

From the Flows list, select the DataMigrator flow that you want to execute from the ETL object. Click *Next*.

Important: Designer supports ETL objects with names of 45 characters or less in length.

The Properties dialog box opens, as shown in the following image. This dialog box shows the properties of the ETL object you are creating.

new ETL Obje	ct 🗆 🗖 🗖	×
Object Prope i Provide all re	erties quired parameters	
•		-
Data Source URL	URL to reach the data source. Required unless using JNDI DataSource.	
	-	=
Document	Document to return	
Return Type	-	
User ID	Default user id for the connection	
	-	
Password	Default password for the connection	
	Password:	
	Expression:	
Statistic	The RPC to call to fetch ETL statistics. Called only when	÷
?	Back Next > Finish Cancel	

You can change the Document Return Type and Pool Connections properties, if necessary. Click *Next*.

The ETL Options dialog box opens, as shown in the following image.

New ETL Object		
ETL Options Enter the ETL job options.		
Do not wait for compl Run with limits Start loading at record: Stop loading at record:	etion 0 0	
Rack	Next > Finish	Cancel

Select *Do not wait for completion* to start a long-running ETL job asynchronously, in which case the process flow will continue immediately with the next object, or objects, in the flow.

The Run with limits options (Start loading at record and Stop loading at record) allow you to define a range of records for the ETL job. For example, selecting a small set of records might be preferable for testing purposes.

Click *Next*. The ETL Parameters dialog box opens. Enter parameters for ETL stored procedures and click *Next*. The ETL Return Codes dialog box opens, as shown in the following image.

🔬 New ETL Object	
ETL Return Co Create a list of exp for.	des pected return codes to create custom events
	₽ ≍00
Return Codes	Description
? < Ba	ck Next > Finish Cancel

If you want to add a return code to the object, select *Click here to Add*, under the Return Code column, to activate the field, and enter the return code. Type a description of the return code in the Description column. You can enter several return codes, as needed. Click *Finish*.

The ETL object appears in the workspace.

Once an ETL object exists, you can configure it to accept dynamic properties. To do this, open the ETL object for editing, and select the Properties tab. Replace one or more of the properties with a special register (SREG), an XPath statement, or any other iWay Functional Language statement. For information on using variables and Xpaths, see *Using Variables and XPath* on page 162.

File Object

The File object allows a process to perform file operations at execution time. The File object icon is shown below.



When you add a File object to a process flow, the New File Object dialog box opens, as shown in the following image.

🔬 New File Object	
Name and Description Provide object name and description	
A File object allows a process to perform file operations a Name File	it execution time.
Description	
File Object	÷
Tap	
Sack Next > Finish	Cancel

Type a Name to identify the File object and, optionally, type a brief Description.

Click Next. The File Object Type dialog box opens, as shown in the following image.

New File Object
File Object Type Select the type for the File Object object you wish to create
Defined File Object services in the current configuration are listed Type
Description
Seck Next > Finish Cancel

From the *Type* drop-down list, select the file type. The file types are:

- **File Emit Agent.** Emits a document to a physical file.
- □ **PF File Ops Agent.** Performs operations on the specified file(s) including copy, move, rename, prepend, append, delete, size, and exist.
- **PF File Read Agent.** Embeds a specified binary, xml, or text file in the input document.

Click *Next*. The Properties dialog box opens. The properties that appear depend on the File type you selected. The following image is the Properties dialog box for File Read.

🔬 New File Obje	ct	x
Object Prope	rties	
🙆 Provide all re	quired parameters.	
▼ Main		-
Name of File	File to be read. Relative or absolute file paths are supported explicitly or through a SREG or XPath expression evaluated using the incoming document	
		Ŧ
Delete after	Whether to delete the file read on a successful read	
read	false	Ŧ
Format	Format of the input data	
	flat	Ŧ
Tag	Name of the XML tag to wrap the data read in. Requir if format is flat	red
		• •
Kext > Finish Cancel		

Note: All fields with titles in red are required parameters.

Define the appropriate properties and click *Finish*. The File object icon appears in the workspace.

FTP Object

The File Transfer Protocol (FTP) object allows the process flow to send or receive data using FTP. The FTP object icon is shown below.



When you add an FTP object to a process flow, the New FTP Object dialog box opens, as shown in the following image.

New FTP Object
Name and Description Provide object name and description
A FTP object allows a process to send or receive data via FTP protocol.
Name
FTP
Description
FTP Object
Tap
Sack Next > Finish Cancel

Type a Name to identify the FTP object and, optionally type a brief Description.

Click Next. The FTP Object Type dialog box opens, as shown in the following image.

New FTP Object
FTP Object Type Select the type for the FTP Object object you wish to create
Defined FTP Object services in the current configuration are listed Type
Description
Seck Next > Finish Cancel

From the *Type* drop-down list, select the FTP type. The types are:

- **FTP Emit agent.** Emits via FTP protocol to a given host:port using various common FTP commands.
- □ **PFFTP Ops agent.** Performs operations on a specified file(s) including, copy, move, rename, prepend, append, delete, size, and exist.
- **PFFTP Read agent.** Embeds a specified binary, or text file, in the input document.
Click *Next*. The Properties dialog box opens. The properties that appear depend on the FTP type you selected. The following image is the Properties dialog box for FTP Reader.

new FTP Obje	ct 📃 🗖 💌 🔨	
Object Properties		
😣 Provide all required parameters.		
▼ Host Parame	ters	
Host Name	DNS name (or IP address) of the FTP server that you want to connect to. Use host : port if not standard port 21.	
	•	
Remote Port	Port to connect to on the FTP site, blank for default port 21	
	21 👻	
User Name	Is the valid user ID on the FTP server	
	•	
Password	Is the valid password for the FTP server	
	Password:	
	C Expression:	
Seck Next > Finish Cancel		

Note: All fields with titles in red are required parameters.

Type or select the property values to define the FTP object. Click *Finish* when all properties are defined. The FTP object icon appears in the workspace.

HTTP Object

The Hyper Text Transfer Protocol (HTTP) object allows a process flow to send or receive data using HTTP. The HTTP object icon is shown below.



When you add an HTTP object to a process flow, the New HTTP Object dialog box opens, as shown in the following image.

🔬 New HTTP Object
Name and Description Provide object name and description
An HTTP object allows a process to send or receive data via HTTP protocol. Name HTTP
Description
Пар
Sack Next > Finish Cancel

Type a Name to identify the HTTP object and, optionally, type a brief Description.

Click Next. The HTTP Type dialog box opens, as shown in the following image.

🔬 New HTTP Object
HTTP Object Type Select the type for the HTTP Object object you wish to create
Defined HTTP Object services in the current configuration are listed
Туре
·
Description
Seck Next > Finish Cancel

From the *Type* drop-down list, select the HTTP type. The HTTP types are:

HTTP Emit Agent. General HTTP Emitter for use with the agent stack.

HTTP Read Agent. Reads an HTTP source via HTTP GET. Returns result.

Click *Next*. The Properties dialog box opens. The properties that appear depend on the HTTP type you selected.

The following image is the Properties dialog box for HTTP Read Agent.

New HTTP Object	
Object Properties Provide object properties	
✓ Main	
URL Tag Name	Tag (element) name in input document containing the URL. Do not use if URL parm is used.
URL	URL for the read. Used if the tag (element) name is omitted.
Access Timeout	Timeout in milliseconds for the access; applies if greater than 0.
? < Back Next > Finish Cancel	

Note: All fields with titles in red are required parameters.

Define the appropriate properties and click *Finish*. The HTTP object icon appears in the workspace.

Iterator Object

The Iterator object allows you to repeat certain portions of a complete process flow or in specified sections of an input document. The number of times a flow is repeated can be based on either a count, the result of a Boolean expression, or the structure of the document itself.

Important: The iWay Registry contains only those classes that support the iWay Samples. Therefore, when using Iterator objects in a Registry-based project, you must add the iterator classes to the Registry from the base configuration through the iWay Service Manager Administration Console (select *Tools, Archive Manager, Import components from managed server, click Next, and select base).* See the *iWay Service Manager User's Guide* for details. The Iterator object icon is shown below.



The Iterator object works in conjunction with the Loop relation, which defines the path of the flow that is to be iterated. The following image is an example of a process flow using an Iterator object.



When the iterator completion condition is met, execution of the flow continues with the objects after the loop. Despite the appearance of the diagram in the Designer workspace at completion, the flow 'jumps' over the objects inside the loop. Otherwise, the logic inside the loop would be erroneously executed one additional time.

Some guidelines for using Iterator objects in a process flow are:

- The loop must begin with an Iterator object. The flow cannot branch into an iteration loop.
- □ Iterator loops can be nested. However, loop lines cannot cross.
- Only Execution objects (Service, Iterator, Transform, Process, WebService, SQL, ETL, BI, Email, FTP, File, HTTP, Emitter, Queue) can end a loop.
- Loops cannot be drawn from Control objects (Join, Junction, Decision Switch, Decision Test, Set, Sync, Debug).

When you add an Iterator object to a process flow, the New Iterator Object dialog box opens, as shown in the following image.

🔬 New Iterator Object
Name and Description
Provide object name and description
An Iterator object is a unit of work that emits stock events and/or one or more custom events. Availability of iterators is dependent upon Target Server Version specified by the Process Flow. By extending the available iterators list via project Customizations user can enlist custom iterators directly within a process.
Iterator
Description
Iterator Object

Type a Name to identify the Iterator object and, optionally, type a brief Description.

Once you have named the new Iterator object, click *Next*. The Iterator Object Type dialog box opens, as shown in the following image.

New Iterator Object		
Iterator Object Type Select the type for the Iterator Object object you wish to create		
Defined Iterator Object services in the current configuration are listed Type		
Description Image: Concentration Image: Concentration Image: Concentration		

The Iterator objects are instantiated on the iWay Service Manager server as agents. Iterator agents are distinguished from other agents by their name, which contains the string "ITER". The available Iterator types are:

- Attachment Iterator {com.ibi.agents.XDIterAttachments}. Handles each attachment of the current document iteratively.
- □ Chunk Iterator {com.ibi.agents.XDIterChunk}. Splits the input document into chunks of equal size.
- Condition Iterator {com.ibi.agents.XDIterCond}. Iterates a loop while (or until) a condition is met.
- □ Counter Iterator {com.ibi.agents.XDIterCount}. Iterates the loop an exact number of times. Analogous to the programming construct called a "for/next loop."
- □ FTP Emit Iterator {com.ibi.agents.XDIterFTPEmitStdAgent}. Emits via FTP protocol to a given host:port using various common FTP commands.

- □ Island Iterator {com.ibi.agents.XDIterIsland}. Iterates the loop for each island within the document, replacing the island with the result of processing the island text. Islands can be used for Document Enrichment. Islands cannot be nested; only the outermost islands will be iterated.
- □ Iter RVI Relay {com.ibi.agents.XDIterRVIRelay}. Relays messages for external execution.
- □ Line Iterator {com.ibi.agents.XDIterLine}. Iterates the loop for each line in the input document. Lines are assumed to be separated by standard end-of-line delimiters.
- □ SQL Batch Insert Iterator {com.ibi.agents.XDIterSQLBatchAdd}. Batch Add Iterator; Reads SQL rows one at a time and inserts results from the flow.
- □ SQL Select Iterator {com.ibi.agents.XDIterSQLSelect}. SQL Select Iterator; Sends out rows, one at a time, to the flow.
- □ Sibling Iterator {com.ibi.agents.XDIterSibSplit}. Iterates the loop for each portion of the XML document on a specified XML tag. Ideal for processing repeating segments within the document.
- □ **Test Iterator {com.ibi.agents.XDIterTest}.** Iterates the loop while (or until) a specific condition is met. Analogous to the programming construct called a "while loop."
- **XML Iterator {com.ibi.agents.XDIterXMLSplit}.** Iterates a loop for each portion of an XML document.

Note: See the *iWay Service Manager User's Guide* for information on island, sibling, and status, as used in iWay.

Once you have selected an iterator type, click Next.

The Properties dialog box opens. Each iterator type has its own set of parameters, so this dialog box will only list those parameters associated with the selected iterator. The following image shows the Object Properties dialog box for the Counter iterator type.

🔏 New Iterator	Object 📃 🔲 💻 🕮	
Object Prop Provide object	erties properties	
▼ Main		
Iterations	Number of times to repeat the loop	
	1	
Interim Document	The document emitted at each iteration is the original document or the result of the previous iteration.	=
	transient 👻	
Output Document	Final document emitted is a status document, the original document, the result of the last iteration or an accumulation. Accumulations are memory intensive.	
	status	_
?	< Back Next > Finish Cancel]

Select or enter the parameters required to configure the Iterator object and click *Finish*. The Iterator object appears in the workspace.

To view details and examples for iterator object properties, see *Iterator Object Properties* on page 217.

Process Object

The Process object facilitates code reuse and modularity by inserting a published process flow into the current process flow. The affect of using the Process object is much like using a subroutine call in traditional programming languages. The Process object icon is shown below.



When you add a Process object to a process flow, the New Process Object dialog box opens, as shown in the following image.

New Process Object		
Name and Description Provide object name and description		
A Process object inserts a published system process into this process as an object.		
Name		
Process1		
Description		
Process Object		
Пар		
Seck Next > Finish Cancel		

Type a Name to identify the Process object and, optionally, type a brief Description.

Click Next. The Process Object Properties dialog box opens, as shown in the following image.

🔬 New Process Object	
Object Properties Provide object properties	
- Process	Defined process flow name to lookup at runtime.
Description	Enter a properly formatted iWay function such as TPA(), or enter the name of an existing process to lookup at
(?) < Ba	ick Next > Finish Cancel

Key in the process flow you want to associate with the Process object and click Finish.

Note: Only previously published process flows are available to the Process object. In the case of a Registry-based project, the process flow must have been published to the Registry. In the case of a server-based project, the process flow must have been published to the system area of the named Server Configuration, for example, base.

The Process object icon appears in the workspace.

Queue Object

The Queue object sends or receives documents using a supported queue type. The Queue object icon is shown below.



When you add a Queue object to a process flow, the New Queue Object dialog box opens, as shown in the following image.

🔬 New Queue Object 📃 📼 💌		
Name and Description Provide object name and description		
A Queue object sends or receives documents using a supported queue type.		
Name		
Queue		
Description		
Queue Object		
🥅 Тар		
Sack Next > Finish Cancel		

Type a Name to identify the Queue object and, optionally, type a brief Description.

Click Next. The Queue Object Type dialog box opens, as shown in the following image.

🔬 New Queue Object		
Queue Object Type Select the type for the Queue Object object you wish to create		
Туре		
Description		
Seck Next > Finish Cancel		

Select the queue *type* from the Type drop-down list.

The available queue types are:

Sonic MQ. Emits a document using Sonic MQ.

The Sub Type selection is Output.

Oracle AQ. Emits a document using Oracle Advanced Queuing facility (AQ).

The Sub Type selection is Output.

- □ **JMS Queue.** Reads a file from a Java Message Service queue and returns a result. The Sub Type selection is #STARTUP_JMS_L.
- ❑ Tibco Rendezvous. Emits a document using Tibco Rendezvous messaging. The Sub Type selection is Output.
- **MQ Series.** Reads a single file from an MQSeries queue.

The Sub Type selection is Output.

Click *Next*. The Properties dialog box opens. The properties that appear depend on the Queue Type you selected. The following image is the dialog box that appears when you select SonicMQ.

🔏 New Queue Object		
Object Properties		
Ø Provide all rec	quired parameters.	
▼ Main		
Queue Name	The name of the receiver queue	
	·	
Broker URL	Broker url [protocol://]host:port (can be comma separated list used for failover). Protocols are TCP, HTTP	
User ID	User ID for challenges	
	· · · · · · · · · · · · · · · · · · ·	
Password	Password for challenges	
	Password:	
	Expression:	
? < B	ack Next > Finish Cancel	

Note: All field titles in red are required parameters.

Define the appropriate properties and click *Finish*. The Queue object icon appears in the workspace.

ReportCaster Object

The ReportCaster object immediately submits one or more selected schedules to a ReportCaster Distribution server for execution. The ReportCaster object icon is shown below.



When you add a ReportCaster object to the process flow, the New ReportCaster Object dialog box opens, as shown in the following image.

New Report Caster Object
Name and Description Provide object name and description
A ReportCaster object immediately submits selected scheduales to the ReportCaster Distribution Server that runs the scheduales. Name
Description
Report Caster Object
Sack Next > Finish Cancel

Type a Name to identify the ReportCaster object and, optionally, type a brief Description.

Click *Next*. The ReportCaster Server Connect dialog box opens, as shown in the following image.

New Report Caster Object
ReportCaster Server Connect
Connect to a ReportCaster server using the following connection information.
Connection
URL:
User ID:
Password:

Enter the URL to the ReportCaster server, and the login credentials to access the server.

Click *Next*. The ReportCaster Schedules dialog box opens displaying the schedules available to you. The following image shows an example of this dialog box.

New Report	Caster Object						
ReportCaster Schedules Select ReportCaster schedules you would like to trigger.							
		S	Check	k All Uncheck All			
Schedule Id	Description	Next Run T	Distributio	Active			
S14234	test1	None	Email	true			
S14494	mr report	None	Email	true			
?	Back	Next >	Finish	Cancel			

Select the schedule(s) you want to trigger with this ReportCaster object by selecting the corresponding check box in the Schedule Id column.

Click Finish. The ReportCaster object icon appears in the workspace.

Script Object

The Script object allows you to execute java script on a process flow's object. The Script object icon is shown below.



When you add a Script object to the process flow, the New Script Object dialog box opens, as shown in the following image.

New Script Object
Name and Description Provide object name and description
Script Object allows to execute JavaScript on process flow's object Name
Script1
Description
Script Object

Type a Name to identify the Script object and, optionally, type a brief Description.

Click Next. The Script Object Properties dialog box opens, as shown in the following image.

🔬 New Script Object
Object Properties Provide object properties
Script File Script file location
A Back Next > Finish Cancel

Enter the Script file location and click Next to configure further, or click Finish.

The Pre-Execution dialog box opens. The following image shows an example of this dialog box.

👔 New Script	Object						
Pre-Execut Please enter	t ion pre-exe	cution varia	bles.				
Name	Ту	Value	Us	Sc	Ac	Description	
?	< Back	N	lext >		Finisl	h Ca	ncel

Enter the variables in their respective columns. Click Next to configure further, or click Finish.

The Post-Execution dialog box opens. The following image shows an example of this dialog box.

🔬 New Script	Object						
Post-Exect Please enter	ution post-ex	ecution varia	ibles.				
Name	Ту	Value	Us	Sc	Ac	Description	
?	< Back	N	ext >		Finis	h Ca	ancel

Enter the variables in their respective columns. Click Next to configure further, or click Finish.

The Debug Settings dialog box opens. The following image shows an example of this dialog box.

🔬 New Script Object				
Debug Settings Object debug settings				
Tracing Format Standard tracing Iteration tracing limits When an object is part of an iterative loop, the maximum number of iterations which can be captured is 20. Tracing can be controlled by specifying the iteration which begins and ends document capture.				
Tracing begins at iteration	1			
Tracing ends at iteration	10			
? < Back	Next > Finish Cancel			

Choose the Tracing Format from the drop-down list. The choices are

- Standard tracing.
- Standard tracing displayed in hex format.
- □ Just the output document.
- □ None, under any circumstances.

Select the iterations in which you want the tracing to commence and end.

Click Finish. The Script object icon appears in the workspace.

Service Object

The Service object is a unit of work that emits default events and/or one or more custom events. The available services are based on the server side configuration. By extending the configuration, you can enlist custom services directly within a process flow.

Important: The iWay Registry or configuration contains only those service classes that are defined in that Registry or configuration. If the Registry or configuration requires additional service classes for a particular process flow, you must add them through the iWay Service Manager Administration Console (select *Tools, Archive Manager, Import components from managed server,* click *Next,* and select *base.*) See the *iWay Service Manager User's Guide* for details. For example, initially the iWay Registry contains only those service classes associated with the iWay samples. Therefore, you must add any additional classes required when new Service objects are added to projects in the Registry.

The Service object icon is shown below.



When you add a Service object to a process flow, the New Service Object dialog box opens, as shown in the following image.

🖉 New Service Object
Name and Description
Provide object name and description
A Service object is a unit of work that emits stock events and/or one or more custom events. Availability of services is dependent upon Target Server Version specified by the Process Flow. By extending the available services list via project Customizations user can enlist custom services directly within a process.
Name
Service
Description
Service Object
Tap

Type a *Name* to identify the Service object and, optionally, type a brief *Description*, or accept the default option.

Click *Next*. The Service Type dialog box opens, as shown in the following image.

New Service Object	
Service Object Type	
Specify the type for the new Service object	
accumulation	*
attachments	
DOC	
emit	
exit_preemit	-
Description	
·	÷.
Compared Rest > Finish	Cancel

Select a defined service from the list.

Note: If you have selected or defined a Control agent, in order to protect the run-time configuration from tests within Designer, the Control agent will only work correctly after the process flow of which it is a part has been published to a listener. If you choose to Build, Test, and Run the flow before publishing to a listener, the flow will result in an error.

Click Next.

The properties that appear in this dialog box depend on the service you select. The following image shows the Properties dialog box for a control service.

🔬 New Service	Object
Object Prop Provide object	erties t properties
▼ Main	
Delay	Number of seconds to delay before returning the output document (e.g. 2.5) 0
?	Back Next > Finish Cancel

Enter the properties to define the service, and click *Finish*. The Service object icon appears in the workspace.

SQL Object

An SQL object is used to execute simple SQL requests within a process flow. The request can use XPath values within the document to drive the request. The SQL object icon is shown below.



The SQL object enables the execution of standard SQL statements against a target RDBMS from inside your process flow. Supported commands include Data Manipulation Language (DML), Data Definition Language (DDL), and other miscellaneous commands (GRANT, COMMIT, and so on) against the database. Care should be taken when designing process flows that perform potentially destructive operations on the database (DELETE, DROP TABLE, and so on) to prevent unintended data loss. Typical applications include use of the SELECT, INSERT, UPDATE, and DELETE commands.

To invoke stored procedures, the CALL statement is provided and can include properties that are defined for input or input and output use. CALL statement "output-only" parameters are currently not supported, due to limitations in binding iWay registers to SQL statements.

When you add an SQL object to a flow, the New SQL Object dialog box opens, as shown in the following image.

🔬 New SQL Object
Name and Description Provide object name and description
A SQL object uses an iWay request to drive a named JDBC adapter.
Name
SQL
Description
SQL Object
🥅 Тар
Sack Next > Finish Cancel

Type a Name to identify the SQL object and, optionally, type a brief Description.

Click Next. The Data Source URL dialog box opens, as shown in the following image.

🖉 New SQL Object
Data Source URL Enter the Driver and the URL for the data source
Enter the JDBC Driver and the URL to reach the data source by selecting a defined JDBC Data Server or by manually entering the connection information.
JDBC Driver:
A Back Next > Finish Cancel

Specify a Java Database Connectivity (JDBC) driver and URL to access the data source, by either selecting a defined JDBC server from the *Connection Profile* drop-down list, or entering the connection information in the *JDBC Driver* and *URL* fields.

Click Next. The SQL Statement dialog box opens, as shown in the following image.

🔬 New SQL Object		
SQL Statement		
Enter the body of th text, a XPATH, or a S	e SQL statement. The body can be actual special Register	
		÷
		_
Connection Profile:	· · · · · · · · · · · · · · · · · · ·	
Database:		
	SQL Builder	
? <u>Aack</u>	<u>N</u> ext > <u>F</u> inish Cancel	

Type the SQL statement and click Next.

The Properties dialog box opens. An example of this window is shown in the following image.

🔬 New SQL Objec	t
Object Proper Provide object pr	ties roperties
▼ Main	
Evaluate SQL	If set to true, an attempt is made to treat the SQL as an expression. This may not be appropriate to all SQL. For example, quotes in evaluated strings must be escaped.
	false 👻
Output Format	Format of result
	row
Use JNDI	If true, get a connection from a data source accessed via JNDI. Otherwise, connect directly using JDBC DriverManager. When true the <i>Pool Connections</i> configuration parameter should not be used, and pooling is handled in the III
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Cancel

Enter the required properties for the SQL object and click *Next*. For information on the SQL object properties, see *SQL Object Properties* on page 233.

Another Properties dialog box opens, in which you can define your own properties for the SQL object. An example of this dialog box is shown in the following image.

🔬 New SQL C	bject			
User Prope Please enter	e rties variables.			
Name	Туре	Value	Description	<i>[</i> ≁
?	< <u>B</u> ack	Next >	<u> </u>	Cancel

When you have completed defining the properties, click *Finish*. The SQL object icon appears in the workspace.

Data Integration Object

The Data Integration object iterates over an SQL result set and batch inserts.

The Data Integration object icon is shown below.



When you add a Data Integration object to a process flow, the New Data Integration Object dialog box opens, as shown in the following image.

Name and Description Provide object name and description Iterates over a SQL result set and batch inserts. Name Data Integration1 Description
Iterates over a SQL result set and batch inserts. Name Data Integration1 Description
Name Data Integration1 Description
Data Integration1
Description
Description
Data Integration Object
? < Back Next > Finish Cancel

Type a Name to identify the Data Integration object and, optionally, type a brief Description.

Once you have named the new Data Integration object, click *Next*. The Select Statement dialog box opens, as shown in the following image.

🔬 New Data Integrati	on Object	
Select Statement		
Enter the body of the actual text, a XPATH	e SQL select statement. The body can be , or a Special Register	
		*
Connection Profile:	•	
Database:		
	SQL Builder	
? Kack	Next > Finish Cancel	

Enter the body of the SQL statement. The body can be actual text, an XPATH, or a Special Register. If required, select the *Connection Profile* and the *Database*.

Click Next. The Insert Statement Parameters dialog box opens, as shown in the image below.

🔌 New Data Integratio	on Object	
Insert Statement F Please enter paramet	Parameters er names and values	
Table Name:		
		🕂 /* 🗙 😫
Name	Value	
		Generate
		÷
? < Back	Next >	Finish Cancel

Enter the Table Name, parameter name, and parameter values. Click the *Generate* button. The input statement, with the specified parameters, is displayed in the box below.

Click Next. The Properties dialog box opens, as shown in the following image.

🔬 New Data Integ	gration Object	
Object Proper Provide all req	ties uired parameters.	
▼ Source		
Input Provider	Name of the provider for accessing input	=
	-	
Select SQL	SQL Select statement to get rows to pass into the flow	
	test 👻	
Transaction	Transaction isolation level to be set if possible	
Isolation Level	asis 🔻	_
? < Ba	ack Next > Finish Cancel	

Note: All properties with titles in red are required parameters.

When you are finished selecting all of the properties, click *Next*. The Debug Settings dialog box opens, as shown in the following image.

🔌 New Data Integration Obje	ect 🖂 🖂 💌 X
Debug Settings Object debug settings	
Tracing Format Standard tra Iteration tracing limits When an object is part of an iterations which can be cap specifying the iteration whi	n iterative loop, the maximum number of tured is 20. Tracing can be controlled by ch begins and ends document capture.
Tracing begins at iteration	1
Tracing ends at iteration	10
? < Back	Next > Finish Cancel

Choose the Tracing Format from the drop-down list. The choices are:

- Standard tracing.
- Standard tracing displayed in hex format.
- Just the output document.
- None, under any circumstances.

Select the iterations in which you want the tracing to commence and end.

Click Finish. The Data Integration object icon appears in the workspace.

Transform Object

The Transform object applies a selected transform to the document it receives within the process flow. It also allows you to launch iWay Transformer to create a new transform for this object.

The Transform object icon is shown below.



When you add a Transform object to a process flow, the New Transform Object dialog box opens, as shown in the following image.

🔬 New Transform Object
Name and Description Provide object name and description
When inserted into a process a Transform object will apply a selected transform to the payload of the current path of execution.
Transform1
Description
Transform Object
Tap
Sack Next > Finish Cancel

Type a Name to identify the Transform object and, optionally, type a brief Description.
Once you have named the new Transform object, click *Next*. The Object Properties dialog box opens, as shown in the following image.

🔌 New Transform	m Object
Object Prope Provide object	erties properties
▼ Transform	Defined transform name to lookup at runtime.
Description	Enter a properly formatted iWay function such as TPA(), or enter the name of an existing transform to lookup at runtime.
?	< Back Next > Finish Cancel

The Pre-Execution dialog box opens, allowing you to enter the pre-execution properties for the transform type you selected. An example of this dialog box is shown in the following image.

New Tran	sform Ob	ject					
Pre-Exect	ution r pre-exec	ution varial	bles.				
Name	Ту	Value	Us	Sc	Ac	Description	•
							X
							1*
							Ê
							Û
							Ð
?	< Back		lext >		Finish	Car	ncel

When you are finished entering the transform properties, click Next.

The Post-Execution dialog box opens, allowing you to enter the post-execution properties for the transform type you selected.

Click Finish. The Transform object icon appears in the workspace.

Web Service Object

The Web Service object allows the execution of a web service from within a process flow. The web service can be an external service, such as a third party web service, or a web service hosted by iBSP.

The Web Service icon is shown below.



When you add a Web Service object to a flow, the New Web Service Object dialog box opens, as shown in the following image.

New Web Service Object
Name and Description Provide object name and description
A Web Service object executes a Web Service passing the incoming document as the payload. Name
Web Service Description
Web Service Object

Type a Name to identify the Web Service object and, optionally, type a brief Description.

Click Next. The WSDL Location dialog box opens, as shown in the following image.

🔬 New Web Service Object	
WSDL Location Enter the URL of a WSDL or load it from a location on disk.	
WSDL Location:	Browse
Kext > Finish	Cancel

In the WSDL location field, provide the location of the Web Service Descriptive Language (WSDL) file that describes the web service you are adding to the process flow, in one of the following ways:

- Type a URL.
- Browse to a WSDL.
- □ Point to an iBSP repository.

As an option, you can provide the location to a Web Service Descriptive Language (WSDL) file that describes the web service you are adding to the process flow. You can either manually enter the location, click the ellipses (...) to load the Local WSDL File, or click *Browse* to browse to a WSDL file.

If you select the ellipses (...), a local directory of your project opens in which you can locate the WSDL file.

If you select *Browse*, the Windows box of all directories of your computer opens. This dialog box allows you to select a WSDL from an instance of iWay Business Service Provider.

Click Next.

The WSDL Services dialog box opens, as shown in the following image.

🔏 New Web Service Object 📃 🔲 💌 🏹
Service Selection
Please select a service from the WSDL
Service: Material_Master
Description: Retrieves a list of materials from SAP.

Select a service from the Service drop-down list and click Next.

The Web Service End Point dialog box opens, as shown in the following image.

🔬 New Web	Service Object
Port, End	point, Action
Use the Po Operation.	rt Name to load default properties for End Point and Type values into the fields to define the End Point
PortName:	User Defined 🗸
End Point:	
Action:	▼
?	< Back Next > Finish Cancel

The *PortName*, *End Point*, and *Action* fields are automatically populated with the related values when you select the Port Name. You can also manually define all fields when you select *User Defined*. Click *Next*.

The Web Service Header dialog box opens, as shown in the following image.

🔬 New Web Service Object
Service Header
Enter an action to take on the Response of the WebService. The Response actions available are Transformation or XPATH
Header input is: < Empty >

Select or type the header information in the Header Input is field and click Next.

The Web Service Body dialog box opens, as shown in the following image.

🔏 New Web Service Object			
Service Body			
Enter the body of the message. The Body can be the actual text inside <soap-env:body> tags, a transformation file, a path to a</soap-env:body>			
Body input is: <a>Empty >]		

From the *Body Input is* drop-down list, select the body of the web service.

Note: The body can be the actual text inside the <SOAP-Env:Body> tags, a transformation, a path to a file containing the body, an XPath, or an SREG.

Click Next. The Web Service Response dialog box opens, as shown in the following image.

New Web Service Object
Service Response
Enter the Header of the WebService. The Header can be the actual text inside <soap-env:header> tags, a transformation</soap-env:header>
Response Action: < Take No Action >

From the Response Action drop-down list, select an action to take on the response of the web service. Click *Next*. The Web Service Fault dialog box opens, as shown in the following image.

🔬 New Web S	ervice Object	
Service Fa	ult	
Enter an acti occurs. The f	on to take on the Fault of the WebService if one ault actions available are Transformation and	
Fault Action: < Take No Action >		
?	< Back Next > Finish Cancel	

From the Fault Action drop-down list, select an action to take if there is a fault with the web service. Click *Next*.

The Proxy Configuration dialog box opens.

🔬 New Web Servic	e Object	x
Connection	provy connection settings	
	proxy connection settings	
Emit through Pr	roxy server	
Proxy Settings		
Proxy URL:		Ŧ
Proxy User ID:		Ŧ
Proxy Password:		
HTTP Settings		
Provid	e user id and password for Basic Authentication challeng	ges a
User ID		-
Password		
Timeout		Ŧ
		_
? < Bac	k Next > Finish Cance	!

If you are using a proxy server for the web service, select *Emit through Proxy server*, and enter the Proxy URL and the ID and password for accessing the proxy server. Click *Next*.

The Strip SOAP Envelope dialog box opens, as shown in the following image.

🔬 New Web Service Object
SOAP Envelope
The Response Document will be contained in a SOAP Envelope. Select the checkbox to remove the Envelope.
Strip the SOAP Envelope
? < Back Next > Finish Cancel

Check *Strip the SOAP Envelope* if you want to have the response document removed from the SOAP envelope. Click *Finish*.

The Web Service object appears in the workspace.

Note: For Web Service objects, Designer will use all faults defined in the WSDL as custom events. Edges that refer to specific faults can be constructed by selecting from the OnCustom list of cases. For more information, see *Building Relationships Between Objects* on page 132.



Tasks

This section describes specific tasks related to iWay Integration Tools (iIT) Designer.

In this chapter:

- Creating a Project
- Managing Process Flows
- Adding an Adapter to iIT Designer
- Using Variables and XPath
- Adding an Object to a Process Flow from an iSM Registry
- Working with Customizations

Creating a Project

Integration projects are containers that provide a way to organize resources and process flows into a structure relevant to your enterprise or department. Each project you create provides a group of folders:

- Adapters
- Applications
- Channels
- Ebixes
- Flows
- Registers
- Schemas
- Transforms
- XML

Procedure: How to Create a Project

To create a project:

1. From the File menu, choose *New* (or right-click the repository icon), and click *Integration Project*.

The New Integration Project dialog box opens, as shown in the following image.

New Integration Project
Integration Project Create a new Integration project.
Project name: Project location Use default Directory: C:\Program Files (x86)\iIT Browse
Additional options Image: Create integration folders Target Server Version:
Finish Cancel

- 2. In the Project name field, type a descriptive name for the project.
- 3. In the Project location field, use the default location, or deselect the *Use default* box, and browse to the desired location.
- 4. If you do not want to Create integration folders, deselect the corresponding check box in the Additional options section.
- 5. Select the Target Server Version from the drop-down list.
- 6. Click Finish.

The new project, with all the subfolders, appears in the Navigator pane. An example of a new project node, Work05, and its subfolders is shown in the following image.



A newly created project includes the following default elements:

- **Adapters.** Node contains the adapters available in Service Manager.
- **Applications.** Node contains the applications available in Service Manager.
- **Channels.** Node contains the channels available in Service Manager.
- **Ebixes.** Node contains the ebixes available in Service Manager
- **Flows.** Node contains the process flows you create for the project.
- **Registers.** Node contains the registers you create for the project.
- □ **Schemas.** Node contains the schema you define, as well as default schemas (PFIVP, PFIVPResponse, and Any).
- **Transforms.** Node contains the transformation files published on Service Manager.
- **XML.** Node contains the XML files published on Service Manager.
- 7. To save the project, right-click the project node, and select Save from the drop-down list.

Working with a Project

Once a project exists, you can change its properties and begin to build the project by adding elements, such as process flows, services, adapters, emitters, and schemas. Project elements are either created in iIT or imported to the project. For example, a process flow can be created in iIT Designer or imported from another source, while services, adapters, and emitters are imported from Service Manager.

This section explains how to:

- Open and close a project.
- Refresh a project.
- □ Import a process flow to a project.

Opening and Closing a Project

To gain access to a project and its components, you must open the project. To open a project, either click the project icon or right-click the project node and select *Open Project* from the drop-down list. The project node in the Navigator pane expands to display the project components.

To close a project, right-click the project node and select *Close Project* from the drop-down list. When you close a project, you have the option to save the project. Once a project is closed, it cannot be changed in iIT Designer until it is opened again. Although the resources of a closed project do not appear in the Navigator pane, they still reside on the local file system.

Refreshing a Project

The refresh option reloads a project from the repository and displays the most current version of that project in the Designer window.

To refresh a project, select *Refresh* from the File menu, or right-click the project and select *Refresh* from the drop-down list.

Viewing Project Properties

The project properties are the project name, description, where it is stored, and all other information pertaining to that particular project. To view project properties, right-click the project of interest, and select *Properties* from the drop-down list. The properties dialog box opens displaying the following tabs, from which you can obtain project information.

Resource. Contains the Path, Type, and Location of the project. Also contains the last modified date, Text file encoding options, and New text file line delimiter options.

- **Builders.** Allows you to configure the builders for the project.
- iWay Integration. Contains the following sections: Application, Customizations, Default iSM Target Versions, Export Folders, IFL Binding Properties, Process Flow and Transformer Encoding.
- □ **Project References.** Projects may refer to other projects in the same workspace. Use this page to specify what other projects are referenced by this project.
- **Refactoring History.** Allows you to view refactorings that affect this project.
- Run/Debug Settings Allows you to manage launch configurations associated with the currently selected resource.
- **Task Tags.** Allows you to enable project specific task tags.
- **Validation.** Allows you to select validators for the project.

Importing a Process Flow to a Project

Multiple iIT Designer users can be using a single repository, but only those projects and process flows that you create or import into your project will appear in your instance of iIT Designer. The import feature allows you to bring a process flow used by or created for an outside project, into one or more of your projects.

The following procedure explains how to import a process flow.

Procedure: How to Import a Process Flow

To import a process flow:

1. Under the project to which you want to import a process flow, right-click and select Import.

2. The Select dialog box opens, as shown in the following image.



3. Expand the iWay Integration node and select Process Flow, as shown in the image below

A Import	
Select	
Import an existing Process Flow	
Select an import source:	
type filter text	
 ▲ ➢ iWay Integration ➡ Ebix f() IFL Expression ➡ iWay Archive ➡ Process Flow ➡ Schema Set ➡ Transform 	A III
Sack Next > Finish	Cancel

4. Click Next. The General Properties dialog box opens, as shown in the following image.

A Process Flow Import	Wizard						
General Properties Page							
8 Please provide a valid process flow file to import							
Project Folder: /Work05/Flows Browse							
Import:							
Name:							
Description:	A						
	-						
Target Server Version:	7.0.0 💌						
? < Back	Next > Finish Cancel						

The default Flows folder is populated automatically in the Project Folder field. Click *Browse* to change the location.

To browse to a Process Flow file, click the ellipsis (...) found next to the Import field. Navigate to the file you would like to import. Click *Open*.

Select the Target Server Version from the drop-down list.

5. Click Finish.

The process flow is imported to the selected project and appears in the Navigator pane under the Flows folder.

Managing Process Flows

Creating a Process Flow

You must create a project before you can design a process flow. If you have not created a project, see *Creating a Project* on page 119 for instructions.

Creating a process flow in iIT Designer is a drag-and-drop operation. Every process flow begins with a Start object, which appears in the workspace by default when you create a new process flow. A process flow must also contain at least one end object, which you can add to the workspace at any time. You cannot validate or publish a process flow without an end object.

To create a process flow, you will include one or more of the objects provided by iIT Designer. After you add an object to the process flow, you then establish a relationship between that object, and the previous object in the flow. This relationship defines how, and under what conditions, you can route the document through the process flow.

Procedure: How to Create a Process Flow

There are two methods to create a new Process Flow:

1. On the main menu bar, click *File*, then *New*. Select *Other...* from the drop-down list, as shown in the following image.

File	Edit View	Navigate	Search	Project	Run Window	Hel	lp	
	New				Alt+Shift+N	Þ	1	Integration Project
	Open File						Ĵ	Project
	Close				Ctrl+W		۱	Application
	Close All				Ctrl+Shift+W	1		Channel
	Save				Ctrl+S		å	Process Flow
	Save As						fix	Transform
R	Save All				Ctrl+Shift+S			Register Set
1.1.1.1	Revert						67	Schema Set
	Maure					-	fCJ	IFL Expression
	Rename				E2	,	CĴ	Example
	Kename				F2			
8	Refresh				F5		Ľ	Other Ctrl+N
	Convert Line	Delimiters T	0			•]		The second se

2. In the Select a wizard pane, expand the iWay Integration folder, select *Process Flow*, and click *Next*.



3. You may also create a new Process Flow by right-clicking on the *Flows* folder of the desired project. Click *New* and *Process Flow*, as shown in the following image.



4. The General Properties dialog box is displayed, as shown in the following image.

🔬 New Process Flow Wizard	
General Properties	
Please select a project locat new Process Flow	ion and choose a name for the
Project Folder:	/Sample_Project/Flows Browse
Name:	Sample_flow
Description:	
Target Server Version:	7.0.0
	Enable taps
?	Finish Cancel

- 5. In the Project Folder field, use the *Browse* button to select a folder to contain the new Process Flow.
- 6. In the Name field, type a descriptive name for the process flow.
- 7. In the Description field, optionally enter a description for the new Process Flow.
- 8. Using the Target Server Version drop-down list, optionally select a non-default target version, .
- 9. Select the *Enable taps* check box to enable real-time data capture and reporting. For more information on configuring and using Taps, see the *iWay Enable User's Guide*.

10. Click *Finish*. The new Process Flow is listed in the Integration Explorer view in the selected location. The workspace displays a Start object, and the Properties pane appears below, displaying the properties of the Start object.



You are ready to add objects to the process flow.

Adding Objects to a Process Flow

Once you have created a process flow, you can continue to modify its design by adding objects according to your requirements. Every process flow requires at least one End object to terminate the flow. This section describes how to include an object in your process flow. For details about each object available to you in iIT Designer, see *Defining Process Flow Objects* on page 25.

Procedure: How to Add an Object to the Process Flow

To add an object to the process flow:

1. From the Object toolbar, click the object icon you want to add, and either drag it to the workspace, or click a blank area of the workspace.

Note: You can also right-click a blank area of the workspace. Select the object you wish to add from the New Process Flow Object dialog box.

The New (type) Object dialog box for the selected object opens. The following image shows an example of this dialog box for a new File object.

New File Object
Name and Description Provide object name and description
A File object allows a process to perform file operations at execution time. Name
Description
File Object
Tap
? < Back Next > Finish Cancel

See *Defining Process Flow Objects* on page 25 for details on defining the objects that are available in Designer.

- 2. In the Name field, type a name for the object, or accept the default.
- 3. In the Description field, type a brief description of the object, or accept the default.

Depending on the object you are adding, this dialog box also includes the following option.

- □ **Tap.** Select the *Tap* check box to enable real-time data capture and reporting. The Tap option is selected by default if you enabled Taps while creating a new process flow. For more information on configuring and using Taps, see the *iWay Enable User's Guide*.
- 4. Click Next.

Depending on the object you choose, there are one or more dialog boxes that present the parameter options for the object.

5. When you are finished defining the object, click Finish.

The new object appears in the workspace. You can drag it to any position in the workspace.

Building Relationships Between Objects

The relationship between two or more objects in a process flow is represented by a color-coded line with an arrow that directs the flow of the XML input document. You define the relationships between objects as you build your process flow. If necessary, you can change the line properties, just as you can the object properties, of an existing process flow.

This section explains how to create a relationship between objects and how to edit existing relationships.

Procedure: How to Build a Relationship Between Objects

To build a relationship between objects:

- 1. Select the object from where the document originates, and then hold the Shift key to select the object that will receive the document.
- 2. Right-click the object that will receive the document, and select *Create Relation* from the menu options.

The Relation Configuration dialog box opens. The following image is an example of this dialog box with an OnCompletion Event.

🔬 Relation Prope	rties	
Relation Conf	iguration	
Use this dialog t using stock or c	o configure a relati ustom event	onship between two objects
Event: OnComp	letion	•
Case of:		÷/* ×
Case	Туре	Description
Description:	A1	
🕐 Don t ask agai	n. Aiways create re	Finish Cancel

3. Select the event for the selected objects from the Event drop-down list.

The default (stock) events that appear depend on the objects you are working with. For instance, the first relationship you build between a Start object and another object allows only the OnCompletion event for that relationship. The lines in the workspace that represent the relationship between two objects are color coded for quick identification. The following list describes the available events and the corresponding color.

- **OnCompletion.** Represented by a blue line. Indicates there are no conditions; always follow the path.
- **OnError.** Represented by a red line. Follow the path if there is a JAVA exception.
- **OnSuccess.** Represented by a green line. Follow the path if there is normal completion.
- **OnCustom.** Represented by a brown line. Allows you to customize the path condition.
- OnFailure. Represented by a maroon line. Follow the path if a coded error occurs, for example, an agent that is coded to issue an error when it is unable to connect to a server.

If you select OnCustom, a list of *Case of* options appears, allowing you to customize the event for a particular condition, such as true or false. The following image shows this dialog box for the line configuration between a Decision Switch object and an End object, and contains an Event drop-down list, and a table with three columns: Case, Type, and Description.

🔬 Relation Propertie	es la					
Relation Configuration ③ At least one case must be selected for the custom event						
Event: OnCustom		•				
Case of:		₽ /* X				
Case	Туре	Description				
OnError	Stock	Error				
OnSuccess	Stock	Success				
OnDefault	Stock	Used for a Decision of type Switch				
🔲 null	Custom	Expression returns with no nodes.				
empty	Custom	Expression retuns with nodes with no value.				
Description:						
Custom						
🔲 Don't ask again. A	Always create relatio	ons using default event type.				
?		Finish Cancel				

4. Click Finish.

A line appears between the two objects to indicate a relationship is established.

Procedure: How to Edit the Relationship Between Objects

If you want to change the properties of an existing relationship:

- 1. Click the line that represents the relationship you want to edit.
- 2. The relation properties are displayed in the Properties tab, in the bottom portion of the console.

The Properties pane provides an Event drop-down list, along with Case of and Description fields.

Properti	es 🛛 🕻	Complex Properties	📮 Console	🛃 Problems			2 -	- 8
Object R	elation							
Relation	Event:	OnCompletion					•	Â
	Case of	5					+ /* ×	Ξ
	Case				Туре	Description		
								-

3. Change the property as needed. (The properties of the relationship object change simultaneously.)

The new line properties are now in place.

Saving a Process Flow

To save a process flow, highlight the process flow node, and select Save from the File menu.

Validating a Process Flow

Validating a process flow determines that the process flow structure is correct. For example, validating a process flow determines that all objects are connected.

To validate a process flow, right-click the Flows node and select *Validate* from the list. A pop-up window appears, as shown in the following image.



Compiling a Process Flow

Compiling a process flow sends the flow to iWay Service Manager to verify that it works in a run-time environment. Compiling as part of creating a process flow is optional, since it is automatically done as part of the test option. The compile option is useful if you are debugging a process flow. It allows you to check the flow using a step-by-step approach.

To compile a process flow, right-click the process flow you want to compile. Select *Integration Tools* from the list, and then select *Compile*, as shown in the image below.

 ▲ a Sample_Project Adapters ⇒ Applications ⇒ Channels ⇒ Ebixes ▲ a P Flows 						Start	File
> Registers > Schemas		New Open		۲			
Transforms		Open With		•		-	
	P	Сору				End	
	Ê	Paste					
	D	Duplicate					
	×	Delete					
		Move					
		Rename					
	2	Import					
	4	Export					
	8	Refresh					
		Validate					
		Run As		•			
		Debug As		•			
		Team		•			
		Compare With		•			
		Replace With		+	L		
		Integration Too	ols	+		Publish	Ctrl+I, F
		Properties				Publish to	Ctrl+I, P
⊡≞ Outline ⊠					1	Deploy	Ctrl+I, D
<u> </u>					-	Create Report	CH L CL
			Design	Test Results	Loid	Compile	Ctri+I, C

Testing a Process Flow

iIT Designer provides a test feature to test the function of a process flow using an input XML document.

The following procedure explains how to use the Test feature.

Procedure: How to Test a Process Flow

1. Click the *Run* button, or right-click the process flow you want to test and select *Run* as, then *Process Flow* from the drop-down list.

The Edit configuration and launch dialog box opens, as shown in the following image.

🔏 Edit Configura	tion							
Edit configure	dit configuration and launch.							
🔇 Provide a vali	d Process Flow location							
Name: Test fl								
Name: Test_fic	5w (1)							
🛛 Main Test	Server Registers Runtime Options							
Process Flow	·]							
	Browse							
Innut Drawn								
	Import							
Server Enviro	nment:							
URL:	http://localhost:9000 👻							
User Name:								
Password:								
🔘 🔘 Use minin	nal runtime configuration with dependencies deploy							
O Use test se	erver							
🔘 🔘 Use existin	ng configuration:							
	Apply Revert							
?	Run Close							

- 2. In the Main tab, browse to select the process flow in the Process Flow field.
- 3. An input document is required for a Test Run. Click the ellipsis button (...) to browse, or click the *Import* button to import an Input Document.
- 4. In the Server Environment area, in the URL field, type the address of the Service Manager where you want to test the process flow in the URL field.

- 5. To access Service Manager, type the User Name and Password in the corresponding fields.
- 6. Choose between the following options:
 - **Use minimal runtime configuration with dependencies deployed from the registry.**
 - Use test server.
 - **Use existing configuration.** Click the *Get Configurations* button to access existing configurations.
- 7. Click the Test Server tab. The dialog box is displayed, as shown in the following image.

Zedit Configuration									
Edit configuration and launch.									
Name: Test_flow (1)									
Main Test Server Registers Runtime Options									
Application:									
Browse									
Template:									
Browse									
Apply Revert									
Run Close									

8. Browse to select the Application and Template if required.

9. Click the Registers tab. The dialog box is displayed, as shown in the image below.

🛓 Edit C	Edit Configuration						
Edit configuration and launch.							
Name:	Test_flow	(1)					
Main	Test Server	Registers Ru	ntime Options				
Nam	ne	Туре	Value	Usage	Description		
•			III		Þ		
				Apply	Revert		
?				Run	Close		

10. You can enter SREG values to be used in the process flow, which will override the values set through either the iWay Service Manager console, server configuration, or Register Sets.

11. Click the *Runtime Options* tab. The dialog box is displayed, as shown in the image below.

🔏 Edit Configuration							
Edit configuration and launch.							
Name: Test_flow (1)							
Main	Test Server	Registers	Runtime Options				
Process Object Debugging All - Trace all active objects Select - Trace all debug enabled objects None - Tracing is disabled 							
Server Side Tracing Image: Server Side Tracing							
Transaction Control							
Timeout for the response from the server during Test Run 30							
			Apply		Revert		
?			Run		Close	•	

12. Select the Process Object Debugging option, as follows:

All. Traces all active objects.

Select. Traces all debug enabled objects.

None. Disables tracing.

13. Select the server side tracing, as required.

Debug. Reports progress traces of the test.

Deep Debug. Reports additional traces and details about errors.

14. Choose whether or not to set the Transaction Control option, Commit Transaction.

This determines whether or not the work scheduled in the process flow is actually performed during the test, for example, inputting records to a database.

- 15. Input the timeout for the response from the server during Test Run. The default is 30 seconds.
- 16. Click Apply. Click Run.

The test results appear in the Console tab of the Properties pane.

Publishing a Process Flow

Publishing a process flow makes the process flow available to either a system or to a channel that was created in Service Manager. A process flow published to a system can run as a global document or be used by an iWay adapter to create a web service.

You can only publish valid and compiled process flows. A single process flow can be published to multiple listeners and systems. See *Validating a Process Flow* on page 134 for the validation procedure and *Compiling a Process Flow* on page 135 for information on compiling a process flow.

There are three options to publish a process flow:

- **Publish.** Opens a series of publication setup dialog boxes through which you direct the publication.
- Default Publish. Uses the default Service Manager instance, but allows you to select new publishing locations (system and listener(s)).
- **Republish.** Publishes the process flow using the previous settings. This option is useful if you edit a process flow and want to republish it with the previous publication settings.

Designer provides a tool called Publication Management that allows you to view the process flows published to a particular Service Manager configuration. Through this tool, you can also unpublish any process flow currently published to a Service Manager configuration. This section provides the procedures to publish and unpublish a process flow.

Procedure: How to Publish a Process Flow

To publish a process flow:

1. Right-click the process flow you want to publish, and select *Integration Tools* from the list. Then select, *Publish*.

The Publishing Resource Wizard dialog box opens. An example of this dialog box is shown in the following image.

A Publish Resource Wizard					
Server Se i Server UF	lection RL format: http://{serverName}:{SOAP port}				
Server URL: User Name Password	http://				
?	Finish Cancel				

- 2. From the Server URL drop-down list, choose the Service Manager server to which you want to publish the flow.
- 3. In the User Name field, type a valid user name for the Service Manager server.
- 4. In the Password field, type a valid password for the Service Manager server.
- 5. Click Finish.
- 6. Click Next.

The Publish To dialog box opens, as shown in the following image.

- 7. Select *iWay Registry* or *Configuration ID* as the destination to which to publish the process flow.
- 8. Click Next.

The Select Publication Location(s) dialog box opens, as shown in the following image. This dialog box shows the publication locations (systems and listeners) for the selected Service Manager and configuration. The locations are listed in a table with three columns; Name, Type, and Description.

You can publish a flow to a system, a configured listener, and (if defined) a global document. You can also choose to overwrite any flows that are currently published.

- 9. In the Name column, select a system and/or the listener(s) to which you want to publish the process flow. To select all items in the Name column, click *Check All*.
- 10. Select, or deselect Overwrite existing published process(es) and, Refresh listener after publication, as needed.
- 11. Click Finish.

The Message Log pane displays the publishing status.

An example of the status display when publishing to a system is shown in the following image.



Note: You can publish only one process flow to a listener.

Procedure: How to Use the Default Publish Option

This option uses the default publication server (Service Manager) and configuration, but allows you to select the publication locations.

1. In the Project tab of the Navigator pane, right-click the process flow you want to publish, and select *Default Publish* from the drop-down list.

The Select Publication Location(s) dialog box opens, as shown in the following image. This dialog box shows the publication locations (system and listener(s)) for the default Service Manager and configuration. The publication locations are listed in a table with three fields: Name, Type, and Description.

- 2. In the Name column, select the system and/or listeners to which you want to publish the process flow. To select all items in the Name column, click *Check All*.
- 3. Select, or deselect Overwrite existing published process(es) and, Refresh listener after publication, as needed.
- 4. Click Finish.

The Message Log pane displays the publishing status.

Procedure: How to Republish a Process Flow

To republish a process flow:

Right-click the process flow you want to republish and select *Republish* from the drop-down list.

The publication process uses the previous publication settings (publication server and locations) to publish the process flow. The publication results appear in the Message Log pane.

For an example of publication results, see *How to Publish a Process Flow* on page 141.

Procedure: How to Unpublish a Process Flow

You can only unpublish a process flow from a run-time configuration. To remove a process flow from a Registry, use the iWay Service Manager console, as described in the *iWay Service Manager User's Guide*.

Use the following procedure to view the published process flows on a particular publication server and to unpublish a process flow that is currently published.

1. Select *Publication Management* from the Tools menu or click the Publication Management icon located in the toolbar.

The Unpublishing Process(es) - Locate a Publication Server dialog box opens, as shown in the following image.

- 2. From the Server URL drop-down list, choose the Service Manager server you want to access.
- 3. In the Login pane, type a valid user name and password in the corresponding fields for the Service Manager server.
- 4. Click Next.

The Select Server Configuration dialog box opens, as shown in the following image.

- 5. From the Configuration ID drop-down list, choose the configuration you want to access.
- 6. Click Next.

The Unpublish Operation dialog box opens. An example of this dialog box is shown in the following image. The processes are listed in a table with three fields: Name, Type, and Description.

- 7. In the Name column, select the process flow(s) you want to unpublish. If you want to select all process flows on this configuration, click *Check All*.
- 8. Click Finish.
The Publication Management tool closes and the results appear in the Message Log pane. The following image shows an example of the results for a process flow named SWIFT_ROUTER..



The process flow has been removed from the publication server.

Exporting a Process Flow

A process flow can be exported as an XML file where it is accessible by another Service Manager or a version control system, such as Microsoft Visual SourceSafe.

Procedure: How to Export a Process Flow

To export a process flow:

1. In the Navigator pane, right-click the process flow that you want to export, and select *Export* from the drop-down list.

The standard Windows Save As window opens.

- 2. Choose the location where you want the XML file to reside, and choose a name for the exported process flow file. Choose *XML* as the file type.
- 3. Click Save.

The process flow is saved as an XML document in the designated directory.

Adding an Adapter to iIT Designer

The Adapter object is a configured connection to a particular instance of a back-end system, for example, the MySAP adapter. You can only create an adapter object using adapters that have a currently configured connection (target) to a back-end system or a database.

Adapter targets are configured in the iWay Explorer. Adding an adapter consists of the following steps:

- 1. Creating a new iWay Resource
- 2. Selecting the adapters to be displayed in the Adapter Explorer
- 3. Connecting to an Enterprise Information System
- 4. Creating a Registry

5. Exporting an Adapter Target to an Integration Project

Creating a new iWay Resource

The following procedure describes how to create a new Resource in iWay Explorer

Procedure: How to Create a new iWay Resource

To create an iWay Resource in iWay Explorer:

- 1. Start iIT Designer and click the *iWay Explorer* tab in the Navigator pane.
- Right-click in the Navigator pane and select *New iWay Resource* from the drop-down list. The New iWay Connection dialog box opens, as shown in the following image.

🔌 Resource Selection Dialog	a gratager	x
New iWay Connection Select a resource type to crea	ate.	
Туре	Description	Versi
iWay Configuration	Create a connection to an adapter run-time ins	
?	Next > Car	ncel

- 3. Select iWay Configuration and click Next.
- 4. In the Configuration Alias field, type a name for the resource, for example, SampleConfig.
- 5. Select either an HTTP Connection or a JCA connection. Click Next.

6. The Connection Information dialog box opens, as shown in the following image.

Add iWay Configuration				
Enter Connection Information				
Provide the server's con console port.	nection string, SOAP	port and the iSM		
Connection String:	http://localhost			
User Name:	iway]		
Password:	••••]		
SOAP Port/Endpoint:	9000]		
Console Port/Endpoint:	9999]		
		Presets	h	
		Local Connection		
		Servlet		
? < Back	Next >	Finish Cancel		

- 7. Enter information in all of the following fields:
 - Connection String
 - User Name
 - Password
 - □ SOAP Port/Endpoint
 - □ Console Port/Endpoint
- 8. Click Finish.

The new iWay Resource appears in the Navigator pane. Expand the new resource to reveal the Adapters, Services, Events, Applications, and Registry nodes. The default adapters under the Adapters node are iWay and RDBMS. The default adapter under the Events node is RDBMS. The following image shows an expanded resource named SampleConfig.



Selecting the Adapter

Procedure: How to Select an Available Adapter

The following procedure describes how to filter the list of adapters that are displayed in the Adapters node.

1. In the Navigator pane, right-click the resource and right-click. Select *Connect* in the drop down list to connect to the iSM server.

Once connected, all the nodes have green arrow icons to show that a connection has been established.

2. Select the Adapter node and right-click. Select *Edit* from the drop-down list.

The Adapter Selection dialog box opens. An example of the Adapter Selection page is displayed in the following image.

🔬 Edit Adapters	a cancelled		
Adapter Selection Page			
Select which adapters s	hould be displayed in the Adapter Explorer.		
Adapter Name	Description	Select All	
HL7	1.0	Colort Name	
IMS	Supports IMS access via IMS Connect in IMS V7 and up, and the CF	Select None	
👿 iWay	Supports iWay v6.0. iWay Adapter Framework v1.0.		
Java	Supports Plain Old Java Objects (POJOs). iWay Adapter Framework		
LDAP	1		
LogListener	1.0	_	
MSCRM2007	0.5	=	
MUMPS	Supports MUMPS access via TCP/IP.		
MySAP	mySAP solutions, ECC 5.0 and 6.0 (and other SAP Components tha		
RDBMS	Supports JDBC API v. 3.0 compliant drivers. Adapter Framework ve		
Salesforce	Salesforce Adapter 1.0		
Telnet	Supports NVT, TN3270 and TN5250 emulations		
?	Finish	Cancel	

Note: Only the adapters with the corresponding jar files, present in your program files folder, appear on this list.

- 3. Select the adapter(s) in the Available adapters list that you want to add to the display.
- 4. Click Finish.

The selected adapter(s) appears under the Adapters node.

Connecting to an Enterprise Information System

An iWay Explorer target is the connection point to a specific Enterprise Information System (EIS), such as SAP or Oracle. Through a target, you can browse EIS metadata, generate XML schemas, and create web services.

This section explains how to create a new target, connect to a target, and disconnect from a target. In addition, it explains how to edit and delete an existing target.

About Targets

A target is a connection to an EIS. When you create a target, you define the connection parameters, such as the system name and password. The target is saved in iWay Explorer and available for you to use at any time. You will connect to the target to perform iWay Explorer tasks, and then disconnect when the work session is done. Disconnecting from the application system drops the connection, but the target node remains intact. Although you can maintain multiple open connections, we recommend that you close connections that are not in use.

You also have the option to edit the connection parameters of an existing target and to delete a target that is no longer needed.

Procedure: How to Create a Target

To create a new target:

1. In the Navigator pane, right-click the adapter node you want to work with, for example, MySAP, shown in the following image.



2. From the drop-down list, select Add Target.

The New Target properties dialog box opens, as shown in the following image. The Type drop-down list provides connection options that are specific for the adapter target you are creating. The following image shows the New Target dialog box for an SAP adapter.

🔬 Add Target			
Generic Target Properties Please enter the generic properties associated with the new target.			
Name: Description:	SAP_Target		
Type: Application Server Connect to target upon wizard completion.			
(?) < Back Next > Finish Cancel			

- 3. In the Name field, type a descriptive name for the target, for example, SAP_Target.
- 4. In the Description field, optionally type a brief description of the target.
- 5. From the *Type* drop-down list, select the target type.
- 6. Click Next.

The New Target - Configuration dialog box opens. The configuration parameters that appear are specific to the adapter target you are creating. The following image is an example of a new target configuration dialog box for an SAP adapter.

🔬 Add Ta	rget					
Applica Please e	t ion Se r	ver Targ	et Prope ociated wi	erties th the new targ	get.	₽
User	System	Advanced	Security			
Client						
User						
Passwo	ord					
Langua	ige					
EN						
Codep	age					
Auther Passw	ntication m	node +				
?	< Bac	k	Next >	Finish		Cancel

7. Click a field to activate it, and type a value for that target parameter.

A red field name indicates that the parameter is required.

Note: Depending on the adapter target you are creating, there can be more than one Configuration tab. For example, SAP presents separate User configuration and System configuration dialog boxes. If you are creating a target for the GeoLoad adapter, there is only one configuration tab. See the adapter specific user guide for details on the configuration parameters.

8. When configuration parameter entry is complete, click *Finish*.

The target appears in the Navigator pane, under the adapter node for which it was created. For example, the SAP_Target appears under the SAP adapter node in the following image.



Procedure: How to Connect to a Target

To connect to a target:

- 1. In the Navigator pane, right-click the target to which you want to connect.
- 2. Select Connect from the drop-down list.

The Target Connection widow opens. The Target Connection dialog box for an SAP target is shown in the following image.

A Target Connection Dialog				
Application Server Connection Properties Please enter the connection parameters for this target.				
User System Advanced Security				
Client				
800				
User				
iwayqa				
Password				

Language				
EN				
Codepage				
Authentication mode				
Finish Cancel				

3. Type the password for the target, and click *Finish*.

The application system objects appear under the target node.

Procedure: How to Disconnect From a Target

To disconnect to a target:

- 1. In the Navigator pane, right-click the target from which you want to disconnect.
- 2. Select Disconnect from the drop-down list.

The expanded target node closes and is no longer connected.

Procedure: How to Edit a Target

To change the configuration of a target:

1. In the Navigator pane, right-click the target that you want to edit.

Note: You must be disconnected from the target to access the Edit feature.

The following image shows the resulting drop-down list of a target named SAP_Target.



2. Select Edit Target from the drop-down list.

The Edit Target dialog box opens. This dialog box displays the connection parameters that were defined when the target was created. Therefore, this dialog box reflects the EIS parameters of the selected target.

For target editing details, see the specific adapter user's guide.

3. Make the required changes, and click *Finish*.

The target is updated with the new connection information.

Procedure: How to Delete a Target

To delete a target:

1. In the Navigator pane, right-click the target from which you want to delete. The following image shows the resulting drop-down list for a target named SAP_Target.



2. Select Delete from the drop-down list.

A confirmation window opens. The following image is an example of a confirmation window.

nd Confi	irm Delete		×
?	Delete Target: SAP_Target?		
		ОК	Cancel

3. Click OK.

The selected target node is removed from target list in the Navigator pane.

Creating a Registry

To create a registry:

Procedure: How to create a registry

1. Select the target for which you want to create a registry and right- click.

2. Select *Create Registry* from the drop down list. The following image is an example to create a registry from the SAP_Target



3. A confirmation window opens. The following image is an example of a confirmation window.



4. The new target now appears in the Registry node. Expand Components and Adapters to see the newly added target, as illustrated in the following image.



Exporting the Adapter Target

An adapter must be exported to the server so you will be able to use it in a process flow. To export the adapter target to an integration project:

Procedure: How to Export an adapter target

1. Right-click the adapter target in the Registry node. Select *Export* from the drop-down list.

2. The Export Dependencies dialog box opens as shown in the following image.

🔬 Export Resource	
Export Dependencies Project name must be specified.	
Project:	
? Finish	Cancel

3. Click the ellipsis (...) button to browse to the project of choice.

A dialog box with the tree nodes and its dependencies opens as shown in the example below.

Export Resource			
Export Dependencies			
Select nodes that you want to exports from the registry.			
Project: Router_Example			
Below is a tree of node(s) and its dependencies you are exported automatically (you cannot uncheck these nodes you want to export them by checking checkbox next to n	xporting fro s). Nodes th ode you wa		
Note that all component will be exported to their default	folders in tl		
Only show existing components (components that existing components (components that existing components)	st both in p		
Name	Туре		
SAP_Target	adapter		
•			
Select All Deselect All			
? Finish C	ancel		

4. Some nodes are checked. These nodes do not exist in your project and will be exported automatically. Nodes that are not checked by default, exist in your project. You can choose whether you want to re-export them.

Select the nodes you want to export and click Finish.

The adapter target now appears in the Integration Explorer tab under the Adapters node. An example showing the SAP_Target is illustrated in the image below.



You can now drag and drop the adapter into the workspace to use in a process flow.

Using Variables and XPath

Using XPath in Designer

XPath is a language that is mainly used to address parts of an XML document based on a path through the logical structure or hierarchy of the document. In Designer, XPath defines the path to the value or values in an XML document to use for a property or parameter. In Designer, XPath always starts from the root of the document.

Note: The XPath syntax used in Designer is based on a subset of the XPATH 1.0 specification, found at http://www.w3.org/TR/xpath.html.

The default axis is "child". Currently, this is the only axis available in Designer.

The XPath Specification

iIT Designer uses XPath in conjunction with iWay Service Manager.

Important: All iWay functions are available in iIT Designer. and can be used in conjunction with XPath statements. For more information on iWay functions, see the *iWay Service Manager User's Guide*.

XPath populates values with data from Service Manager in one of three ways:

Decision node switch: Designer routes a document based on the value received from XPath.

Decision node test: Creates a True/False test against an XPath statement.

D Configuration parameter: The value of the parameter is the value of the XPath.

The following are examples of location path parameters that use abbreviated syntax taken from section 2.5 of the XPath specification. This specification is found at http://www.w3.org/TR/xpath.html#path-abbrev.

para

Selects the para element children of the current position.

*

Selects all element children of the current position.

text()

Selects all text node children of the current position.

@name

Selects the name attribute of the current position.

para[1]

Selects the first para child of the current position.

para[last()]

Selects the last para child of the current position.

*/para

Selects all para grandchildren of the context node.

/root//para

Selects the para element descendants of the root element.

//user

Selects all the user descendants of the document root.

//olist/item

Selects all the item elements that have an olist parent.

Selects the current position.

.//para

Selects the para element descendants of the current position.

•••

Selects the parent of the current position.

../@lang

Selects the lang attribute of the parent of the current position.

para[@type="warning"]

Selects all para children of the context node that have a type attribute with value warning.

```
para[@type="warning"][5]
```

Selects the fifth para child of the current position that has a type attribute with value warning.

para[5][@type="warning"]

Selects the fifth para child of the context node if that child has a type attribute with value warning.

chapter[title="Introduction"]

Selects the chapter children of the current position that have one or more title children with string-value equal to Introduction.

chapter[title]

Selects the chapter children of the current position that have one or more title children.

Building XPath Statements

Designer provides the XPath Builder tool as an easy method to construct XPath statements that can then be used in object definitions. This section explains how to build XPath statements with XPATH Builder.

The Designer XPATH Builder tool guides you through the process of creating an XPath statement. The major steps are:

Load an instance of an XML file.

Select the related elements and attributes that will make up the XPath statement.

- □ If needed, add methods and operators.
- Validate the XPath statement against a test document.

Approve to set the service XPath.

Procedure: How to Use XPATH Builder

To use XPATH Builder:

1. Click the Launch XPATH Builder button on the ribbon toolbar menu.

The XPATH Builder - Load a document instance dialog box opens, as shown in the following image.

🔏 XPATH Builder 💿 💌			
Load a document instance			
Load a sample of the document that will be used in the flow in order to browse its schema and build an XPATH.			
This wizard will assist in creating a valid XPATH for use with an object. By creating an XPATH and running it againsts the sever you can see the resuts that will be returned to your object property.			
Load Import Reload			
Previously loaded files			
Clear history			
*			
XML Source: {no document loaded}			

2. Click *Load...* to load a sample XML document that will be used in the process flow. From this document, you can browse the schema and build an XPath.

The Select an XML file dialog box appears.

3. Browse to the XML document you want to load and click Open.

The XML file appears in the display pane of XPATH Builder.

4. Click Next.

The Build an XPath dialog box opens, as shown in the following image. Double click the element, attribute or attribute value you wish include in the statement.

🔬 XPATH Builder		
Build an XPath		
Browse the schema and t input to the flow.	find the elements you wisł	to use as
Double click the element	t, attribute or attribute valu	ie you wish to use to build tł
XML Data Type	XPath Methods	XPath Operands
e SWIFT2	count(<node-set></node-set>	=
e USER	local-name(<node< th=""><th>< =</th></node<>	< =
	name()	<=
	position()	>=
	۲ (III)	and 👻
? Kack	Next > Fi	nish Cancel

5. Expand the schema elements located in the left pane. These elements and attributes are those found in the XML document that you loaded earlier. Double-click the item you want to add to the XPath statement.

The item appears in the XPath Expression field. The following image is an example of this dialog box with *count* selected in the center pane and the related expression appearing in the XPath Expression field. The values and occurrences of this element appear in the Description pane, in the bottom of the dialog box.

🔬 XPATH Builder				
Build an XPath Browse the schema and find the elements you wish to use as input to the flow.				
Double click the element, attrib	ute or attribute value you wish to use to buil	d the XPath statement		
XML Data Type	XPath Methods	XPath Operands		
e SWIFT2 e USER	namespace-uri(<node-set>) position() concat(<string>, <string>) starts-with(<string>, <string>) starts-with(<string>, <string>) string(<object>) substring-after(<string>, <string< td=""> substring-before(<string>, <string< td=""> not(<boolean>)</boolean></string<></string></string<></string></object></string></string></string></string></string></string></node-set>	= != < < > >= and or		
count(<node-set>)//SWIET2/</node-set>	ISERcontains(<string>_<string>)</string></string>	*		
C Description	oscholmanis(sannys, sannys)	Execute		
Description:Returns true if the first argument string contains the second argument string, and otherwise returns false Example: XPATH(//a/b/c[contains(@cattrib.'c')]) (returns all elements <c> if there attribute contains string 'c')</c>				
?	< Back Next >	Finish Cancel		

- 6. Continue to add the elements and attributes needed to build the statement.
- 7. Click *Execute* to validate the XPath statement on the server.

The results appear on the top of the XPath builder dialog box.

8. Click Finish.

The XPath statement is now available from the various object parameter selections.

Adding an Object to a Process Flow from an iSM Registry

To add a pre-configured object from an iSM registry to a process flow

Procedure: How to Add a pre-configured object from an iSM registry to a process flow

- 1. Double-click the required process flow and make sure it is open for editing in the workspace..
- 2. Click on the iWay Explorer tab. Right-click on the Registry node to connect to an iSM Registry, as shown in the following example.



3. When the registry is connected, you can expand the node to find the Conduits, Components and Variables nodes. Find the object that you wish to use in your process flow. Left click, hold, and drag the object to your open flow in the workspace area. In the following image, an SAP_Target object has been dragged and added to the process flow in the workspace.



4. Once your release your mouse an export dialog will open as shown in the following image.

Export Resource		
Export Dependencies		
Select nodes that you want to exp	orts from the registry.	
Project: Router_Example		
exported automatically (you can you want to export them by chec Note that all component will be e	Inturcheck these nodes). Nodes that are not c king checkbox next to node you want re-expor exported to their default folders in the project. Its (components that exist both in project and	hecked by default, exist in your project. You can choose whether t Default folders can be customized in <u>Export Folders</u> preference pag on the server)
Name	Туре	Description
SAP_Target	adapter	
•	m	
Select All Deselect A	All	

- 5. Select the resource associated with the object from the registry to export and click *Finish*. The resource is now exported to the default export folder for the resource type.
- 6. The new object is now added to the flow and ready to use.

Working with Customizations

Adding a Customization JAR

To use a custom service or non-default service that is packaged in a java jar file and make that service available to a process flow, add the jar to the Customizations classpath.

Procedure: How to Add a Customization JAR

To add a customization JAR:

- 1. Right-click the project that you wish to use, and select *Properties* from the drop-down list.
- 2. Expand the iWay Integration category.
- 3. Select *Customizations*. Select the Customizations tab as shown in the following example.

Properties for Router_Example	The second se	
type filter text	Customizations	$\Leftrightarrow \bullet \bullet \bullet \bullet \bullet$
Resource Builders ■ iWay Integration Application Customizations Default iSM Target Vers Export Folders IFL Binding Properties Process Flow ▶ Transformer Project References Refactoring History Run/Debug Settings Task Tags	Additional Components Customizations for Router_Example:	Add JARs Add JARs Add External JARs Add Class Folder Remove
Validation		
<	Restore D	efaults Apply Cancel

- 4. If the JAR containing the custom service is in your workspace click the *Add JARs...* button. If the JAR is located in an external location, click the *Add External JARs...* button.
- 5. Once you have selected your jar click Apply. Click OK.

The selected service is now available to use in your process flow.

Adding Additional Services and Components

To use a custom service or non-default service that is installed on a particular instance of an iSM server, and to make that service available to a process flow, add it to the list of components for that server version.

Procedure: How to Add Additional Services and Components

To add Additional Services and Components:

- 1. Right click on the project that you wish to use the service and select *Properties* from the drop-down list.
- 2. Expand the iWay Integration category and select *Customizations* as shown in the following example.

Properties for Router_Example	
	Customizations $\Leftrightarrow \bullet \bullet \bullet \bullet$
Resource Builders Jay Integration Application Customizations Default iSM TargetWers Export Folders IFL Binding Properties Process Flow Transformer Project References Refactoring History Run/Debug Settings Task Tags Validation	Additional Components Customizations Dependencies The default list of component types that are available in process flow designer is pre-installed into iIT. Here you can manage component types that may have been added to the server as extensions. Image: Component types that may have been added to the server as extensions. Server version: Image: Component types that may have been added to the server as extensions. Image: Component types that may have been added to the server as extensions. Server version: Image: Component types that may have been added to the server as extensions. Image: Component types that may have been added to the server as extensions.
<	Restore Defaults Apply
(?)	OK Cancel

3. Select the Additional Components tab.

4. Click the *Add* button. The Server Selection dialog box opens as shown in the following image.

🔬 Additiona	I Components Wizard
Server Se	lection {L format: http://{serverName}:{SOAP port}
Server URL: User Name Password	http://
?	< Back Next > Finish Cancel

5. Select a Server URL from the drop-down list or, enter a hostname and the soap port of the target server.

6. Click Next. The Additional Components dialog box opens as shown in the following image.

🔬 Additional Components Wizard
Additional Components Select the additional components you would like to add.
 Emitters Services
Seck Next > Finish Cancel

- 7. From the tree select the component you would like to add.
- 8. Click Finish.
- 9. Once the dialog box closes, click *OK* to save and close the Properties dialog box.

If you drag a service object to a process flow in the project to which you just added the component, the new component will be available in the tree.



Tutorial

This section provides an example of a process flow that routes a document based on the document type.

In this chapter:

An Example Process Flow

An Example Process Flow

Provides a step-by-step example of creating a process flow.

About This Example

The function of the process flow in this example, is to route two specific XML document types, identified by the root element, to specific directories on your system. To accomplish this, we must use multiple Decision Test objects using a True/False test. A single Decision Test Object testing for multiple value switching cannot handle testing for a document tag name. Once the XML documents are identified, they will be routed through the process flow to File objects that are configured to write the XML document contents to a file in a designated directory.



The following image shows the process flow we will create. It includes a Start object, two Decision Test objects, each with a true and false path, three File objects, and an End object.

Create the Input XML Documents

Normally, you will have multiple XML documents as input to a process flow. For illustrative purposes, we will use just two documents as input to our example process flow.

Use a text editor, such as Notepad, to create the following two XML documents.

□ Name the following file Route1:

```
<SWIFT1>Route1
<USER>Atlas</USER>
</SWIFT1>
```

□ Name the following file Route2:

```
<SWIFT2>Route2
<USER>USA</USER>
</SWIFT2>
```

Creating the Process Flow

This section walks you through the steps to create the example process flow. You will begin by creating a new project and process flow, and continue by adding and defining each object that makes up the flow. As you add objects, you will define the relationship between them. Finally, you will validate and test the flow using the XML documents you created as input to the process flow.

Procedure: How to Create the Project and Process Flow

To create the Router_Example project and the SWIFT_ROUTER process flow:

- From the File menu, select New and select Integration Project from the drop-down list. The Integration Project Configuration dialog box opens.
- 2. In the Project name field, type *Router_Example*.
- 3. Click Finish.

The Router_Example project node appears in the Navigation Pane.

- 4. Expand the Router_Example project node to expose the project elements.
- 5. Right-click the *Flows* folder, select *New*, and then select *Process Flow*.

The New Process Flow properties dialog box opens, as shown in the following image.

🔬 New Process Flow Wizard	
General Properties Other Name field is required	l
Project Folder:	/Router_Example/Flows Browse
Name:	
Description:	¢
Target Server Version:	7.0.0 🔻
	🕅 Enable taps
?	Finish Cancel

6. In the Name field, type SWIFT_ROUTER as the process flow name.

7. Click Finish.

The new SWIFT_ROUTER process flow node appears under the Flows folder, and the workspace displays a Start object, as shown in the following image.

🔬 I 🖾 🛋 L 🧵 i 🖓 🗖	🐯 SWIFT_ROUTER 🛛	
수 수 🗟 📄 🔄 🎽		😳 Palette 🔉 🔈
🔐 Router_Example		Select
🗁 Adapters	Start	[]] Marquee
🔁 Applications		
🔁 Channels		n-n-n Relation
🔁 Ebixes		⊷⊷ Loop 🗧
Flows		C Paris Objects
SWIFT_ROUTER		Basic Objects 🖏
Registers		<u>la</u> End
🔁 Schemas		h Tart
🗁 Transforms		💡 Control Obje 🚸
🗁 XML		🗮 Decision Switch 😓
	Design Test Results	1

You are ready to build the SWIFT_ROUTER process flow.

Procedure: How to Create the First Decision Test Object

To create the first Decision Test object:

- 1. Drag the Decision Test object from the toolbar to the workspace. The New Decision Test Object dialog box opens.
- 2. Name the Decision Test object *IsType1*, and click *Next*.

The Test Operands dialog box opens, as shown in the following image.

New Decision Test Object	×
Test Operands Ø Provide Operand One	
Test Parameters Operands can be literals, variables, or functions (XPATH(), LDAP(), or F Operand One Operation Equal To Operand Two	ile())
Seck Next > Finish Car	ncel

3. We will create an XPath statement for Operand One of the test parameters. Select *Create XPath statement* from the Operand One drop-down list.

The XPath Builder tool opens.

4. Click Import..., and choose the Route1 XML document that you created earlier.

The Select Destination dialog box opens.

The Route1 XML document contents appear in the XPath Builder dialog box, as shown in the following image.

XPATH Builder
Load a document instance Load a sample of the document that will be used in the flow in order to browse its schema and build an XPATH.
This wizard will assist in creating a valid XPATH for use with an object. By creating an XPATH and running it againsts the sever you can see the resuts that will be returned to your object property.
Load Import Reload
Clear history
<swift1>Route1 <user>Atlas</user> </swift1>
TML Source: /Router_Example/Flows/route1.xml
Cancel < Back

5. Click Next.

The XPath Builder - Build an XPath dialog box opens.
6. In the left pane of XPath Builder, double-click the Swift1element. This adds the //SWIFT1 element to the XPath Expression field, as shown in the following image.

🔬 XPATH Builder				
Build an XPath				
Browse the schema and find the elements you wish to use as input to the flow.				
Double click the element, att	ibute or attribute value you	wish to use to bu	ld the XPath st	atement
XML Data Type	XPath Methods	XPat	h Operands	
Image: Swift 1 Image: Swift 1 Image: Swift 1 Image: Swift 1 Image: Image: Swift 1 Image: Swift 1 Image: Swift 1 Image: Image: Image: Swift 1 Image: Swift 1 Image: Swift 1 Image: Image: Image: Swift 1 Image: Swift 1 Image: Swift 1 Image: Image: Image: Swift 1 Image: Swift 1 Image: Swift 1 Image: Image: Image: Image: Swift 1 Image: Swift 1 Image: Swift 1 Image: Image: Image: Image: Image: Swift 1 Image: Swift 1 Image: Swift 1				
- XML Details				Execute
Values		Occurences	Path	
Route1 Atlas		0	SWIFT1	
•				•
?	< Back Nex	t> Fi	nish	Cancel

7. Click Finish.

You are returned to the Test Operands dialog box, which now contains the XPath expression you just created, as shown in the following image.

🔬 New Decision	Test Object				
Test Operan	Test Operands				
Enter a literal, v evaluate.	Enter a literal, variable, or function and an operation for the test to evaluate.				
- Test Paramete	rs				
Operands can	be literals, variables, or functions (XPATH(), LDAP(), or FILE())				
Operand One	XPATH(//SWIFT1)				
Operation	Is Not Null				
Operand Two					
? < Back Next > Finish Cancel					

8. From the Operation drop-down list, select Is Not Null.

These parameters allow any XML input documents with a <SWIFT1> root tag to be captured.

9. Click Next.

The Value Occurrences dialog box opens, as shown in the following image.

🔬 New Decision Tes	t Object		
Value Occurrences Select the number of occurrences you wish to be returned.			
Test Occurrences Select 'Unique' if only one value may exist in the document, 'Any' if at least one must exist, and 'All' if every value must exist.			
🔘 Unique	Any	© All	
? Kack	Next >	Finish	Cancel

10. Select Any and click Finish.



The Decision Test object appears in the workspace, as shown in the following image.

Procedure: How to Define the Relationship Between Start and IsType1

To define the relationship between the Start and IsType1 objects:

- 1. Select the *Start* object, press the Shift key, and then select the *IsType1* object.
- 2. Right-click *IsType1*, and select *Create Relation* from the drop-down list.

The Configuration dialog box opens, as shown in the following image.

🔬 Relation Prop	perties		
Relation Configuration Use this dialog to configure a relationship between two objects using stock or custom event			
Event: OnCom	pletion	•	
Case of:			
Case	Туре	Description	
Description:			
Don't ask again. Always create relations using default event type.			
?		<u>F</u> inish Cancel	

- 3. Since this is the first relationship in the process flow, you can only choose OnCompletion as the Event. This means the document will always follow this path.
- 4. Click Finish.

A line appears between Start and IsType1, showing that a relationship has been established. This line is blue to indicate an OnCompletion event.

Procedure: How to Route the Document Based on Decision Test Result

We want all XML documents captured by IsType1 that contain the <SWIFT1> element, sent to a File object named File_1. In addition, we want the File_1 object to write its output document to a disk.

To create the File_1 object:

1. Drag the File Execution object from the toolbar to the workspace.

The New File Object dialog box opens, as shown in the following image.

New File Object
Name and Description Provide object name and description
A File object allows a process to perform file operations at execution time. Name File_1
Description
File Object
Tap
< Back Next > Finish Cancel

- 2. In the Name field, type *File_1*, and leave the Description as File object.
- 3. Click Next.

The File Type dialog box opens, as shown in the following image.

🔬 New File Object				
File Object Type Select the type for the File Object object you wish to create				
Defined File Object services in the current configuration are listed Type File Emit Agent {com.ibi.agents.XDFileEmitAgent}				
Description Emits a document to a physical file				
A Back Next > Finish Cancel				

- 4. From the Type drop-down list, select *File Emit Agent*. This tells the File_1 object to write the incoming document to the disk.
- 5. Click Next.

The New File Object - Properties dialog box opens, as shown in the following image.

🔬 New File Object			
Object Properties			
Provide object p	roperties		
▼ Main			
Source of Data	Source of data to write. If omitted, document will be used, else specify a data source location via xpath() function or any other function		
Target	The target output directory		
Directory	C:\Temp\Out 👻		
File Pattern	The output file name, which can contain a '*' which gets expanded to a fine timestamp		
	TYPE_1_*.xml		
Avoid	Should any preemitter be avoided?		
Preemitter	true 👻		
Return 'status': status document will be the out document. 'input': in document will become the out document. 'swap': as input, but replace written data in the source nodes with the file name to which the data was written.			
	status 👻		

- a. In the Target Directory field, type a path to the directory where the file will be written. In this example, we direct the file to C:\Temp\Out.
- b. In the File Pattern field, type *TYPE_1_*.xml*.
- c. In the Avoid Preemitter field, select *true*. This ignores any assigned preemitters.
- d. In the Return field, select *status* to return a status document.

6. Click Finish.

The File_1 object appears in the workspace, as shown in the following image.



Procedure: How to Define the Relationship Between IsType1 and File_1

Build a relationship between the IsType1 Decision Test object and the File_1 File object, so that an XML document coming into IsType1 that has a <SWIFT1> element, is sent onto the File_1 object. To build this relationship:

- 1. Select the *IsType1* object, press the Shift key, and then select the *File_1* object.
- 2. Right-click *File_1* and select *Create Relation* from the drop-down list.

The Relation Configuration dialog box opens.

- 3. Select OnCustom from the Event drop-down list.
- 4. Select *true* as the case for the event.

The following image shows the Relation Configuration dialog box with the OnCustom event and true case selections.

🔬 Relation Prope	rties		
Relation Configuration			
Use this dialog to objects using sto	o configure a rela ock or custom ev	ationship between two ent	
Event: OnCustor	m	•	
Case of:			
Case	Туре	Description	
OnSucce	Stock	Success	
OnDefault	Stock	Used for a Decision of type S	
V true	Custom		
- falce	Custom	•	
Description:			
? Finish Cancel			

5. Click Finish.

A line appears between IsType1 and File_1, showing that a relationship has been established. This line is brown to indicate a Custom event.

Procedure: How to Create the Second Decision Test Object

Create the second Decision Test object, which will receive all documents that do not contain the <SWIFT1> element. We want this object to capture all documents containing <SWIFT2> elements and route them to the second File object. We will route all other files to a dead-end File object.

1. Drag the Decision Test object from the toolbar to the workspace. The New Test Object dialog box opens.

2. Name the Decision Test object *IsType2*, and click *Next*.

The Test Operands dialog box opens, as shown in the following image.

A New Decision	Test Object
Test Operand	ds and One
Test Parameter Operands can Operand One	e literals, variables, or functions (XPATH(), LDAP(), or FILE())
Operation Operand Two	Is Not Null
? < E	Back Next > Finish Cancel

3. From the Operand One drop-down list, select Create XPath statement.

The XPath Builder tool opens.

4. Click Load XML File, and browse to the Route2 XML document that you created earlier.

The Route2 XML document contents appear in the XPath Builder dialog box, as shown in the following image.

XPATH Builder				
Load a document instance				
Load a sample of the document that will be used in the flow in order to browse its schema and build an XPATH.				
This wizard will assist in creating a valid XPATH for use with an object. By creating an XPATH and running it againsts the sever you can see the resuts that will be returned to your object property.				
Load Import Reload				
Previously loaded files				
Clear history				
<swift2>Route2 <user>USA</user> </swift2>				
XML Source: /Router_Example/route2.xml				
Sack Next > Finish Cancel				

5. Click Next.

The XPath Builder - Build an XPath dialog box opens.

6. In the left pane of XPath Builder, double-click the *Swift2*element. This adds the //SWIFT2 element to the XPath field, as shown in the following image.



7. Click Finish.

You are returned to the Test Operands dialog box, which now contains the XPath expression you just created, as shown in the following image.

🔬 New Decision Test Object			
Test Operands			
Enter a literal, variable, or function and an operation for the test to evaluate.			
Test Parameters			
Operands can be literals, variables, or functions (XPATH(), LDAP(), or FILE())			
Operand One XPATH(//SWIFT2)			
Operation Is Not Null			
Operand Two			
(?) < Back Next > Finish Cancel			

8. From the Operation drop-down list, select *Is Not Null*.

These parameters allow any XML input documents with a <SWIFT2> root tag to be captured.

9. Click Next.

The Value Occurrences dialog box opens, as shown in the following image.

🔬 New Decision Tes	t Object		
Value Occurrences Select the number of occurrences you wish to be returned.			
Test Occurrences Select 'Unique' if only one value may exist in the document, 'Any' if at least one must exist, and 'All' if every value must exist.			
🔘 Unique	Any	© All	
? Kack	Next >	Finish	Cancel

10. Select Any and click Finish.

The Decision Test object, IsType2, appears in the workspace, as shown in the following image.



Procedure: How to Define the Relationship Between IsType1 and IsType2

Build a relationship between Decision Test objects IsType1 and IsType2, so that an XML document coming into IsType1 does not have the tag <SWIFT1>, and then send it to IsType2.

- 1. Select the *IsType1* object, press the Shift key, and then select the *IsType2* object.
- 2. Right-click IsType2, and select Create Relation from the drop-down list.

The Line Configuration dialog box opens.

- 3. Select OnCustom from the Events drop-down list.
- 4. Select *false* as the case for the event.

The following image shows the Line Configuration dialog box with the OnCustom event, false case selections.

🔬 Relation Prope	erties			
Relation Configuration Use this dialog to configure a relationship between two objects using stock or custom event				
Event: OnCusto	m			
Case of:		✓ /* ×		
Case	Туре	Description		
OnError	Stock	Error		
OnSucc	Stock	Success		
📄 OnDefault	Stock	Used for a Decision of type		
📄 true	Custom			
🔽 false	Custom			
Description:				
Custom				
Don't ask again. Always create relations using default event type.				
?		Finish Cancel		

5. Click Finish.

A line appears between IsType1 and IsType2, showing that a relationship has been established. This line is brown to indicate a Custom event.

The following image shows the routing of XML documents from the Decision Test object IsType1, based on a true or false condition. If the document coming into IsType1 has the <SWIFT1> element, it will be routed to File_1. If it does not, it will be routed to IsType2.



Procedure: How to Route the Document Based on a Second Decision Test True Result

We want the XML documents captured by the IsType2 Decision Test object that contain a <SWIFT2> element, to be sent to a File object named File_2. To create the File_2 object:

1. Drag the File Execution object from the toolbar to the workspace.

The New File Object dialog box opens, as shown in the following image.

🔬 New File Object
Name and Description Provide object name and description
A File object allows a process to perform file operations at execution time. Name File_2
Description File Object
Тар
A Back Next > Finish Cancel

- 2. In the Name field, type *File_2*, and leave the Description as File object.
- 3. Click Next.

The File Type dialog box opens, as shown in the following image.

🔬 New File Object
File Object Type Select the type for the File Object object you wish to create
Defined File Object services in the current configuration are listed Type File Emit Agent {com.ibi.agents.XDFileEmitAgent}
Emits a document to a physical file
Seck Next > Finish Cancel

- 4. From the Type drop-down list, select *File Emit Agent*. This tells the File_2 object to write the incoming document to the disk.
- 5. Click Next.

The New File Object - Properties dialog box opens, as shown in the following image.

🔬 New File Object	t 📃 🗖 🗾 🔁	3
Object Proper Provide object pr	ties roperties	
▼ Main		
Source of Data	Source of data to write. If omitted, document will be used, else specify a data source location via xpath() function or any other function	
	v	=
Target	The target output directory	
Directory	C:\Temp\Out 👻	
File Pattern	The output file name, which can contain a '*' which gets expanded to a fine timestamp	
	TYPE_2_*.xml 👻	
Avoid	Should any preemitter be avoided?	
Preemitter	true 🔻	
Return	'status': status document will be the out document. 'input': in document will become the out document. 'swap': as input, but replace written data in the source nodes with the file name to which the data was written.	
	status 👻	÷
? < Ba	ack Next > Finish Cancel	

- a. In the Target Directory field, type a path to the directory where the file will be written. In this example, we will send the file to C:\Temp\Out.
- b. In the File Pattern field, type TYPE_2_*.xml.
- c. In the Avoid Preemitter field, select *true*. This ignores any assigned preemitters.
- d. In the Return field, select *status* to return a status document.
- 6. Click Finish.

The File_2 object appears in the workspace.

Procedure: How to Define the Relationship Between IsType2 and File_2

Build a relationship between Decision Test object IsType2 and the File_2 object, so that XML documents with a <SWIFT2> element coming into IsType2 will be sent to File_2.

- 1. Select the IsType2 object, press the Shift key, and then select the File_2 object.
- 2. Right-click File_2 and select Create Relation from the drop-down list.

The Line Configuration dialog box opens.

- 3. Select OnCustom from the Events drop-down list.
- 4. Select *true* as the case for the event.

The following image shows the Line Configuration dialog box with the OnCustom event and true case selections.

🔏 Relation Prope	erties	
Relation Conf Use this dialog t objects using st	i guration o configure a rel ock or custom ev	lationship between two vent
Event: OnCusto	m	•
Case of:		₽ /* X
Case	Туре	Description
OnError	Stock	Error
OnSucc	Stock	Success
📄 OnDefault	Stock	Used for a Decision of type
🔽 true	Custom	
📄 false	Custom	
Description:		
Custom		
🔲 Don't ask aga	in. Always create	e relations using default event type.
?		Finish Cancel

5. Click OK.

A line appears between IsType2 and File_2, showing that a relationship has been established. This line is brown to indicate a Custom event.

Procedure: How to Route the Document Based on a Second Decision Test False Result

We want all XML documents captured by the IsType2 Decision Test object that *do not* contain the <SWIFT1> or <SWIFT2> XML element, to be sent to the NONE File object. To set this up:

1. Drag the File Execution object from the toolbar to the workspace.

The New File Object dialog box opens, as shown in the following image.

🖉 New File Object
Name and Description Provide object name and description
A File object allows a process to perform file operations at execution time. Name NONE Description
File Object
Tap Image: Cancel Image: Cancel

- 2. In the Name field, type *NONE*, and leave the description as File object.
- 3. Click Next.

The File Type dialog box opens, as shown in the following image.

New File Object
File Object Type
Select the type for the File Object object you wish to create
Defined File Object services in the current configuration are listed
Туре
File Emit Agent {com.ibi.agents.XDFileEmitAgent}
Description
Emits a document to a physical file

- 4. From the Type drop-down list, select *File Emit Agent*. This tells the NONE File object to write the incoming document to the disk.
- 5. Click Next.

The New File Object - Properties dialog box opens, as shown in the following image.

new File Objec	t	x		
Object Proper Provide object pr	ties roperties			
▼ Main		-		
Source of Data	Source of data to write. If omitted, document will be used, else specify a data source location via xpath() function or any other function			
		•		
Target	The target output directory	_		
Directory	C:\Temp\Out	Ŧ		
File Pattern	The output file name, which can contain a '*' which gets expanded to a fine timestamp			
	NONE_OF_THE_ABOVE_*.xml	Ŧ		
Avoid	Should any preemitter be avoided?			
Preemitter	true	Ŧ		
Return	'status': status document will be the out document. 'input': in document will become the out document. 'swap': as input, but replace written data in the source nodes with the file name to which the data was writte	e en.		
	status	Ψ		
? < Ba	ack Next > Finish Can	cel		

- a. In the Target Directory field, type a path to the directory where the file will be written. In this example, we will write the file to C:\Temp\Out.
- b. In the File Pattern field, type NONE_OF_THE_ABOVE_*.xml.
- c. In the Avoid Preemitter field, select *true*. This ignores any assigned preemitters.
- d. In the Return field, select *status* to return a status document.
- 6. Click Finish.

The NONE File object appears in the workspace.

Procedure: How to Define the Relationship Between IsType2 and NONE

Build a relationship between Decision Test object IsType2 and the NONE File object, so that any XML documents coming into IsType2 without a <SWIFT2> tag are *not* routed to any other object in the process flow.

- 1. Select the *IsType2* object, press the Shift key, and then select the *NONE* object.
- 2. Right-click NONE and select Create Relation from the drop-down list.

The Line Configuration dialog box opens.

- 3. Select OnCustom from the Events drop-down list.
- 4. Select *false* as the case for the event.

The following image shows the Line Configuration dialog box with the OnCustom event and false case selections.

🔏 Relation Prop	perties				
Relation Configuration					
Use this dialog two objects us	Use this dialog to configure a relationship between two objects using stock or custom event				
Event: OnCust	tom				
Case of:		₽ /* ×			
Case	Туре	Description			
OnError	Stock	Error			
OnSuc	Stock	Success			
OnDef	Stock	Used for a Decision of typ			
📄 true	Custom				
👽 false	Custom				
Description:					
Custom	Custom				
Don't ask again. Always create relations using default event type.					
?		Finish Cancel			

5. Click Finish.

A line appears between IsType2 and NONE, showing that a relationship has been established. This line is brown to indicate a Custom event.

The following image shows the process flow with all of the required Decision Test and File objects.



The final step is to terminate the SWIFT_ROUTER process flow with an End object.

Procedure: How to Terminate the Process Flow

To terminate the process flow, add an End object, and then build a relationship between it and the File_1, File_2, and None objects.

1. Drag the End object onto the workspace.

The New End Object - End Name and Description dialog box opens.

- 2. Keep the default name, End, and leave the description as, End object.
- 3. Click Next.

The End Object Schema dialog box opens, as shown in the following image.

🔬 New End Object				
End Object Schema Select a schema to be assigned to the End Object				
Schemas available in t	he current server configuration are shown below. To viev			
Schema	Name of the schema which describes the output doc			
Root Tag Element	Name of the root tag element of the output document to this process flow. (* Required if a Schema is selected)			
Terminate	Select if this end object is the completion point			
Description	Description for the selected schema			
? < Back	Next > Finish Cancel			

- 4. Leave the Terminate box unchecked. This indicates that you want to obtain output from the process flow.
- 5. Click Finish.



The End object appears in the workspace, as shown in the following image.

Procedure: How to Define Relationships Between the End Object, File_1, File_2, and NONE

Build a relationship between the File_1 object and the End object to end the process flow.

- 1. Select the *File_1* object, press the Shift key, and then select the *End* object.
- 2. Right-click the *End* object and select *Create Relation* from the drop-down list. The Relation Configuration dialog box opens.

3. Select *OnCompletion* from the Events drop-down list. This directs the process flow to always follow this route.

🔬 Relation Pro	perties			
Relation Configuration Use this dialog to configure a relationship between two objects using stock or custom event				
Event: OnCor	npletion	•		
Case of:		÷/* ×		
Case	Туре	Description		
Description:				
Completion				
🔲 Don't ask ag	gain. Always cre	eate relations using default event type.		
?		Finish Cancel		

4. Click Finish.

A line appears between File_1 and End, showing that a relationship has been established. This line is blue to indicate an OnCompletion route, meaning the document will always follow this path.

5. Follow Steps 1 through 4, using the OnCompletion event type, to establish a relationship between File_2 and End, and then between NONE and End.

You are now ready to save, validate, and test the process flow.

Save the Process Flow

The complete process flow is shown in the following image.



Note: You can add text to the diagram to easily identify an area of the process flow. In this example, the text identifies the true and false decision paths. To add text to the workspace, drag the Text icon to the workspace, type the text you want to appear, click outside of the text box to paste the text in the workspace, then drag the text to its location in the diagram.

To save the process flow, in the File menu, select Save from the drop-down list.

Validate and Test the Process Flow

The validate feature verifies that the process flow structure is complete and properly assembled. Once the process flow is validated, you can test it with one of the input documents we created at the beginning of our example.

Procedure: How to Validate the Process Flow

To validate the SWIFT_ROUTER process flow:

- 1. In the Navigator pane, right-click the SWIFT_ROUTER process flow node.
- 2. Select Validate from the drop-down list.

The status of the validation appears in the Validation Results dialog box, as shown in the following image.



Procedure: How to Test the Process Flow

To test this process flow:

1. In the Navigator Pane, right click the *SWIFT_ROUTER* process flow node, select *Run* As from the drop-down list, and then select *Process Flow*.

The Edit configuration and launch dialog box opens, as shown in the following image.

ame: SWIFT	ROUTER				
Aain Test Sen	ver Registers Rur	ntime Options			
Process Flow	/:				
/Router_Exa	mple/Flows/SWIF	T_ROUTER.iwp			Browse
Input Docum	nent:				
/Router_Exa	mple/Flows/route	2.xml			Import
Server Enviro	onment:				
URL:	http://localhost:	9000			•
User Name:	iway				
Password:	••••				
O Use minin	nal runtime config	uration with depe	ndencies deplo	ved from the regis	trv.
O Use test se	erver	Janaalon maaraapa	indeneres depro	, ca non anc rega	
O Use existir	ng configuration:				Get Configurations
				Apply	Revert
				Run	Close

- 2. The Process Flow field contains the path to the SWIFT_ROUTER process flow.
- 3. In the Input document field, type the path or browse to the Route2.XML file you created at the beginning of this example.
- 4. In the Server Environment section, enter the URL, User Name, and Password.
- 5. If you are using a test server, select *Use test server*.
- 6. In the Test Server tab, browse to enter the Application and Template details.

Click the Runtime Options tab, as shown in the following image.

Z Edit Configuration				
Edit configuration and launch.				
Name: SWIFT_ROUTER (1)				
Main Test Server Registers Runtime Options				
Process Object Debugging				
 All - Trace all active objects 				
Select - Trace all debug enabled objects				
None - Tracing is disabled				
Server Side Tracing Debug - Report message based on debug level Deep Debug - Report message based on deep debug level				
Transaction Control				
Timeout for the response from the server during Test Run				
30				
Apply Revert				
? Run Close				

- 7. You can select the level of debugging that you want through the Process Object Debugging and Server Side Tracing options. In this example, we will select to trace all active objects and Commit Transactions, so that the transaction of the process flow will be performed during the test.
- 8. Click Apply.
- 9. Click Run.

The process flow goes through the test run and displays a status dialog box, as shown in the following image.

🔏 Test	Results Viewer Usage Information		x	
1	To view the output document(s) right click on the corresponding End object(s) in the Test Results view and select "View Output Document" or use the "Output Document" link found under the Trace tab in Properties view. To see previous test results browse to [your pflow]/Test Results under the Integration Explorer, right click on one of the dated test result folders and select "Open Test Results Viewer".			
Don't show this dialog again.				
		0	к	

The test results are displayed in the Message Console pane, as shown in the following image.



The workspace area displays a message that the test was successful, as shown in the following image.



To view the output document, right-click on the corresponding End object in the Test Results view, and select *View Output Document*. You can also use the Output Document link found under the Trace tab in the Properties view.


Process Flow Object Properties

This appendix provides detailed information about process flow objects that have extended properties.

In this chapter:

- Iterator Object Properties
- SQL Object Properties

Iterator Object Properties

The following sections describe Iterator Object properties and provide an example for each iterator type.

Reference: Attachment Iterator Properties

Property Name	Allowable Value	Description
Handling Option	A literal, variable, or XPath that evaluates to "leave unchanged" or "delete", or "replace."	Determines handling options based on the following criteria: <i>delete</i> removes each attachment from the current document after processing, <i>replace</i> replaces the original attachment with the modified document, <i>leave</i> leaves the original attachments unchanged.
Start Index	A numeric expression, constant, XPath, or variable.	Index of first attachment to process in the mail object. The first attachment to the main document has index 0.
Header Namespace	A literal, variable, or XPath expression.	Special register namespace where MIME headers for the current attachment will be stored. If replace handling option is selected, registers in this namespace will be added to the replaced attachment.

Property Name	Allowable Value	Description
Main Body Part Header Namespace	A literal, variable, or XPath expression.	If the current attachment is itself a Multipart, this is the special register namespace where the MIME headers for the main body part will be stored.
Keep Document Flat	A literal (true/false), variable or XPath expression.	Keeps the body of the document as an array of bytes

Reference: Chunk Iterator Properties

Property Name	Allowable Value	Description
Chunk Size in Bytes	A numeric expression, constant, XPath, or variable.	Size of the chunk in bytes.

Reference: Condition Iterator Properties

Property Name	Allowable Value	Description
Conditional Expression	A literal, variable, or XPath expression.	This is the expression controlling the termination of the loop
Iterate Rule	A literal, variable, XPath expression, 'do- while', or 'do-until'.	Determines the loop logic between do-while or do-until
Interim Document	A literal, variable, XPath expression, 'transient', or 'original'.	Determines if the document emitted at each iteration is the original document or the result of the previous iteration

Property Name	Allowable Value	Description
Output Document	A literal, variable, or XPath that evaluates to "original" or "transient", or "status."	Determines whether the document passed out of the Iterator object upon completion of the loop is the original unparsed document or the result status.

Reference: COUNTER Iterator Properties

Property Name	Allowable Value	Description
Iterations	A numeric expression, constant, XPath, or variable.	Number of times this iterator loop is to be repeated.
Interim document type	A literal, variable, or XPath that evaluates to "original" or "transient."	Determines whether the document passed out of the Iterator object into the loop is the original or the result of the previous iteration (transient).
Output document type	A literal, variable, or XPath that evaluates to "original" or "transient", or "status."	Determines whether the document passed out of the Iterator object upon completion of the loop is the original or the result of the previous iteration (transient).

Example: Counter Iterator Type

The com.ibi.agents.XDIterCount operator repeats the body of the loop a specified number of times. The number can be constant or computed at run time, for instance, with an XPath expression, as in this example. The inner part of the loop can be used to generate multiple outputs, read multiple inputs from a source, or perform any other operation repeatedly. The count iterator can also be used with conditional logic to add retry functionality to an adapter that does not natively support retry. The number of iterations of the loop is not dependent on the structure of the input document, nor does the count iterator parse the document before injecting it into the loop.

The Iterator object in this example passes the input document into the loop the number of times specified by the <count> XML tag in the input document. The following sections show the process flow, the Iterator object parameters, and the input and output documents of this example.

Process Flow Diagram



Iterator Object Parameters

Input Document

```
<IterCountTest>
<count>3</count>
</IterCountTest>
```

Output Document(s)

Three documents are output by the FileWrite object: IterCount01.xml, IterCount02.xml, and IterCount03.xml.

Reference: FTP Iterator Properties

Property Name	Allowable Value	Description
Host Name	A string expression, constant, XPath, or variable.	DNS name (or IP address) of the FTP server that you want to connect to. Use host : port if not standard port 21.
Remote Port	A numeric expression, constant, XPath, or variable.	Port to connect to on the FTP site, blank for default port 21.

Property Name	Allowable Value	Description
User Name	A string expression, constant, XPath, or variable.	The valid user ID on the FTP server.
Password	A string expression, constant, XPath, or variable.	The valid password for the FTP server.
Account Name	A string expression, constant, XPath, or variable.	Account name which is the valid account for the FTP server.
User Passive Command	A literal (true/false), variable or XPath expression.	Uses PASV command if true, otherwise uses PORT command.
Timeout	A numeric expression, constant, XPath, or variable.	Timeout interval, in seconds, for socket.
Retry Interval	A string expression, constant, XPath, or variable.	Retry interval in seconds (allows for <i>xxhxxmxxs</i> format). Omit or use 0 for no retry.
Connection Retry	A string expression, constant, XPath, or variable.	Number of retry attempts after failed connection to FTP server.

Reference: Island Iterator Properties

Property Name	Allowable Value	Description
Island identifier	A string expression, constant, XPath, or variable.	Name of the XML tag that identifies the Island command string.

Property Name	Allowable Value	Description
Remove Identifier	A Boolean literal, variable, or XPath.	Determines whether the Island tag and text will be removed or preserved in the final document.

Example: Island Iterator Type

The {com.ibi.agents.XDIterIsland} iterator is designed to perform document enhancement, that is, to replace parts of the input XML document with the results of a call to an agent or adapter. Any external data source can be targeted (Web Services, SQL data, and so on), but in this example embedded file names are replaced with the content of the named disk file. The replacement would behave identical regardless of the source, although the replacement data must be in XML format. Island replacement is iterated for as many tags as are present in the input document. The tags to be replaced may be homogeneous, as in this example, or heterogeneous by using multiple sequential Island iterators targeting different sources.

The following sections show the process flow, the Iterator object parameters, and the input and output documents of this example.



Process Flow Diagram

Iterator Parameters

Input Document

The primary input document is the document that contains the Islands to be replaced. In this example, they are identified by the XML tag <file>.

```
<mydoc>
<file>file1.txt</file>
<file>file2.txt</file>
<file>file3.txt</file>
</mydoc>
```

file1.txt

<content>Contents of file 1</content>

file2.txt

```
<content>
<line>Contents of file 2</line>
<line>There are 2 lines</line>
</content>
```

file3.txt

<content>Contents of file 3</content>

Output Document(s)

The input document and the island documents are combined according to the behavior of the iterator.

Reference: Iter RVI Iterator Properties

Property Name	Allowable Value	Description
Service Name	A literal, variable, or XPath expression.	The name of the service to execute the current message.
Tolerance	A string expression, constant, XPath, or variable.	Determines the time, seconds, to await circuit availability to access the names service.
Timeout	A string expression, constant, XPath, or variable.	Time, in seconds, to await response from the gateway.

Property Name	Allowable Value	Description
Attempt Retry	A literal (true/false), variable or XPath expression.	If true, a retry exception is thrown when no gateway connections are available, message cannot be written to the gateway, or this service times out while awaiting response. The retry exception will activate the channel's pending system, when available. If false, this service will attempt to acquire a new connection if message cannot be written to the gateway, and if a connection is available, it will retry automatically. If no connections are available, or if the service times out or otherwise fails while reading the response, the service will return a failure edge.
Method of compression to use	A literal, variable, XPath expression, none, smallest, fastest, standard, or Huffman.	Describes the form of compression that should be used on the output.

Reference: Line Iterator Properties

Property Name	Allowable Value	Description
Keep blanks	A literal (true/false), variable or XPath expression that evaluates to a Boolean result.	Determines whether blank lines will be passed into the loop for processing.
Output	A literal, variable, or XPath that evaluates to "original" or "status".	Determines whether the document passed out of the Iterator object upon completion of the loop is the original unparsed document or the result status.

Example: Line Iterator Type

The {com.ibi.agents.XDIterLine} operator repeats the body of the loop once for each line in the document, as defined by standard end-of-line delimiters. The inner part of the loop can be used to generate multiple outputs, read multiple inputs from a source, or perform any other operation repeatedly. When using this iterator, the number of times through the loop and how the input is parsed is directly related to the structure of the input document, in particular, how many lines are present. We recommend that non-XML, line-oriented input be read into the process flow with a File object rather than being passed through the Start node.

The following sections show the process flow, the Iterator object parameters, and the input and output documents of this example.



Process Flow Diagram

Iterator Object Parameters

Input Document

```
This is the first line
This is line 2
Line 3
```

Output Document(s)

Three documents are output: IterLine01.txt, IterLine02.txt, and IterLine03.txt.

```
This is the first line
This is line 2
Line 3
```

Reference: SQL Batch Iterator Properties

Property Name	Allowable Value	Description
Input Provider	A literal, variable, or XPath expression.	Name of the provider for accessing input

Property Name	Allowable Value	Description
Select SQL	A literal, variable, or XPath expression.	The SQL Select statement to get rows to pass into the flow
Transaction Isolation Level	A literal, variable, XPath expression, As Is, Read Committed, Read Uncommitted, Repeatable Read or Serializable.	Transaction isolation level to be set if possible
Base 64 if Needed	A literal (true/false), variable, or XPath expression.	If set, all fields are checked for base 64 (this is expensive)
Input Encoding	A literal, variable, XPath expression, Leave, Platform, US- ASCII, CP-037/ EBCDIC, ISO-8859-1, UTF-8, UTF16-BE, UTF16-LE, UTF16.	The IANA character set to be used for the input data base. Select from the standard list or enter an encoding name.
Format as SQL Agent	A literal (true/false), variable, or XPath expression.	If true, the generated rows are formatted according to SQLAgent field schema.
Three Part Name	A literal (true/false), variable, or XPath expression.	If true, names will be presented a full three part names if supported by the data base.

Reference: SQL Select Iterator Properties

Property Name	Allowable Value	Description
Input Provider	A literal, variable, or XPath expression.	Name of the provider for accessing input

Property Name	Allowable Value	Description
Select SQL	A literal, variable, or XPath expression.	The SQL Select statement to get rows to pass into the flow
Transaction Isolation Level	A literal, variable, XPath expression, As Is, Read Committed, Read Uncommitted, Repeatable Read or Serializable.	Transaction isolation level to be set if possible
Base 64 if Needed	A literal (true/false), variable, or XPath expression.	If set, all fields are checked for base 64 (this is expensive)
Input Encoding	A literal, variable, XPath expression, Leave, Platform, US- ASCII, CP-037/ EBCDIC, ISO-8859-1, UTF-8, UTF16-BE, UTF16-LE, UTF16.	The IANA character set to be used for the input data base. Select from the standard list or enter an encoding name.
Output Document Type	A literal, variable, or XPath that evaluates to "original" or "status."	Determines whether the document passed out of the Iterator object upon completion of the loop is the original or the result status. By definition, the Interim document type is always the next sibling (that is, transient), never the original.
Format as SQL Agent	A literal (true/false), variable, or XPath expression.	If true, the generated rows are formatted according to SQLAgent field schema.
Three Part Name	A literal (true/false), variable, or XPath expression.	If true, names will be presented a full three part names if supported by the data base.

Reference: Sibling Iterator Properties

Property Name	Allowable Value	Description
Output document type	A literal, variable, or XPath that evaluates to "original" or "status."	Determines whether the document passed out of the Iterator object upon completion of the loop is the original or the result status. By definition, the Interim document type is always the next sibling (that is, transient), never the original.

Example: Sibling Iterator Type

The {com.ibi.agents.XDlterSibSplit} construct is used to contain the output of multi-row SQL queries. By bundling sets of rows, siblings makes the handling of very large result sets more practical. When used with the IterSibSplit operator, the logic inside the loop is executed once for each sibling document. One result row per sibling document is recommended for this iterator. Because the entire answer set is fetched prior to executing the loop, flows using this technique may have high memory requirements.

The following sections show the process flow, the Iterator object parameters, and the input and output documents of this example.



Process Flow Diagram

Iterator Object Parameters

In this case, the Iterator object parameters are not material, and the parameters of the SQL object that generates the internal document are significant. Note that the properties are set to produce siblings (which are internal, non-XML iWay constructs), with one SQL row for each sibling.

Property Name	Allowable Value	Description
Operand One	A literal, variable, or XPath that evaluates to a value.	The first of two operands to compare.
Operation	The comparison to apply to the two operands, or a null test predicate on Operand One.	Used to compare Operand One and Operand Two, producing a Boolean result.
Operand Two	A literal, variable, or XPath that evaluates to a value.	The second of two operands to compare. In the case of a null/not null test, this parameter is optional.
Iterate Rule	A literal (true/false), variable, or XPath expression that evaluates to a Boolean result.	Used to invert the logic of the Iteration test from do-while to do-until. Using a non-constant expression can simplify the flow logic by reversing the sense of the test dynamically.
Interim document type	A literal, variable, or XPath that evaluates to "original" or "transient."	Determines whether the document passed out of the Iterator object into the loop is the original or the result of the previous iteration (transient).
Output document type	A literal, variable, or XPath that evaluates to "original" or "transient", or "status."	Determines whether the document passed out of the Iterator object upon completion of the loop is the original or the result of the previous iteration (transient).

Reference: Test Iterator Properties

Example: Test Iterator Type

The {com.ibi.agents.XDlterTest} operator repeats the body of the loop while a specified logical comparison evaluates to True. The predicate is reevaluated each time through the loop. The inner part of the loop can be used to generate multiple outputs, read input multiple times from a source, or perform any other operation repeatedly. Similar to the count iterator, the test iterator can be used to retry an operation until it is successful or other criteria are met.

The following sections show the process flow, the Iterator object parameters, and the input and output documents of this example.





Iterator Object Parameters

Reference: XML SPLIT Iterator Properties

Property Name	Allowable Value	Description
Snip Levels	A literal, variable, or XPath that evaluates to an integer that is zero or greater.	Determines how many tag levels to trim off of the new XML document, starting from the top (root) node. Zero equals preserve all levels, one equals cut off the root, two equals cut off the root and immediate child, and so on, down to the XPath location being iterated. The default is zero, which will return all parent tags up to the root. Note: Cross Section data may be lost depending on the snip level setting.
Expression	An XPath string, derived from a constant, XPath, or variable.	An XPath that indicated where in the document to begin processing.

Property Name	Allowable Value	Description
Cross Section	A literal, variable, or XPath that evaluates to true or false.	When set to false, only the direct parent tags of the iterated section are present in the split document. When set to true, all sibling sub-trees of
		parent tags are preserved.
Output document type	A literal, variable, or XPath that evaluates to "original" or "status."	Determines whether the document passed out of the Iterator object upon completion of the loop is the original, or the result status. By definition, the Interim document type is always the split document (that is, transient), never the original.

When you are finished entering the properties, click *Finish*. The Iterator object appears in the workspace.

The following examples show an implementation of each iterator type.

Example: XML SPLIT Iterator Type

Often, a single transaction document contains repeating groups of data, in this example, a simplified invoice with multiple line items. The {com.ibi.agents.XDIterXMLSplit} iterator extracts the repeating group specified by the XPath expression. The parsed document contains the indicated group and preserves its direct parental lineage of XML nodes. If the logic inside your loop needs data from a parallel XML path (the invoice number in this example), then you can set Cross Section to true. Cross Section will preserve the entire document above the node that is the target of Expression.

The body of the loop is repeated once for each occurrence of the indicated group. The inner part of the loop can be used to process the repeating groups, call a Web service, insert the data into an SQL database, and so on. When using this iterator, the number of times through the loop and how the input is parsed depends on the structure of the input document and the XPath expression.

The following sections show the process flow, the Iterator object parameters, and the input and output documents of this example.

Process Flow Diagram



Iterator Object Parameters

Input Document

```
<invoice>
    <invoice-number>1001</invoice-number>
    <line-item>
        <item-number>1</item-number>
        <item-desc>Red Widget</item-desc>
        </line-item>
        <item-number>2</item-number>
        <item-desc>Blue Widget</item-desc>
        </line-item>
        <line-item>
        <item-number>3</item-number>
        <item-desc>Green Do-dad</item-desc>
        </line-item>
        </li
```

Output Document(s)

Three documents are output by the FileWrite object: IterXmlSplit01.xml, IterXmlSplit02.xml, and IterXmlSplit03.xml.

```
<?xml version="1.0" encoding="UTF-16" ?>
<invoice>
   <line-item>
      <item-number>1</item-number>
      <item-desc>Red Widget</item-desc>
   </line-item>
</invoice>
<?xml version="1.0" encoding="UTF-16" ?>
<invoice>
   <line-item>
      <item-number>2</item-number>
      <item-desc>Blue Widget</item-desc>
   </line-item>
</invoice>
<?xml version="1.0" encoding="UTF-16" ?>
<invoice>
   <line-item>
      <item-number>3</item-number>
      <item-desc>Green Do-dad</item-desc>
   </line-item>
</invoice>
```

The following is an output document with Cross Section set to true.

```
<?xml version="1.0" encoding="UTF-16" ?>
<invoice>
    <invoice-number>1001</invoice-number>
    <line-item>
        <item-number>1</item-number>
        <item-desc>Red Widget</item-desc>
        </line-item>
</invoice>
```

The following is an output document with Snips Level set to 1.

SQL Object Properties

This section lists and describes the properties available to the SQL object.

Property	Valid Values	Description
Evaluate SQL	true or false.	Determines whether the SQL statement will be subject to iWay Functional Language evaluation.
		When set to true, iSM attempts to evaluate the statement, in which case, care must be taken to escape characters that are significant to iWay (for example, single quotes).
		The default value is false. This prevents ambiguity and protects against side effects.
Output Format	row, column, or field.	Specifies how to construct the output XML document for the answer set (required, but meaningful for SELECT only).
Use JNDI	True, False, Edit, Create variable, Create XPath statement, Load IFL expression	If value is set to true, a connection is established from a data source, accessed via JNDI
JNDI Name	Edit, Create variable, Create XPath statement, Load IFL expression	JNDI Name for the requested Data Source. Required field if using JNDI.
JNDI Factory	Edit, Create variable, Create XPath statement, Load IFL expression	JNDI initial context factory class. Default is blank.
Transaction Isolation Level	asis, readUncommitted, readCommitted, repeatableRead, or serializable.	Sets the isolation level for this connection. For information on locking, see your database documentation.

Property	Valid Values	Description
Pool Connections	true or false.	When set to true, all connections are pooled by the URL. This enables the first connection to the URL to be shared by subsequent uses of the SQL Agent that use the same URL.
		The true value is appropriate for SELECT, but may not be appropriate for SQL statements that require commit. The default value is false.
Attempt Read Only	true or false.	If set to true, for SELECT statements sets JDBC 'read only' flag. Some drivers cannot handle this optimization and may report security failures. The default value is false.
Connection Properties	String literal or iWay Functional Language expression.	These are additional vender-specific properties to be added to the JDBC URL.

Property	Valid Values	Description
Need Commit	true or false.	Set if the SQL executed modifies the data table(s). In this case, pooling performed in response to the Pool Connections setting is on the worker level. This makes the connection serial, enabling the system to maintain proper commit scopes. This property affects only pools. SQL statements other than SELECT are always committed.
		Note: If configured, this service participates in iWay server transaction management.
		The default value is false.
Timeout		Describes the number of seconds the service will wait for an operation to complete.
		If a value is set, it alerts a driver to attempt to stop the connection after the defined period.
Max Rows	0 (zero) or any positive integer.	Maximum number of rows to be returned for a select DML statement.
		To return all rows, use zero. If the statement returns more rows than the maximum, the remainder will not be part of the response document. These rows can, however, be appended as sibling documents as explained in the Make Siblings property.

Property	Valid Values	Description
Make Siblings	true or false.	Indicates whether or not to make sibling documents.
		A sibling is a complete XML document that is related to another XML document. In the case of the SQL Agent, each sibling carries the max rows number of response rows. Specialized agents and components are 'sibling-aware' and can work with sibling documents (for example, IterSibSplit and emitters). Although many agents carry the siblings forward and most protocols can properly deal with siblings, it is not guaranteed that all will do so.
		When siblings are used, we recommend that the SQL agent be placed immediately before an object that can process sibling documents. The default value is false.
Max Siblings	A positive integer.	If Make Siblings is set to true, then this is the Maximum number of siblings created in one invocation of the SQL Object.
Emit No Empty	true or false.	If set to true, empty documents are not emitted.
		The default value is false.

Property	Valid Values	Description
Issuance Strategy	construct or bind.	Choose whether to issue data through SQL construction performed by iSM, or through SQL bind (prepare/execute).
Reply Node Name	String literal or iWay Functional Language expression.	If entered, this represents the XML node (that is, tag) that will contain the response from the issued SQL statement.
Base 64 if Needed	true or false.	When set to true, all fields are checked to see if they need base64 encoding.
		Note: The true setting has a negative impact on performance.
xLOB Handling	none, external, inline, String literal or iWay Functional Language expression Default is none	Defines how blobs and clobs are to be treated
Want Generated Keys	True, false, String literal or iWay Functional Language expression	If set to true, generated keys on a non-bound INSERT statement are returned, if available, from the data base in the iway.genkey' register.

Property	Valid Values	Description
Output Document	result, input, String literal or iWay Functional Language expression	Determines if an operation does not return a result set, if the agent should return the standard result document or if it should return only its input.
		If input is selected, the response node from the result document that would otherwise have been returned will be stored in a special register named "iway.sqlstatus".
		If the operation is not successful, an error or result document will always be returned.
Call at EOS	True, false, String literal or iWay Functional Language expression	Determines if when using a streaming preparser in a channel, a last call is made after the last document.
User ID	String literal or iWay Functional Language expression (SREG, and so on).	The user ID to connect to the RDBMS.
Password	String literal or iWay Functional Language expression (SREG, and so on).	The password to connect to the RDBMS.

Feedback

Customer success is our top priority. Connect with us today!

Information Builders Technical Content Management team is comprised of many talented individuals who work together to design and deliver quality technical documentation products. Your feedback supports our ongoing efforts!

You can also preview new innovations to get an early look at new content products and services. Your participation helps us create great experiences for every customer.

To send us feedback or make a connection, contact Sarah Buccellato, Technical Editor, Technical Content Management at Sarah_Buccellato@ibi.com.

To request permission to repurpose copyrighted material, please contact Frances Gambino, Vice President, Technical Content Management at *Frances_Gambino@ibi.com*.

Inf%rmation Builders

iWay

['] iWay Integration Tools Designer User's Guide

Version 7.0.x and Higher

DN3502272.0418

Information Builders, Inc. Two Penn Plaza New York, NY 10121-2898

