Information Builders

iWay

iWay Protocol Adapter for Connect:Direct User's Guide

Version 7.0.x and Higher

August 16, 2018

Contents

Preface

This document is written for system integrators who develop client interfaces between Connect:Direct and other applications. It describes how to use the iWay Protocol Adapter for Connect:Direct to integrate with your Connect:Direct server. It is assumed that readers understand web technologies and have a general understanding of Microsoft Windows and UNIX systems.

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	Introducing the iWay Protocol Adapter for Connect:Direct	Provides an overview of the iWay Protocol Adapter for Connect:Direct and summarizes how to use it to integrate Connect:Direct systems with other applications.
2	Connect:Direct Supported Platforms Matrix	Specifies version, platform, and database support information for iWay Protocol Adapter for Connect:Direct.
3	Installing the iWay Protocol Adapter for Connect:Direct	Describes how to install the iWay Protocol Adapter for Connect:Direct.
4	Creating XML Schemas and Business Services	Describes how to use iWay Explorer to create XML schemas and Business Services for integration between the iWay Protocol Adapter for Connect:Direct and a Connect:Direct target server.
5	Configuring Event Handling for the iWay Protocol Adapter for Connect:Direct	Describes how to configure event handling for the iWay Protocol Adapter for Connect:Direct.

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

Customer Support

Do you have 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. 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 table lists the environment information that our consultants require.

Platform	
Operating System	
OS Version	
JVM Vendor	
JVM Version	

The following table lists the deployment information that our consultants require.

Adapter Deployment	For example, iWay 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).	

Request/Question	Error/Problem Details or Information
Any change in the application environment: software configuration, EIS/database configuration, application, and so forth?	
Under what circumstance does the problem *not* occur?	

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 services 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. You can 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 1

Introducing the iWay Protocol Adapter for Connect:Direct

This section provides an overview of the iWay Protocol Adapter for Connect:Direct and summarizes how to use it to integrate Connect:Direct systems with other applications.

In this chapter:

Features of the iWay Protocol Adapter for Connect:Direct

The iWay Protocol Adapter for Connect:Direct is a bi-directional adapter offering message and event flows to and from Connect:Direct systems. The adapter enables integration with Connect:Direct using:

- **Services.** Services allow the adapter to invoke various Connect:Direct API commands to the application through standard XML messages.
- **Events.** Events allow the adapter to monitor process status, statistics, and file directories through Connect:Direct API invocations and generate application events accordingly.

Key features of the iWay Protocol Adapter for Connect:Direct include the ability to:

- Submit file-based processes.
- Configure modes and nodes.
- Construct and submit composite processes.
- Monitor process status and generate events.
- Monitor statistics and generate events.
- Monitor file directories and generate events.

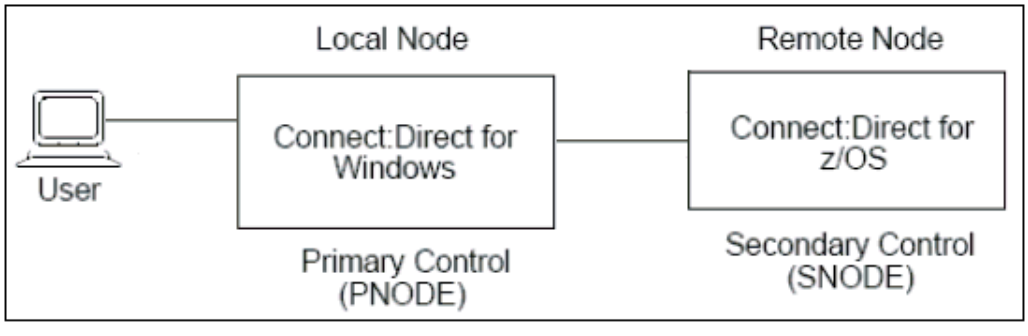
Understanding Connect:Direct

Connect:Direct is an enterprise solution that provides secure and automated file delivery between business applications. Session management and extended recovery features ensure data delivery 24 hours a day, 7 days a week across business operations.

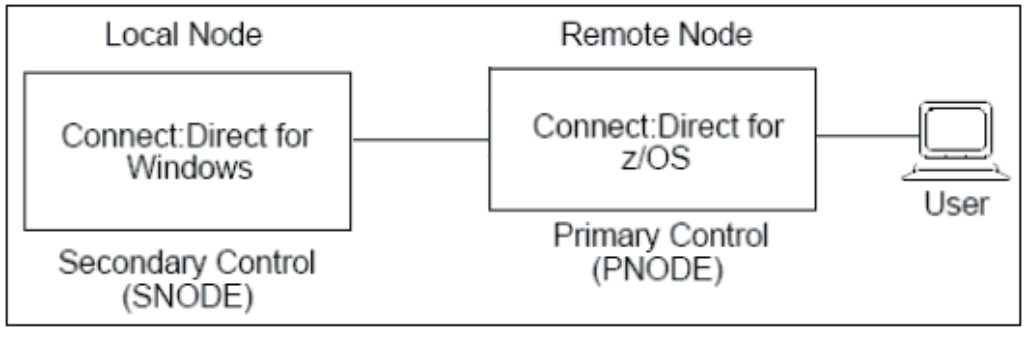
Connect:Direct links technologies, moves all types of information, and manages high performance transfers by providing easy-to-use automation, checkpoint and restart, and error recovery.

Connect:Direct also provides choices in communications protocols, hardware platforms, and operating systems that can be used. It provides the flexibility to move information from mainframes to midrange systems to desktop and LAN-based workstations.

The following diagram illustrates a Connect:Direct Windows user attaching to the server and submitting a process to run on the Windows node. The process copies data from the Windows node to the z/OS node. The Windows node is the local node and has primary control because it initiates the activity.



The following diagram illustrates a Connect:Direct z/OS user submitting a process to run on the z/OS node. The process copies data from the Connect:Direct for Windows node to the z/OS node. The z/OS node has primary control because it initiates the activity and is called the PNODE.



Typical requests to a Connect:Direct system involve submitting a process or querying for status or statistics information.

For example, the following statements represent a process used to copy a file from a source to a destination:

```
/*BEGIN_REQUESTER_COMMENTS
$PNODE$="MYCDSERVER" $PNODE_OS$="Windows"
$SNODE$="MYCDSERVER" $SNODE_OS$="Windows"
$OPTIONS$="WDOS"
END_REQUESTER_COMMENTS*/
MYPROC1 PROCESS
  SNODE=MYCDSERVER
//Starting MyProc1
COPY1 COPY
FROM (
   FILE=C:\filein.doc
)
TO(
   FILE=C:\fileout.doc
   DISP=NEW
)
CKPT=100M
MYIFFY IF (COPY1 EQ 0) THEN
//Success
ELSE
ELSE IF
PEND
```

The iWay Protocol Adapter for Connect:Direct interprets request XML documents and then constructs and submits this process accordingly.

Service Adapter Functionality

The Service adapter:

- Receives request documents.
- Interprets the request to be either a simple or a composite command.
- Constructs the command and process accordingly.
- Invokes the command and process on the target Connect:Direct server.
- Transforms the Connect:Direct response to a valid XML document according to the published schemas.
- Returns the response document to the invoking component.

The adapter does not require any pre-processing or post-processing logic and also does not enforce any kind of security. Security is provided as part of the iWay container.

In order to fully expose the integration between iWay and Connect Direct, the adapter includes two service types:

- **Composite Command Service.** The composite command service can be used to construct and submit a Connect:Direct process.
- **Simple and API Commands.** Simple and API commands are directly invoked as atomic API requests to the Connect:Direct application. These commands include statistics, process, and node commands among others.

The composite command service differs from the simple and API command service as follows:

- There is no API specification for the composite commands. Composite commands are built according to the syntax used to build a .cdp file. A Connect:Direct process is saved on disk as a .cdp file. The syntax of a .cdp file can be found by referring to the CDRequester tool.
- The composite commands have to be invoked within the context of a process command. Since composite commands cannot be invoked on their own, they can be considered as nested commands.
- Since composite commands are submitted as part of a process, command execution may not be performed immediately. The process can be scheduled to start at a specified time.
- Simple commands are directly invoked against the Connect:Direct target server and usually are immediately executed.

Composite Command Service

The composite command service can be used to construct and submit a Connect:Direct process. The composite command can consist of any combination of the following Connect:Direct operations:

- **Copy.** Copy files to and from an SNODE.
- **Submit.** Submit a process to a PNODE/SNODE. The process is contained on the file system in the form of a .cdp file.
- **Run Task.** Run a Connect:Direct task on the SNODE/PNODE.
- **Run Job.** Run a Connect:Direct job on the SNODE/PNODE.

Using iWay Explorer, the iWay Protocol Adapter for Connect:Direct provides XML schemas for process commands within each target to represent composite commands as XML requests.

Simple and API Commands

Simple commands contain all of the commands exposed by the Connect:Direct API specification. These commands include:

- **Process.** Select, submit, delete, and change.
- **Statistics.** Select.
- **Tracing.** Select and change.
- **Mode.** Select, add, delete, and change.
- **Node.** Select, add, delete, and change.
- **Lsdir.**
- **Proxy.** Select, submit, delete, and change.

The API commands are also represented by individual XML request and response schemas that are provided within each operation using iWay Explorer.

Event Adapter Functionality

The Event adapter provides the ability to monitor the Connect:Direct node for:

- **Statistics.** The Event adapter issues a select statistics command internally and uses the configured UI parameters as command parameters. This determines what kind of statistics are returned or cannot be returned.

Issue select statistics to examine records in the Connect:Direct statistics database. The type of information in the output from this command includes data such as copy status and execution events. The search criteria provides flexibility in selecting information you want to retrieve. Additional parameters determine the form in which the information is presented. When using select statistics, specify your selection criteria carefully to avoid displaying excessive volumes of records. If you do not provide selection criteria, then all records for the day are retrieved.

- **Process.** The Event adapter internally issues a select process command and uses the configured UI parameters as command parameters. Use select process to display information about processes in the TCQ. Select processes by name, number, queue, SNODE, status, submitter, or a combination of these identifiers.

- **Directory.** Provides the ability to monitor a remote directory for files. The LSDIR command returns information about files present in the remote directory. The function only monitors and does not manipulate the files themselves. The information returned includes file name, file owner, access rights, and time stamp.

For more information about the above commands, see the *Connect:Direct System Guide*.

The Event adapter also provides the capability to embed SQL statements as configuration parameter values. The statements are evaluated at runtime when the adapter is activated. The command parameters are then seeded with the results from the SQL statement. As a prerequisite before a select statement is executed, a database must be configured. The database can be configured through the relevant Event adapter configuration page.

Additionally, the Event adapter offers the ability to load any configuration parameter value from a SQL statement. This can be used to monitor processes submitted by the Service adapter.

Submitting a process using the Service adapter returns the process ID as the response. The process ID can be stored in a database.

The Event adapter can be configured to extract the process IDs from the database during its poll cycle and use the IDs as a filter for its process select command.

As a result, real-time status events for processes that have been submitted using iWay can be provided.

Chapter 2

Connect:Direct Supported Platforms Matrix

iWay Software is committed to support the diverse environments and varied systems of our users through support for leading enterprise applications, platforms, and databases.

This section specifies version, platform, and database support information for iWay Protocol Adapter for Connect:Direct. It is designed to provide a consolidated view of Connect:Direct releases and the various operating systems and databases, on which they are supported.

In this chapter:

Connect:Direct Supported Platforms Matrix Overview

Connect:Direct provides high-volume, security-rich, and assured delivery of files within and among enterprises through Connect:Direct transmission protocol using CDJAI JAVAI API.

Supported Connect:Direct Versions

iWay Protocol Adapter for Connect:Direct supports the following versions:

- Connect: Direct 4.4 on Windows
- Connect: Direct 4.6 on Windows

Connect:Direct Operating Systems

iWay Protocol Adapter for Connect:Direct supports all of the operating systems that are listed in the *iWay Installation and Configuration Guide* under *Operating System Requirements*.

Databases

iWay Protocol Adapter for Connect:Direct functions only with the FILE System.

Java Development Kit (JDK)

iWay Protocol Adapter for Connect:Direct supports the Java Development Kit (JDK) versions that are listed in the *iWay Installation and Configuration Guide* under *Java Requirements*.

Communication Modes

iWay Protocol Adapter for Connect:Direct supports the following communication modes:

- **Services.** The iWay Protocol Adapter for Connect:Direct can receive messages from the Connect:Direct server.
- **Events.** The iWay Protocol Adapter for Connect:Direct events are not supported.

Connect:Direct Object Types and Interfaces

Objects are lists of group commands. Process is the compound command which consists of COPY, RUNTASK, and RUNJOB. The iWay Protocol Adapter for Connect:Direct supports the following Connect:Direct Object Types and Interfaces:

- PROCESS
- COPY
- RUN TASK
- STATISTICS
- LISDIR
- RUN JOB

Connect:Direct Communication Types

iWay Protocol Adapter for Connect:Direct supports the following communication types:

- **PROCESS:** Asynchronous

- **COPY:** Synchronous
- **STATISTICS:** Synchronous
- **LISDIR:** Synchronous

Connect:Direct Operations

iWay Protocol Adapter for Connect:Direct supports the following operations:

- **PROCESS.** The composite command uses COPY to transfer files between different computers and platforms.
- **LISDIR.** Lists the directory structure of the specified node.
- **STATISTICS.** The statistics details report contains detailed statistics information about processes occurring on managed Sterling Connect:Direct servers during a specified time period.

Connect:Direct Data Types

iWay Protocol Adapter for Connect:Direct uses the String data type.

Other Connect:Direct Functions

There is no known list related to other functions for iWay Protocol Adapter for Connect:Direct.

Known Connect:Direct Limitations

The iWay Protocol Adapter for Connect:Direct contains no unknown limitations.

Related Information for Connect:Direct in Specific iWay Releases

For more information, see the *iWay New Features Bulletin and Release Notes* documentation for a specific release (for example, iWay Version 7.0.2).

Chapter 3

Installing the iWay Protocol Adapter for Connect:Direct

This section describes how to install the iWay Protocol Adapter for Connect:Direct.

In this chapter:

Connect:Direct Prerequisites

Before you install the iWay Protocol Adapter for Connect:Direct, ensure that the following prerequisites are met:

- iWay Service Manager (iSM) is deployed to the host environment.

  For more information, see the *iWay Installation and Configuration Guide*.

- A Connect:Direct system is installed and available.

Connect:Direct Installation Files

The iWay Protocol Adapter for Connect:Direct requires the following installation files:

- **iwconnectdirect.jar.** The .jar file containing core adapter classes.

  Note: The iwconnectdirect.jar file is automatically installed in the <*ism_home*>\lib directory during the iSM installation process. Ensure that *ConnectDirect* is selected in the Protocol Adapters category of the Adapter Selection pane during the iSM installation.

- **CDJAI.jar.** The Connect:Direct system client library.

  Note: This file is not provided by the iSM installation and must be obtained from a third-party.

- **JMS.jar.** The Java Message Service (JMS) API library.

  Note: This file is not provided by the iSM installation and must be obtained from a third-party.

Installing the Protocol Adapter for Connect:Direct

To install the iWay Protocol Adapter for Connect:Direct, ensure that the following files:

- **iwconnectdirect.jar**
- **CDJAI.jar**
- **JMS.jar**

are copied and available in the following directory:

<ism_home>\lib

where:

<ism_home>

Is the root directory (home) as specified during the iSM installation.

Restart iSM when you have finished.

Chapter 4

Creating XML Schemas and Business Services

This section describes how to use iWay Explorer to create XML schemas and Business Services for integration between the iWay Protocol Adapter for Connect:Direct and a Connect:Direct target server.

In this chapter:

Creating XML Schemas and Business Services Overview

The service adapter provides a facility where multiple commands can be submitted in a single XML request document. The XML response document will return a response for individual commands, as well as provide correlation for the command that is invoked.

The commands can be any combination of simple or composite commands.

All commands are represented by a top level *command* element, which has the following attributes:

- **name.** For example, *process*, *statistics*, *COPY*, and *RUNJOB*.
- **action.** For example, *select*, *submit*, *delete*, and so on.

  Note: In case of composite commands, nested commands do not use the action attribute. In addition, the action for the *PROCESS* command is fixed to *submit*.
- **type.** For example, *simple* and *composite*.

Additionally, the *command* element can contain several child elements, which represent parameters allowed by the particular command.

The child element names are mapped to command parameter names. The child element values are used to set command parameter values.

Composite Commands

Composite commands are always submitted as single level nested commands under a parent process composite command.

The following is an example of a composite command (COPY) that is being sent through an XML request document:

```
<?xml version="1.0" encoding="UTF-8"?>
  <CDServiceRequest>
    <command name="PROCESS" action="submit" type="composite">
      <label>MYPROC</label>
      <snode>MYCDSERVER</snode>
      <command name="COPY" type="composite">
         <label>MYCPY</label>
         <snode>MYCDSERVER</snode>
         <direction>SEND</direction>
         <source>c:\downloads\iWay_brochure.pdf</source>
         <destination>myout.pdf</destination>
         <disposition>NEW</disposition>
         <ckpt></ckpt>
         <compress></compress>
         <srcsysopts></srcsysopts>
         <destsysopts></destsysopts>
      </command>
      <command name="RUNTASK" type="composite">
         <label>MYTSK</label>
         <submitto>SNODE</submitto>
         <program>Windows</program>
      </command>
   </command>
</CDServiceRequest>
```

This command request submits a process called MYPROC to a Connect:Direct system with two commands (operations) as part of the process:

- COPY directive
- RUNTASK directive

The following is an example of a composite command (SUBMIT) that is being sent through an XML request document:

```
<CDServiceRequest>
  <command name="PROCESS" action="submit" type="composite">
      <label>MYPROC</label>
      <snode>MYCDSERVER</snode>
      <wait>UNLIMITED</wait>
      <command name="SUBMIT" type="composite">
         <label>MYSUB</label>
         <file>C:\ConnectDirect\procs\Process4.cdp</file>
      </command>
  </command>
</CDServiceRequest>
```

Simple Commands

As the name indicates, a simple command is an API command that does not have any nested children. All Connect:Direct API commands can be submitted as simple commands.

The following is an example of a simple command (SELECT) that is being sent through an XML request document:

```
<?xml version="1.0" encoding="UTF-8"?>
<CDServiceRequest>
    <command name="statistics" action="select" type="simple">
          <limit>1</limit>
    </command>
    <command name="process" action="select" type="simple"/>
    <command name="tracing" action="select" type="simple"/>
</CDServiceRequest>
```

This command request invokes the following commands on the Connect:Direct server:

- Select Statistics with a limit parameter set to 1
- Select Process
- Select Tracing

The commands have various attributes that can be omitted, for example:

- The first command evaluates to select statistics with limit=1.
- The second command is simply a select process without any filter parameters.

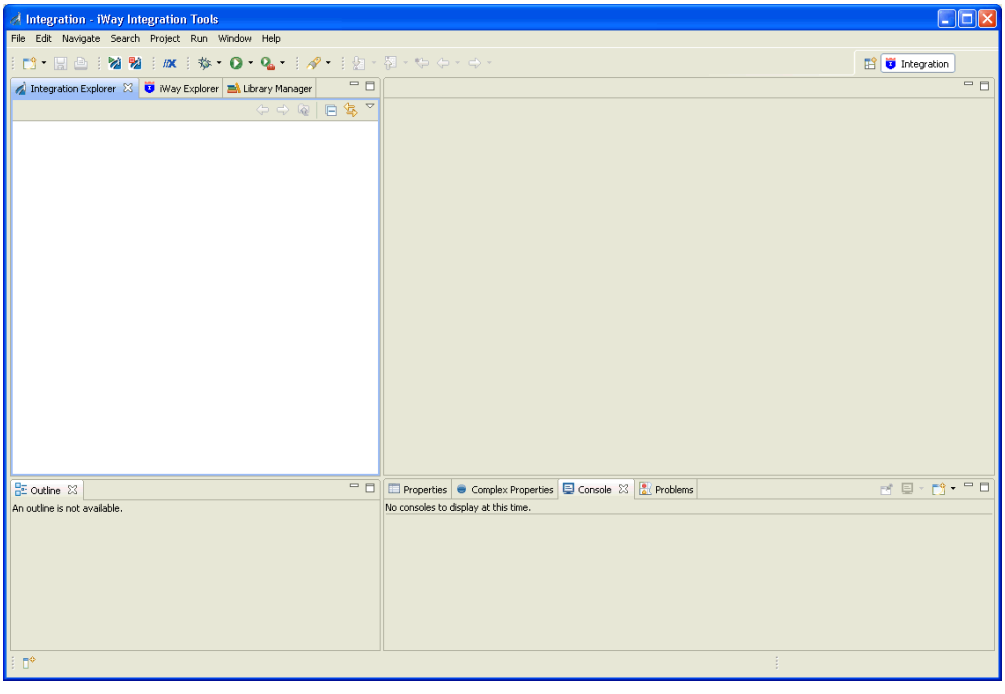
Although the current runtime implementation allows you to submit any API command, refer to the Connect:Direct API documentation to check the names of command parameters for any API command. Use the names as child elements for the required command element. The names will be used to build request schemas for the same.

Starting iWay Explorer

Procedure: How to Open iWay Integration Tools

1. Navigate to your local drive where you have iIT installed, and open the *eclipse* folder.
2. Double-click *iit.exe*.

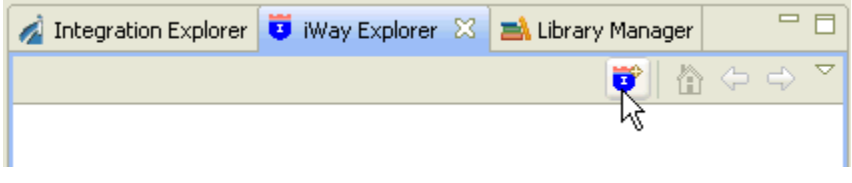
   iWay Integration Tools suite opens.



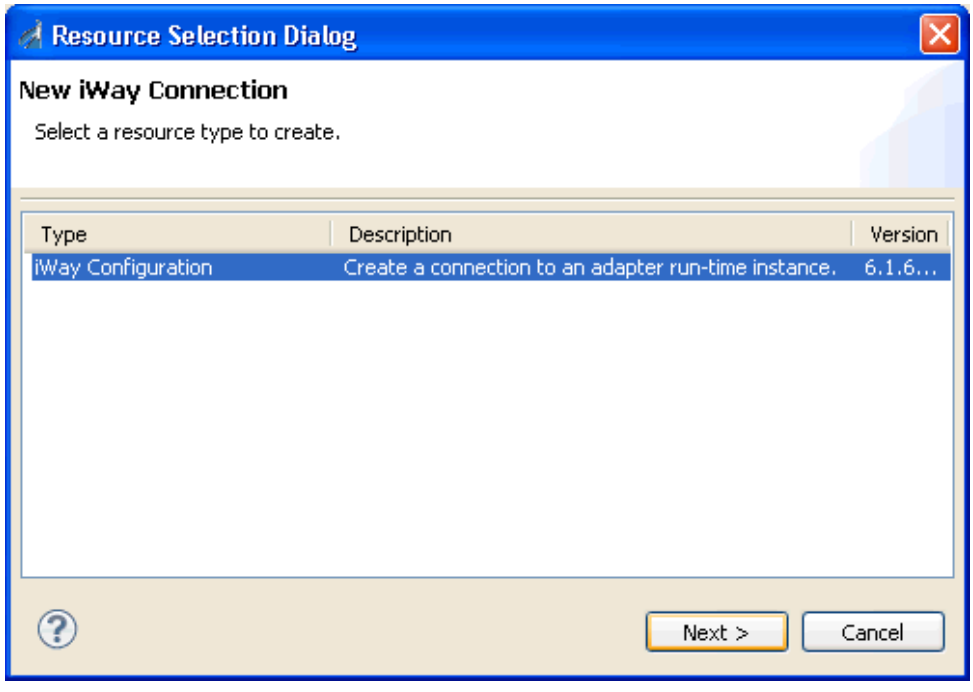
Procedure: How to Create an iWay Explorer Connection to an iSM Server

This procedure assumes that you have opened iWay Integration Tools (iIT) and are in the Workbench.

1. Click the *iWay Explorer* tab to make it active.
2. Click the *Launch iWay Resource Creator Wizard* button on the tool bar.
   In the following image, the iWay Explorer tab is active, and the cursor is pointing to the Launch iWay Resource Creator Wizard button.



When you click the button, the Resource Selection Dialog opens and displays the New iWay Connection pane, as shown in the following image.



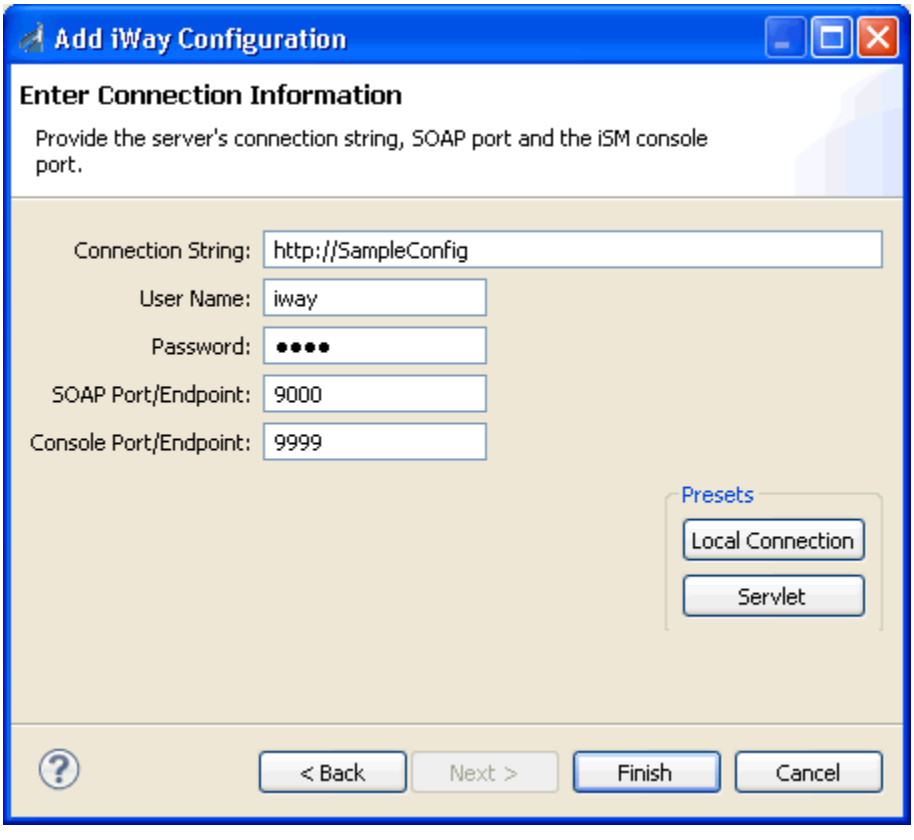
3. Under the Type heading, click *iWay Configuration*, which is the type of resource that you are going to create.
4. Click *Next*.

   The Add iWay Configuration dialog box opens and displays the Select Connection Types pane.
5. In the Configuration Alias field, type a name for the new configuration (for example, *SampleConfig*).

   Tip: The name that you supply is used only for display purposes in the tree. It is not a server connection property.
6. For Connection Type, click the radio button for the method that you are using to connect to iSM.
7. Optionally, select the *Connect to Host upon Wizard Completion* check box if you want iWay Explorer to automatically connect to this instance of iSM after you have created it. If you select this option, all the explorer environments under the new iSM connection are automatically connected to iSM when this procedure is finished.

If you do not select this option, the explorer environments are not automatically connected to iSM. You can connect to an individual explorer environment when you want to access it.

8. Click *Next* to continue the procedure.
9. If you selected an HTTP Connection, the Enter Connection Information pane opens, as shown in the following image.



- Verify the values in the three fields, or type the valid value or values.
  - The Connection String field contains the URL that connects to the iSM.
  - The SOAP Port/Endpoint field contains the SOAP port number.
  - The Console Port/Endpoint field contains the port number that the iSM Administration Console is listening on.
- Optionally, under Presets, click *Local Connection* to insert values for a local default iSM connection in the fields, or click *Servlet* to insert values for a sample servlet connection.

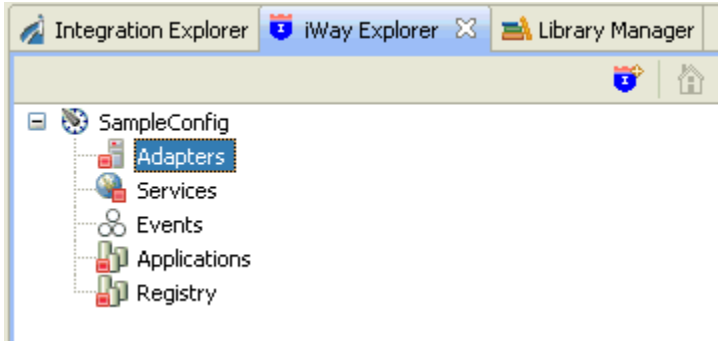
- Click *Finish*.
- In the File Path field, browse to the full path for your iWay installation directory and insert the path in the field. This path is used to locate the iWay adapters and store the XML schemas. For example:

  ```
  C:\Program Files\iWay7
  ```

- In the Configuration Name field, verify the name of your iWay server configuration. The base configuration is specified by default. Alternatively, you can type the name of the server configuration.
- Click *Finish*.

The new iSM connection is added to the tree on the iWay Explorer tab.

In the following image, an iSM connection named SampleConfig was added to iWay Explorer. The tree is expanded to show the five explorer environments that are available.



Adding the Connect:Direct Adapter to iWay Explorer

iWay Explorer supports access to many different application systems. When you connect to and expand the Adapters node, the iWay adapters for the supported application systems are displayed. They are the iWay adapters that you have installed and are licensed to use.

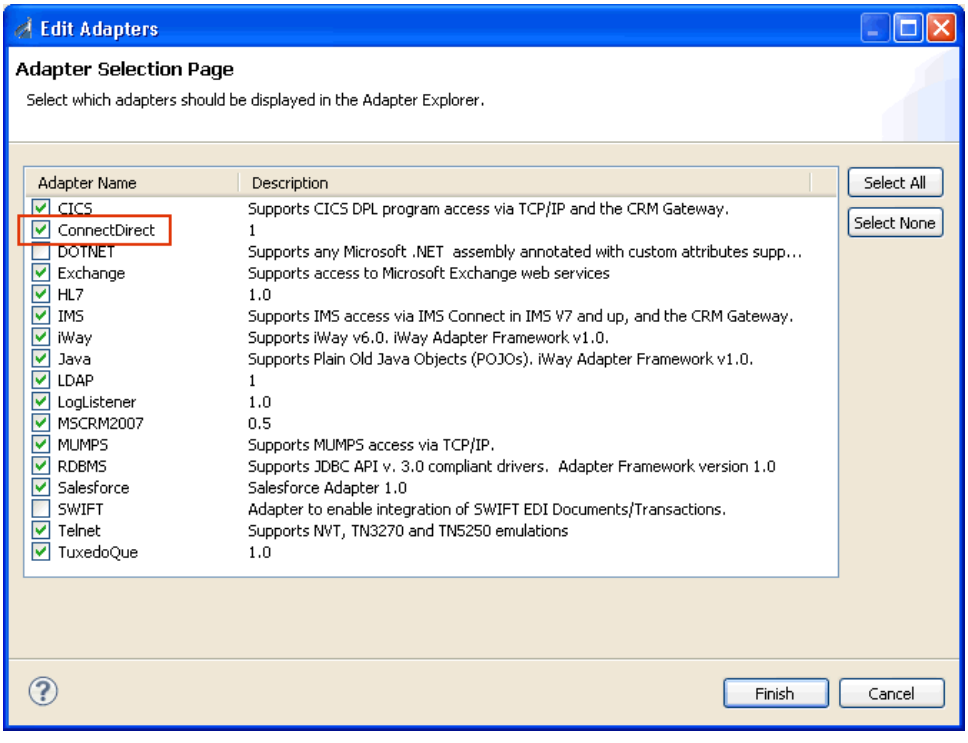
Procedure: How to Add the Connect:Direct Adapter to iWay Explorer

In this procedure, you are going to add the iWay Protocol Adapter for Connect:Direct to the list of adapters displayed in the Adapters node.

1. Right-click the *Adapters* node, and click *Edit* from the menu.

   The Edit Adapters dialog opens, prompting you to select the iWay adapter or adapters to add to iWay Explorer.

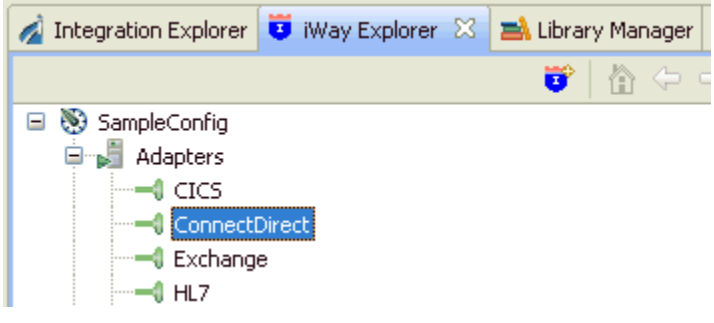
2. Select the check box for *ConnectDirect*, as shown in the following image.



3. Click *Finish*.

   The tree is automatically refreshed and displays the new adapter.

   In the following image, the ConnectDirect node is displayed in the Adapters node of iWay Explorer, as shown in the following image.



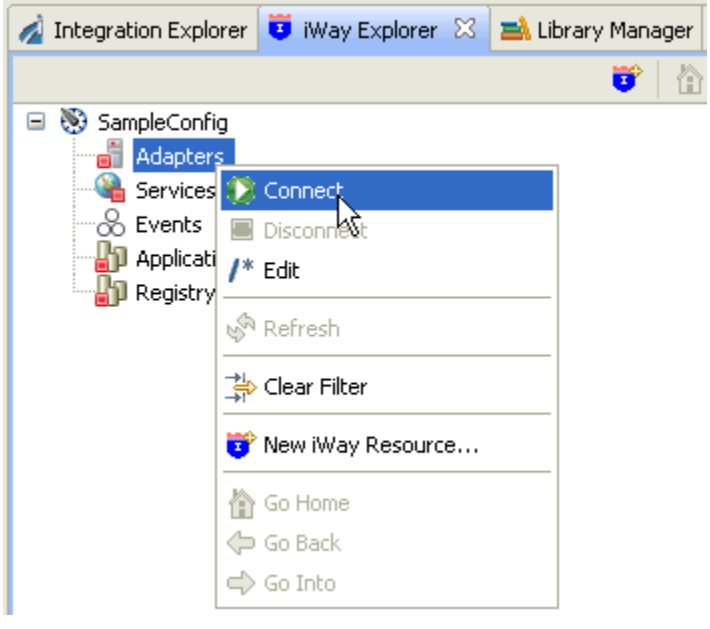
Working With a Target

To browse the business objects of an application system, you must create a target for that system. The target is the means by which you connect to the system. It contains the logon properties used to access the system.

Using the target, you must establish a connection to an application system every time you want to browse the system in iWay Explorer.

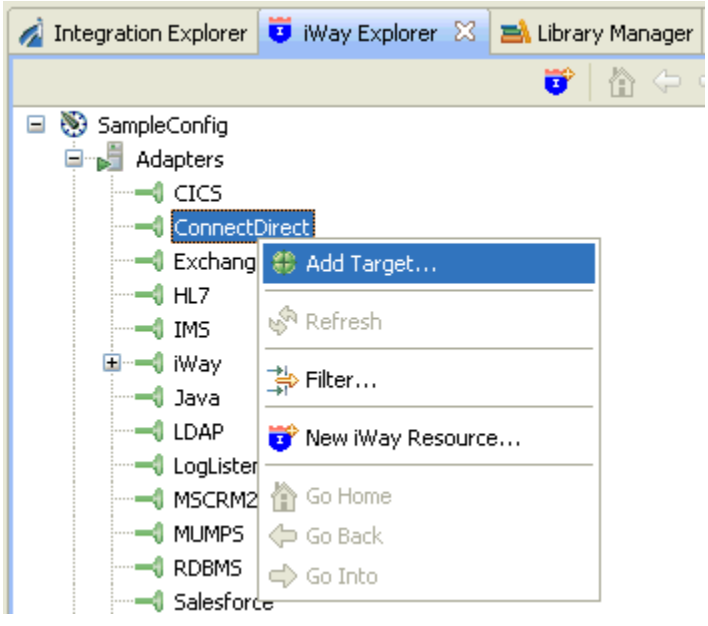
Procedure: How to Create a Target

1. Right-click the *Adapters* node, and click *Connect* from the menu, as shown in the following image.

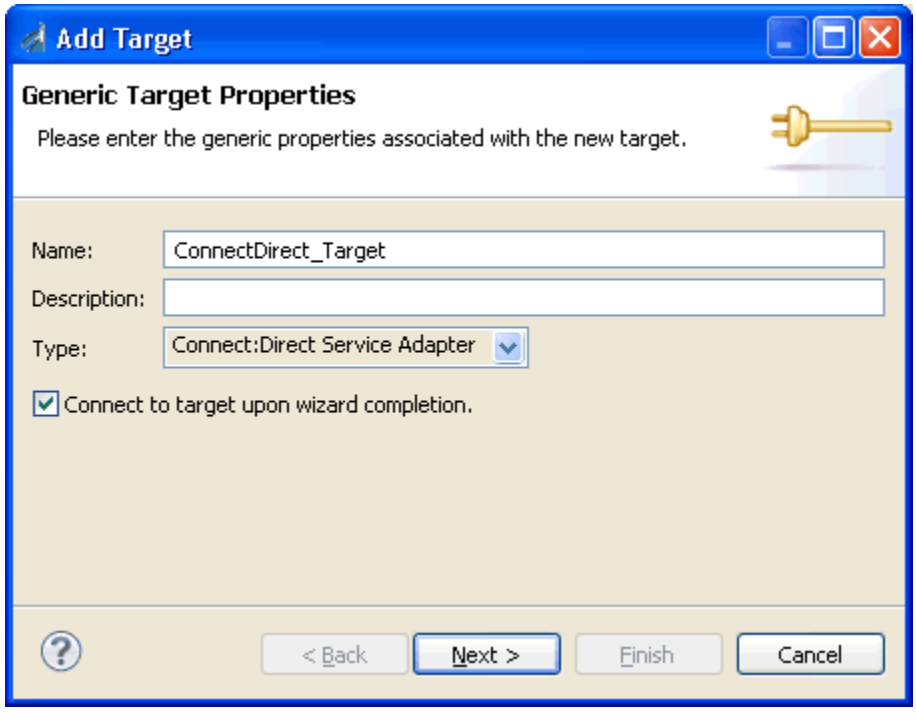


2. Once you are connected, expand the *Adapters* node.

3. Right-click *ConnectDirect*, and click *Add Target* from the menu, as shown in the following image.



The Add Target dialog opens and displays the Generic Target Properties pane, as shown in the following image.



4. Supply the values for the fields on the dialog box as follows.

   a. In the Name field, type a descriptive name for the target (for example, *ConnectDirect_Target*).

   b. In the Description field, optionally type a brief description of the target.

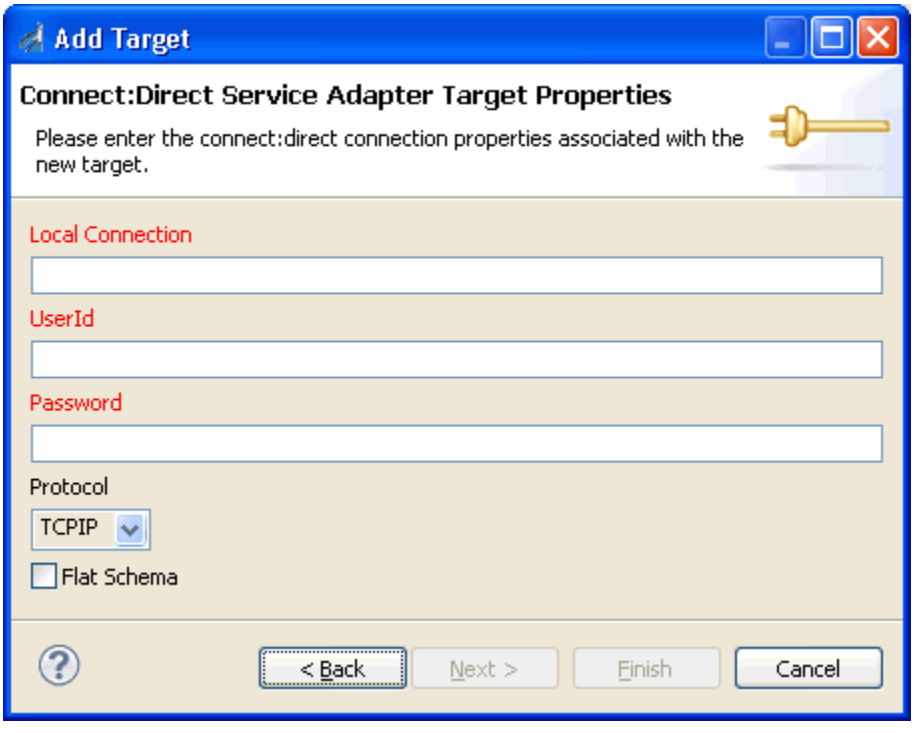
   c. From the Type drop-down list, select *Connect:Direct Service Adapter* (default).

5. Select the *Connect to target upon wizard completion* check box if you want iWay Explorer to automatically connect to this target after you have created it.

   If you deselect this option, iWay Explorer will not automatically connect to the target. From the tree, you can connect to an individual target when you want to access the associated application system.

6. Click *Next*.

The Add Target dialog opens and displays the Connect:Direct Service Adapter Target Properties pane, as shown in the following image.



7. Supply the connection information for the Connect:Direct target server to which you are connecting.

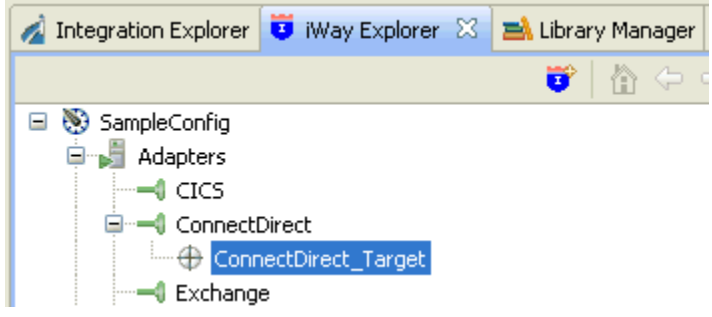
   The following table lists and describes the Connect:Direct target parameters that are available.

Parameter	Description
Local Connection	The name of the machine or IP address (and port) where the Connect:Direct server is being hosted.
UserId	The user name used to connect to the Connect:Direct server.
Password	The password that is associated with the user name.
Protocol	The type of protocol to use for the connection between the adapter and the server. The default protocol is TCP/IP.

Parameter	Description
Flat Schema	Select this option if you want flat XML schemas to be generated.

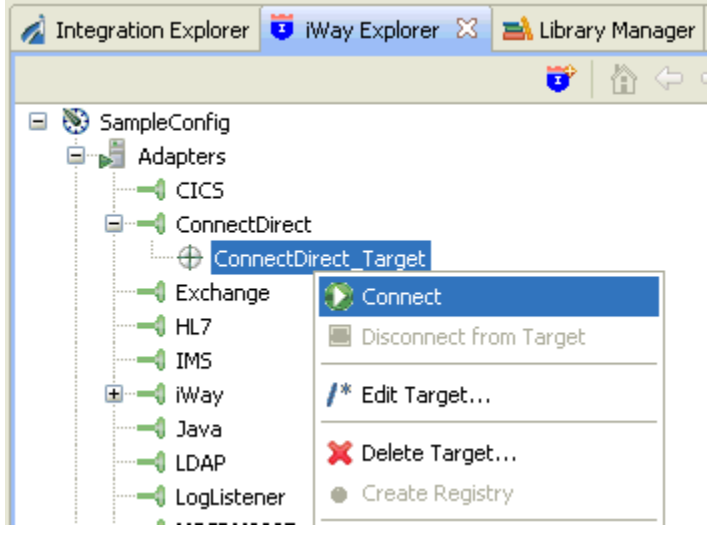
8. Click *Finish* when you are done.

   The new target is added to the Adapters node of iWay Explorer, as shown in the following image.

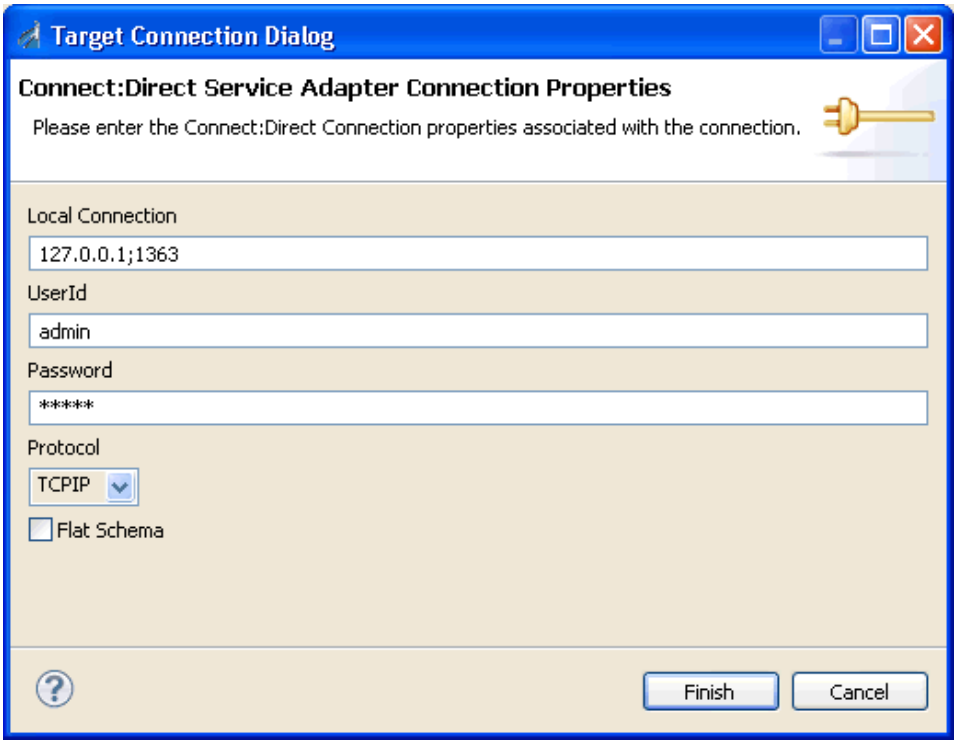


Procedure: How to Connect to a Target

1. Expand the *ConnectDirect* node to locate the name of the target that you want to connect to, for example, *ConnectDirect_Target*.
2. Right-click the target, and click *Connect* from the menu, as shown in the following image.

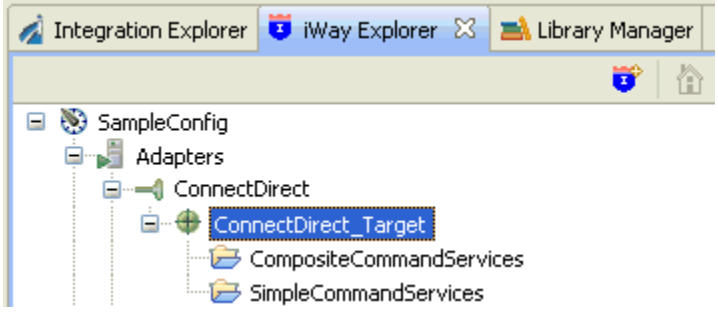


The Target Connection Dialog opens, as shown in the following image.



3. Enter a valid password for the configured target and click *Finish*.

   The ConnectDirect_Target node icon changes to green, and two folders are displayed (CompositeCommandServices and SimpleCommandServices), reflecting a successful connection. You can click a folder and then expand it to display its contents.



Procedure: How to Disconnect From a Target

Although you can maintain multiple open connections to different application systems, it is a good practice to close a connection when you are not using it.

1. In the tree, expand the *ConnectDirect* node to locate the name of the target from which you want to disconnect, for example, *ConnectDirect_Target*.
2. Right-click the target, and click *Disconnect from Target* from the menu.

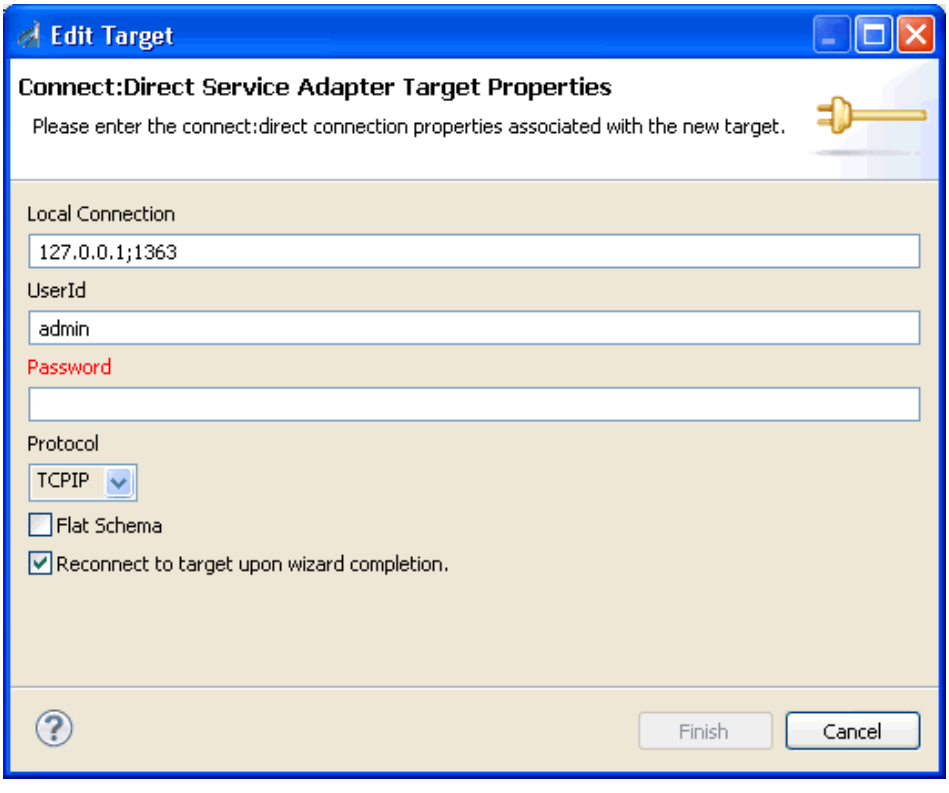
   The connection to the application system is closed.

Procedure: How to Edit a Target

After you create a target, you can edit the information that you provided during the creation procedure.

1. In the tree, expand the *ConnectDirect* node to locate the name of the target that you want to edit, for example, *ConnectDirect_Target*.
2. Right-click the target, and click *Edit Target* from the menu.

The Edit Target dialog opens and displays the Connect:Direct target properties, as shown in the following image.



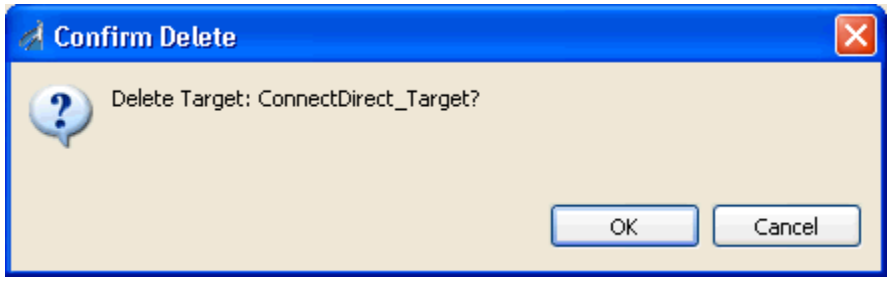
3. Modify the connection properties as required.
4. Optionally select the *Reconnect to target upon wizard completion* check box if you want iWay Explorer to automatically connect to this target after you have edited it. iWay Explorer will use the modified properties to connect.
5. Click *Finish* when you have made your edits.

Procedure: How to Delete a Target

You can delete a target that is no longer needed. You can delete it whether or not it is closed. If open, the target automatically closes before it is deleted.

1. In the tree, expand the *ConnectDirect* node to locate the name of the target that you want to delete, for example, *ConnectDirect_Target*.
2. Right-click the target, and click *Delete Target* from the menu.

iWay Explorer displays a prompt, asking you to confirm the deletion of the selected target, as shown in the following image.



3. Click *OK* to proceed with the deletion.

Viewing Application System Objects

After you create and connect to the target for an application system, iWay Explorer displays the application objects for that system. You can explore and browse the application object metadata.

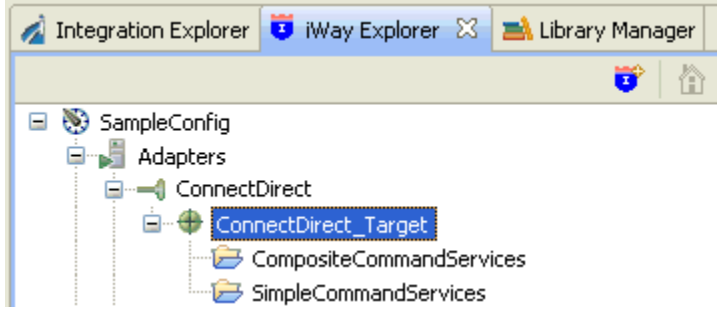
For example, for Connect:Direct, you can view metadata for Composite Command Services and Simple Command Services.

Procedure: How to View Application System Objects

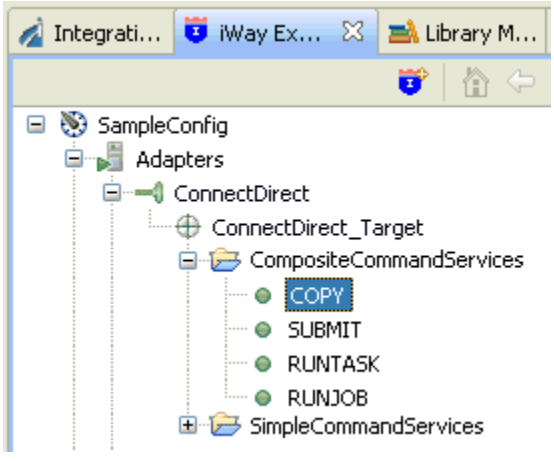
1. Expand the iWay Explorer tree to locate the name of the target for the application system whose objects you want to view. For example, for Connect:Direct, locate *ConnectDirect_Target*.

   Two folders are displayed beneath the target name:

   - CompositeCommandServices
   - SimpleCommandServices



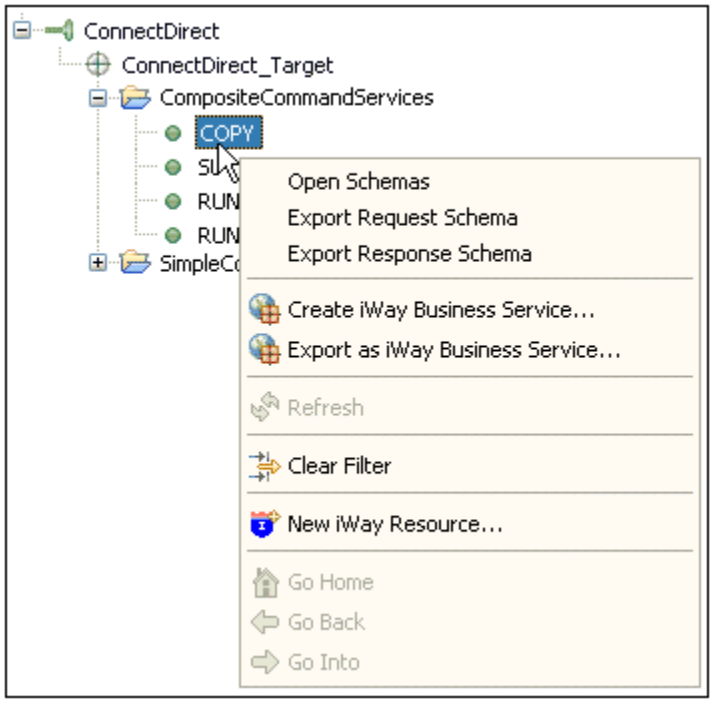
2. Click a desired folder, for example, *CompositeCommandServices*, and expand it, as shown in the following image.



A list of Composite Command Services are displayed.

3. Right-click the *COPY* service to display the menu options that are available.

The following image shows the options that are available.



Creating an XML Schema

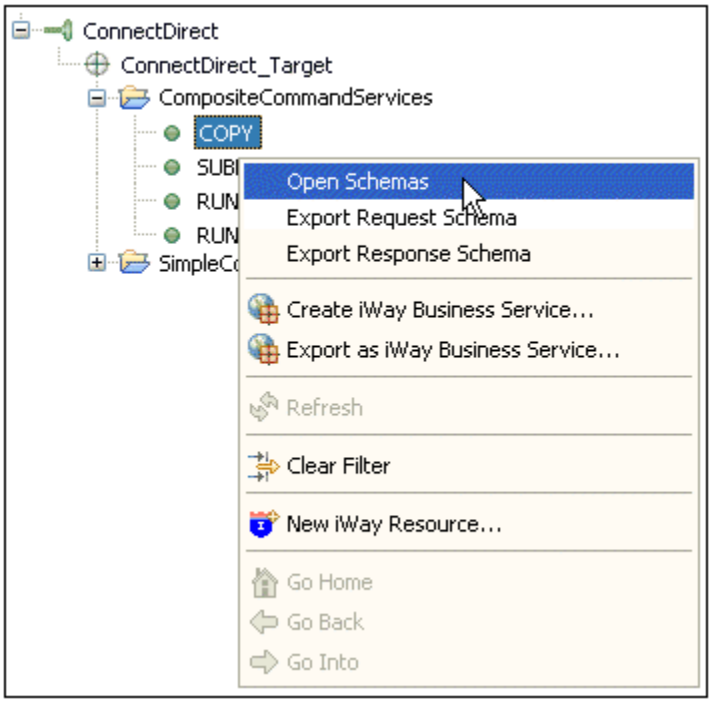
You can create XML request and response schemas for the Connect:Direct metadata that you want to use with your adapter. Optionally, you can store the schemas in a folder (directory) on your file system, using the iWay Explorer export feature.

Procedure: How to Create an XML Schema

1. Expand the connected target node and locate the method for which you want to create XML request and response schemas.

   For example, for Connect:Direct, expand *CompositeCommandServices* and select *COPY*.

2. Right-click *COPY*, and click *Open Schemas* from the menu, as shown in the following image.



iWay Explorer generates XML request and response schemas for the selected command. By default, the Response tab in the right pane is selected (active), and iWay Explorer displays the response schema in that pane.

3. In the right pane, click the *Request* tab to display the request schema.

   To view sample XML request and response schemas that are generated by iWay Explorer for the Connect:Direct adapter, see *Sample XML Request and Response Schemas* on page 47.

Procedure: How to Export an XML Schema

1. Right-click the Connect:Direct command whose schemas you want to export, for example, *COPY*.
2. From the menu, click either *Export Request Schema* or *Export Response Schema*.
3. In the Save As dialog box that opens, select the folder on your file system in which to store the exported schema. By default, iWay Explorer stores the file in your workspace folder, followed by the path that you specify on the Save As dialog box.

4. Type a name for the exported schema. By default, the file name extension is .xsd.
5. Click *OK* when you are done.

   iWay Explorer stores the exported schema in the folder that you selected, using the name that you supplied.

Creating Business Services

Business Service Explorer provides web developers with a simple, consistent mechanism for extending the capabilities of an iWay adapter. The iWay Business Services Provider (iBSP) exposes iWay functionality as a number of web services. It serves as a gateway to heterogeneous backend applications and databases.

A web service is a self-contained, modularized function that you can publish and access across a network using open standards. It is the implementation of an interface by a component, and is an executable entity. For the caller or sender, a web service can be considered a *black box* that may require input while typically delivering a result.

Web services integrate within an enterprise and across enterprises on any communication technology stack, whether asynchronous or synchronous, in any format.

Creating an iWay Business Service

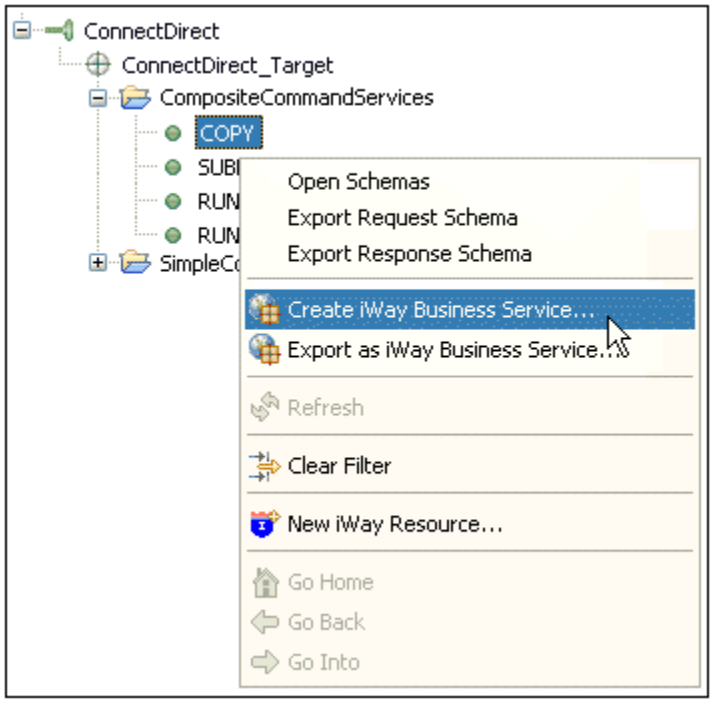
After you browse the business object repository for an application system, and generate XML schemas for an object that you want to use with an iWay adapter, you can create an iWay Business Service for that object.

The Web Service Description Language (WSDL) file is an XML file that describes the web service documents and provides access to the service. It specifies the location of the service and the operations (or methods) that the service exposes.

You can delete an iWay Business Service that you no longer need.

Procedure: How to Create an iWay Business Service

1. In the iWay Explorer tree, expand the target node to which you are connected and locate the method for which you want to create an iWay Business Service. For example:



2. Right-click the command, for example, *COPY*, and click *Create iWay Business Service* from the menu.

   The Create iWay Business Service dialog box opens, prompting you for information about the new service.

3. Supply the values for the fields on the dialog box as follows.

   a. From the Existing Service Names drop-down list, click *<new service>* if you want to create a new service name or select an existing service name.

   b. If you are creating a new service name, type the name in the Service Name field, for example, *COPY_Service*.

   c. In the Service Description field, optionally type a brief description of the new business service.

4. Click *Next*.

   The Add Business Service window in the Select Business License dialog box opens.

5. Supply the values for the fields on the dialog box as follows.

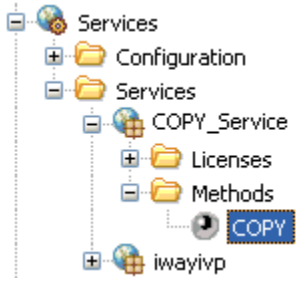
   a. From the License drop-down list, select the license definition that you want to use with this business service.

   b. In the Method Name field, accept the default value or type a descriptive name for the method that the service exposes (for example, COPY).

   c. In the Method Description field, optionally type a brief description of the method.

6. Click *Finish*.

   Business Service Explorer adds the new iWay Business Service beneath the Business Service Explorer node in the tree.



   The right pane displays the available licenses.

7. To test the new iWay Business Service, click the *test* link in the right pane.

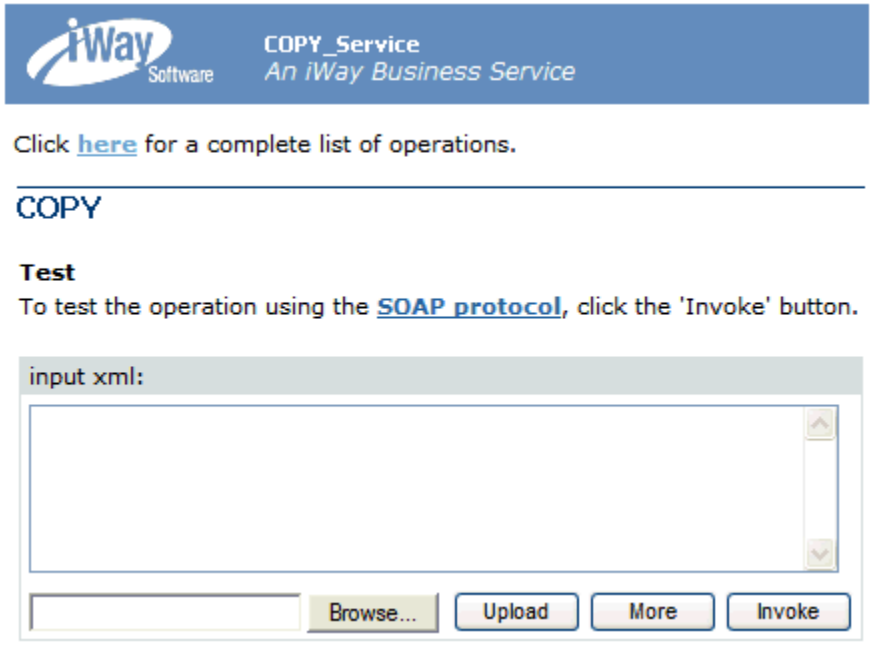
   The iWay Business Services that are licensed under test are displayed.

8. Click the *COPY_Service* link.

   The operations (methods) that are supported are displayed.

9. Click the *COPY* link.

The test pane for the COPY command opens.



10. In the input xml field, enter an XML request document that queries the iWay Business Service named COPY. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
 <CDServiceRequest>
   <command name="PROCESS" action="submit" type="composite">
     <label>MYPROC</label>
     <snode>MYCDSERVER</snode>
     <!--<wait>UNLIMITED</wait>-->
     <command name="COPY" type="composite">
        <label>MYCPY</label>
        <snode>MYCDSERVER</snode>
        <direction>SEND</direction>
        <source>c:\downloads\iWay_brochure.pdf</source>
        <destination>myout.pdf</destination>
        <disposition>NEW</disposition>
        <ckpt>2M</ckpt>
        <compress>extended</compress>
        <srcsysopts>1</srcsysopts>
        <destsysopts>2</destsysopts>
     </command>
</CDServiceRequest>
```

11. Click *Invoke*.

The result of the test is displayed in the right pane.

Procedure: How to Export a WSDL File

1. Connect to the Business Service Explorer and expand the tree to locate the name of the iWay Business Service whose WSDL file you want to export.
2. Right-click the name of the iWay Business Service, for example, *COPY*, and click *Export WSDL* from the menu.
3. In the Save As dialog box that opens, select the folder on your file system in which to store the exported WSDL file. By default, Business Service Explorer stores the file in your workspace folder followed by the path that you specify on the Save As dialog box.
4. Type a name for the exported WSDL file. By default, the file name extension is .wsdl.
5. Click *OK* when you are done.

   Business Service Explorer stores the exported WSDL file in the folder that you selected, using the name that you supplied.

Procedure: How to Delete an iWay Business Service

1. Connect to the Business Service Explorer, and expand the tree to locate the name of the iWay Business Service that you want to delete.
2. Right-click the name of the iWay Business Service, for example, *COPY*, and click *Delete* from the menu.
3. Business Service Explorer displays a prompt, asking you to confirm the deletion of the selected iWay Business Service.
4. Click *OK* to proceed with the deletion.

Sample XML Request and Response Schemas

As a reference, this section provides sample XML request and response schemas for Connect:Direct composite and simple commands.

Composite Submit Request and Response Schema

The following is a sample XML request and response schema for the composite Submit command.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:iwaysoftware:adapter:iwconnectdirect:composite:submit"
   xmlns:tns="urn:iwaysoftware:adapter:iwconnectdirect:composite:submit"
   xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
   attributeFormDefault="unqualified">
  <xs:complexType name="command">
    <xs:sequence>
      <xs:element name="label" type="xs:string"/>
      <xs:element name="snode">
        <xs:simpleType>
          <xs:restriction base="xs:string"/>
        </xs:simpleType>
      </xs:element>
      <xs:element name="class" type="xs:string" minOccurs="0"/>
      <xs:element name="prty" type="xs:string" minOccurs="0"/>
      <xs:element name="startt" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string"/>
        </xs:simpleType>
      </xs:element>
      <xs:element name="retain" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="No"/>
            <xs:enumeration value="Yes"/>
            <xs:enumeration value="Initial"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="hold" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="No"/>
            <xs:enumeration value="Yes"/>
            <xs:enumeration value="Call"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
```

```
            <xs:element name="crc" minOccurs="0">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="OFF"/>
                  <xs:enumeration value="ON"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="file" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="name" type="xs:string" use="required"
        fixed="SUBMIT"/>
          <xs:attribute name="action" type="xs:string" use="optional"/>
          <xs:attribute name="type" type="xs:string" use="required"
        fixed="composite"/>
          </xs:complexType>
          <xs:element name="command" type="tns:command"/>
        </xs:schema>
```

Composite Copy Request and Response Schema

The following is a sample XML request and response schema for the composite Copy command.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:iwaysoftware:adapter:iwconnectdirect:composite:copy"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:tns="urn:iwaysoftware:adapter:iwconnectdirect:composite:copy"
   elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:complexType name="command">
    <xs:sequence>
      <xs:element name="label" type="xs:string"/>
      <xs:element name="direction">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="SEND"/>
            <xs:enumeration value="RECEIVE"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="source" type="xs:string"/>
      <xs:element name="destination" type="xs:string"/>
      <xs:element name="disposition" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="NEW"/>
```

```
        <xs:enumeration value="RPL"/>
        <xs:enumeration value="MOD"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="ckpt" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:element>
  <xs:element name="compress" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:element>
  <xs:element name="srcsysopts" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:element>
  <xs:element name="destsysopts" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:element>
  <xs:element name="dcb" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:element>
  <xs:element name="SRCSYSOPTS" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:element>
  <xs:element name="DESTSYSOPTS" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:element>
  <xs:element name="DCB" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
    </xs:simpleType>
  </xs:element>
  <xs:element name="COMPRESS" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:string"/>
```

```
            </xs:simpleType>
          </xs:element>
          <xs:element name="DISPOSITION" minOccurs="0">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="NEW"/>
                <xs:enumeration value="RPL"/>
                <xs:enumeration value="MOD"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="name" type="xs:string" use="required" fixed="COPY"/>
        <xs:attribute name="action" type="xs:string" use="optional"/>
        <xs:attribute name="type" type="xs:string" use="required"
fixed="composite"/>
      </xs:complexType>
      <xs:element name="command" type="tns:command"/>
</xs:schema>
```

Note: Composite commands must be wrapped with a parent process command, as shown in the following schema, when the request is submitted to the adapter.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:nscopy="urn:iwaysoftware:adapter:iwconnectdirect:composite:copy"
xmlns:nssubmit="urn:iwaysoftware:adapter:iwconnectdirect:composite:submit"
xmlns:nsruntask="urn:iwaysoftware:adapter:iwconnectdirect:composite:runtask"
xmlns:nsrunjob="urn:iwaysoftware:adapter:iwconnectdirect:composite:runjob"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:import namespace="urn:iwaysoftware:adapter:iwconnectdirect:composite:copy"
      schemaLocation="composite_copy.xsd"/>
  <xs:import namespace="urn:iwaysoftware:adapter:iwconnectdirect:composite:submit"
      schemaLocation="composite_submit.xsd"/>
  <xs:import namespace="urn:iwaysoftware:adapter:iwconnectdirect:composite:runtask"
      schemaLocation="composite_runtask.xsd"/>
```

```
   <xs:import
namespace="urn:iwaysoftware:adapter:iwconnectdirect:composite:runjob"
      schemaLocation="composite_runjob.xsd"/>
  <xs:element name="command" type="commandType">
    <xs:annotation>
      <xs:documentation>Comment describing your root element</
xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="commandType">
    <xs:sequence>
      <xs:element name="label" type="xs:string"/>
      <xs:element name="snode">
        <xs:simpleType>
          <xs:restriction base="xs:string"/>
        </xs:simpleType>
      </xs:element>
      <xs:element name="wait" type="xs:string" minOccurs="0"/>
      <xs:element name="class" type="xs:string" minOccurs="0"/>
      <xs:element name="prty" type="xs:string" minOccurs="0"/>
      <xs:element name="startt" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="NEW"/>
            <xs:enumeration value="RPL"/>
            <xs:enumeration value="MOD"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="retain" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="No"/>
            <xs:enumeration value="Yes"/>
            <xs:enumeration value="Initial"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="hold" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="No"/>
            <xs:enumeration value="Yes"/>
            <xs:enumeration value="Call"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
```

```
      <xs:element name="crc" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="OFF"/>
            <xs:enumeration value="ON"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element ref="nscopy:command" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="nssubmit:command" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element ref="nsrunjob:command" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element ref="nsruntask:command" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"
fixed="PROCESS"/>
    <xs:attribute name="action" type="xs:string" use="optional"/>
    <xs:attribute name="type" type="xs:string" use="required"
fixed="composite"/>
  </xs:complexType>
  <xs:element name="CDServiceRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="command"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Simple Statistics Request and Response Schemas

The following is a sample XML request schema for the simple statistics Select command.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:iwaysoftware:adapter:iwconnectdirect:statistics:select"
    xmlns:tns="urn:iwaysoftware:adapter:iwconnectdirect:statistics:select"
    xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:element name="command" type="tns:commandType">
    <xs:annotation>
      <xs:documentation>Comment describing your root element</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="commandType">
    <xs:sequence>
      <xs:element name="limit" type="xs:string" minOccurs="0"/>
      <xs:element name="pname" type="xs:string" minOccurs="0"/>
      <xs:element name="pnumber" type="xs:string" minOccurs="0"/>
      <xs:element name="reccat" type="xs:string" minOccurs="0"/>
      <xs:element name="recids" type="xs:string" minOccurs="0"/>
      <xs:element name="snode" type="xs:string" minOccurs="0"/>
      <xs:element name="startt" type="xs:string" minOccurs="0"/>
      <xs:element name="stopt" type="xs:string" minOccurs="0"/>
      <xs:element name="submitter" type="xs:string" minOccurs="0"/>
      <xs:element name="sfile" type="xs:string" minOccurs="0"/>
      <xs:element name="dfile" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required" fixed="statistics"/>
    <xs:attribute name="action" type="xs:string" use="required" fixed="select"/>
    <xs:attribute name="type" type="xs:string" use="required" fixed="simple"/>
  </xs:complexType>
  <xs:element name="CDServiceRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="tns:command" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

The following is a sample XML response schema for the simple statistics Select command.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="urn:iwaysoftware:adapter:iwconnectdirect:statistics:response"
    xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="CDServiceResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="CDCommand">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CDCommandScript" type="xs:string"/>
              <xs:element name="CDServiceResponseDetail">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="LogDateTime" type="xs:string"/>
                    <xs:element name="RecordCategory" type="xs:string"/>
                    <xs:element name="RecordID" type="xs:string"/>
                    <xs:element name="CondCode" type="xs:boolean"/>
                    <xs:element name="Feedback" type="xs:string"/>
                    <xs:element name="MsgID" type="xs:string"/>
                    <xs:element name="MsgShortText" type="xs:string"/>
                    <xs:element name="ProcName" type="xs:string"/>
                    <xs:element name="ProcNumber" type="xs:boolean"/>
                    <xs:element name="SubmitNode" type="xs:string"/>
                    <xs:element name="Submitter" type="xs:string"/>
                    <xs:element name="SubmitterNode" type="xs:string"/>
                    <xs:element name="StepName" type="xs:string"/>
                    <xs:element name="PNode" type="xs:string"/>
                    <xs:element name="SNode" type="xs:string"/>
                    <xs:element name="Retain" type="xs:string"/>
                    <xs:element name="Class" type="xs:string"/>
                    <xs:element name="Priority" type="xs:string"/>
                    <xs:element name="ExecPriority" type="xs:string"/>
                    <xs:element name="StdCompression" type="xs:string"/>
                    <xs:element name="ExtCompression" type="xs:string"/>
                    <xs:element name="Checkpoint" type="xs:string"/>
                    <xs:element name="LinkFail" type="xs:string"/>
                    <xs:element name="Translation" type="xs:string"/>
```

```
<xs:element name="SubDateTime" type="xs:string"/>
<xs:element name="SchDateTime" type="xs:string"/>
<xs:element name="LocalNode" type="xs:string"/>
<xs:element name="FromNode" type="xs:string"/>
<xs:element name="Queue" type="xs:string"/>
<xs:element name="Restart" type="xs:string"/>
<xs:element name="Status" type="xs:string"/>
<xs:element name="Function" type="xs:string"/>
<xs:element name="PNodePlexclass" type="xs:string"/>
<xs:element name="SNodePlexclass" type="xs:string"/>
<xs:element name="Debug" type="xs:string"/>
<xs:element name="SrcFile" type="xs:string"/>
<xs:element name="SrcDisp1" type="xs:string"/>
<xs:element name="SrcDisp2" type="xs:string"/>
<xs:element name="SrcDisp3" type="xs:string"/>
<xs:element name="DestFile" type="xs:string"/>
<xs:element name="DestDisp1" type="xs:string"/>
<xs:element name="DestDisp2" type="xs:string"/>
<xs:element name="DestDisp3" type="xs:string"/>
<xs:element name="MemberName" type="xs:string"/>
<xs:element name="SourceMemberName" type="xs:string"/>
<xs:element name="TargetMemberName" type="xs:string"/>
<xs:element name="AliasMemberName" type="xs:string"/>
<xs:element name="Sysopts" type="xs:string"/>
<xs:element name="BytesRead" type="xs:string"/>
<xs:element name="RecordsRead" type="xs:string"/>
<xs:element name="BytesSent" type="xs:string"/>
<xs:element name="RUsSent" type="xs:string"/>
<xs:element name="BytesWritten" type="xs:string"/>
<xs:element name="RecordsWritten" type="xs:string"/>
<xs:element name="BytesReceived" type="xs:string"/>
<xs:element name="RUsReceived" type="xs:string"/>
<xs:element name="RUSize" type="xs:string"/>
<xs:element name="LocalCondCode" type="xs:string"/>
<xs:element name="LocalMsgID" type="xs:string"/>
<xs:element name="OtherCondCode" type="xs:string"/>
<xs:element name="OtherMsgID" type="xs:string"/>
<xs:element name="PNodeAcctInfo" type="xs:string"/>
<xs:element name="SNodeAcctInfo" type="xs:string"/>
<xs:element name="Hold" type="xs:string"/>
```

```
                    <xs:element name="SecureEnabled" type="xs:string"/>
                    <xs:element name="PNodeEncAlgList" type="xs:string"/>
                    <xs:element name="PNodeEncData" type="xs:string"/>
                    <xs:element name="SNodeEncAlgList" type="xs:string"/>
                    <xs:element name="SNodeEncData" type="xs:string"/>
                    <xs:element name="CBEncAlg" type="xs:string"/>
                    <xs:element name="MergeEA" type="xs:string"/>
                    <xs:element name="PNodeSign" type="xs:string"/>
                    <xs:element name="SNodeSign" type="xs:string"/>
                    <xs:element name="MergeSign" type="xs:string"/>
                    <xs:element name="CurSignVerified" type="xs:string"/>
                    <xs:element name="PrevSignVerified" type="xs:string"/>
                    <xs:element name="ServerName" type="xs:string"/>
                    <xs:element name="StartDateTime" type="xs:string"/>
                    <xs:element name="StopDateTime" type="xs:string"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Note: The Statistics command must be enclosed within a parent *CDServiceRequest* node.

Chapter 5

Configuring Event Handling for the iWay Protocol Adapter for Connect:Direct

This section describes how to configure event handling for the iWay Protocol Adapter for Connect:Direct.

In this chapter:

Connect:Direct Event Handling Overview

Event handling for the iWay Protocol Adapter for Connect:Direct can be configured to:

- Generate events based on Processes.
- Generate events based on Statistics.
- Generate events based on monitoring a directory.

Any configuration property can be set to seed itself from an SQL statement. The SQL statement is evaluated when the adapter is activated and the properties are used in the underlying CD command.

The following is a sample output when using the Statistics mode:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CDEvent>
  <CDCommand>
     <CDCommandScript>select STATISTICS limit=3</CDCommandScript>
     <CDEventDetail>
        <LogDateTime>Tue Nov 20 00:04:39 EST 2007</LogDateTime>
        <RecordCategory>CAEV</RecordCategory>
        <RecordID>LSDI</RecordID>
        <CondCode>8</CondCode>
        <Feedback>10054</Feedback>
        <MsgID>LSDI010I</MsgID>
        <MsgShortText>The call to recv() in sdipc_srvr_recv() failed
        </MsgShortText>
        <ProcNumber>0</ProcNumber>
     </CDEventDetail>
     <CDEventDetail>
       <LogDateTime>Tue Nov 20 00:04:39 EST 2007</LogDateTime>
       <RecordCategory>CAEV</RecordCategory>
       <RecordID>LIPT</RecordID>
       <CondCode>8</CondCode>
       <Feedback>0</Feedback>
       <MsgID>LIPT027I</MsgID>
       <MsgShortText>The call to recv() failed. OS message=&amp;amp;OSMSG.
       </MsgShortText>
       <ProcNumber>0</ProcNumber>
     </CDEventDetail>
     <CDEventDetail>
      <LogDateTime>Tue Nov 20 00:04:39 EST 2007</LogDateTime>
      <RecordCategory>CAEV</RecordCategory>
      <RecordID>CMOT</RecordID>
      <CondCode>8</CondCode>
      <Feedback>10054</Feedback>
      <MsgID>LCOT000I</MsgID>
      <MsgShortText>The comm thread termination has started.
      </MsgShortText>
      <ProcNumber>0</ProcNumber>
      <StartDateTime>Tue Nov 20 00:04:39 EST 2007</StartDateTime>
     </CDEventDetail>
  </CDCommand>
</CDEvent>
```

The event schema will be provided. However, the event XML message itself will only contain elements that are not null or empty.

Understanding Ports and Channels

Events are generated as a result of activity in an application system. You can use an event to trigger an action in your own application.

To consume events, you must configure an iWay listener to capture the event within the application system (for example, Connect:Direct) and transmit the event notification to your own application.

After you create a connection to an application system, you can add events using the Events node in iWay Explorer. To add an event, you must create a port and a channel.

- **Port.** A port associates a particular business object exposed by an adapter with a particular disposition. A disposition defines the protocol and location of the event data. The port defines the end point of the event consumption.
- **Channel.** A channel represents configured connections to particular instances of backend systems or other types of systems. A channel binds one or more ports to a particular listener managed by an iWay adapter.

Creating a Port

The procedures in this topic describe how to create, edit, or delete a port using the Events node in iWay Explorer.

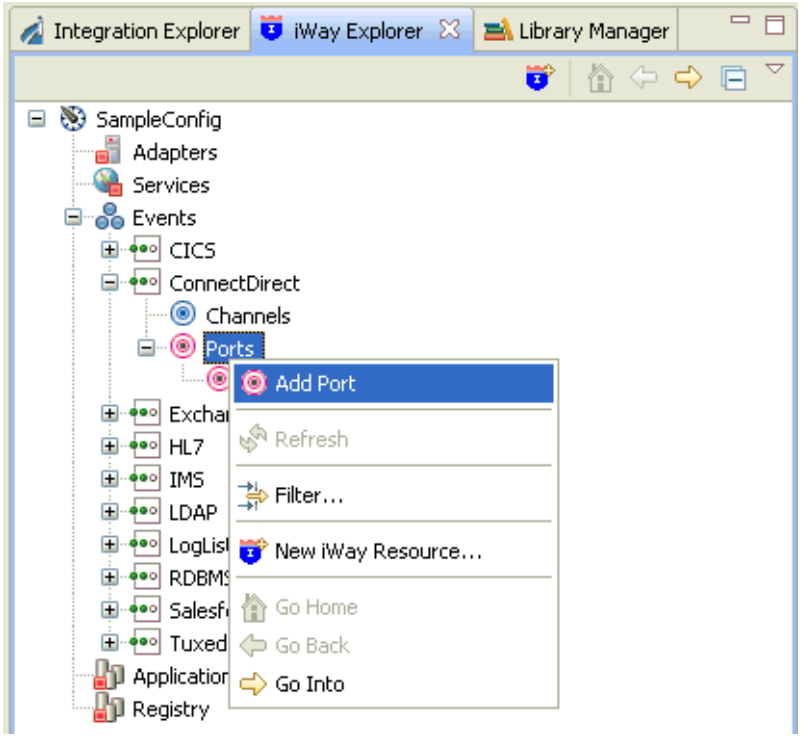
When you the Events node in iWay Explorer with an iWay Business Services Provider (iBSP) implementation, the following port dispositions are available.

- **File.** The File disposition uses a file URL to specify the destination file name or directory in which the event document will be written. During run time, the destination file name may require indexing to avoid overwriting.
- **iBSE.** The iBSE disposition enables an event to launch a business service method.
- **MSMQ.** The Microsoft Message Queuing (MSMQ) disposition supports public and private queues.
- **JMSQ.** The JMSQ disposition allows an event to be added to a JMS queue.
- **SOAP.** The SOAP disposition allows an event to launch a business service specified by a WSDL file. A SOAP action is optional. A double quote ("") is the default value.
- **HTTP.** The HTTP disposition uses an HTTP URL to specify an HTTP endpoint to which an event document is posted.

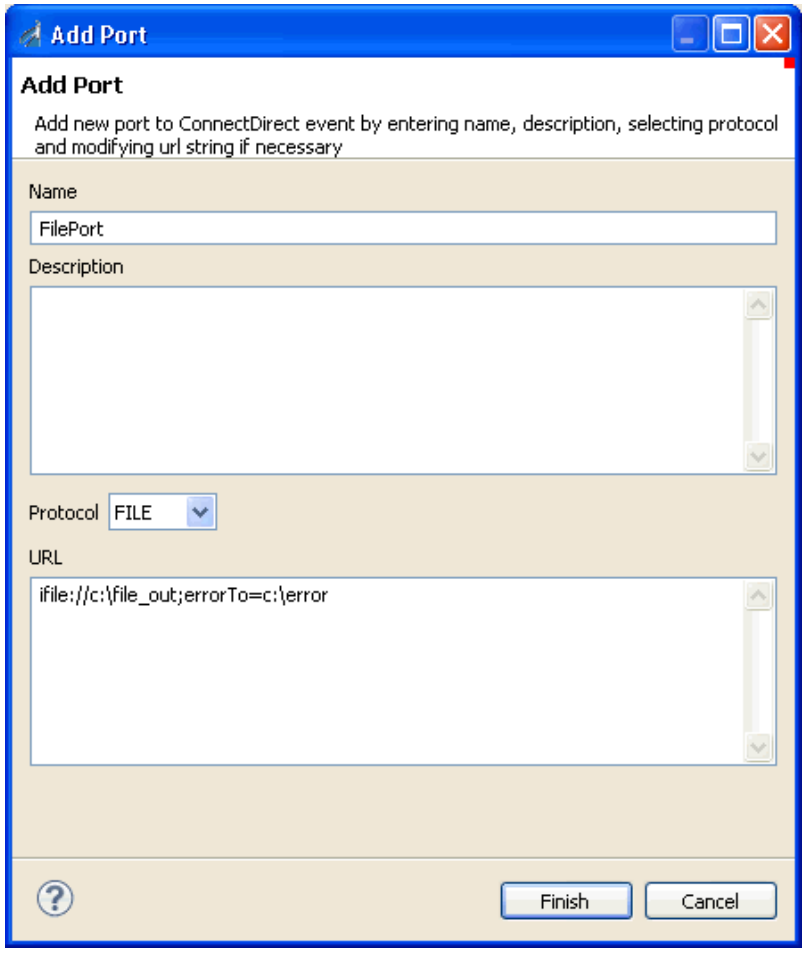
Procedure: How to Create a Port for the File Disposition

1. Connect to the Events node in iWay Explorer and expand the node for the application system (for example, *ConnectDirect*).

2. Right-click the *Ports* node, and click *Add Port* from the menu.



The following image shows the Add Port dialog box where you can supply information about the port.



a. In the Name field, type a name for the port.

b. In the Description field, optionally type a brief description.

c. From the Protocol drop-down list, select *FILE*.

d. In the URL field, type a File destination to which event data is written.

   When pointing the Events node in iWay Explorer to an iBSP deployment, specify the File destination using the following format:

```
ifile://[location];errorTo=[pre-defined port name or another
disposition url]
```

The following table describes the parameters for the File disposition.

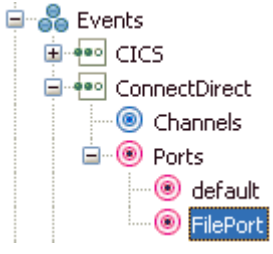
Parameter	Description
location	Destination and file name of the document where event data is written.
errorTo	Predefined port name or another disposition URL where error logs are sent. Optional.

For example:

```
ifile://D:\in\x.txt;errorTo=ifile://D:\error
```

3. Click *Finish* when you have supplied the values on the Add Port dialog box.

   The following image shows the port that you added, which appears beneath the Ports node in the Events area of iWay Explorer. In this example, the port is named FilePort.



Procedure: How to Create a Port for the IBSE Disposition

1. Connect to the Events node in iWay Explorer and expand the node for the application system (for example, *ConnectDirect*).
2. Right-click the *Ports* node, and click *Add Port* from the menu.

   The Add Port dialog box opens.
3. Supply the values for the fields on the dialog box as follows.
   a. In the Name field, type a name for the port.
   b. In the Description field, optionally type a brief description.
   c. From the Protocol drop-down list, select *IBSE*.
   d. In the URL field, type an iBSE destination using the following format:

```
ibse:[svcName].[mthName];responseTo=[pre-defined port name or
another disposition url];errorTo=[pre-defined port name or another
disposition url]
```

The following table describes the parameters for the IBSE disposition.

Parameter	Description
svcName	Name of the business service created with iBSE.
mthName	Name of the method created for the business service.
responseTo	Location where responses to the business service are posted. Predefined port name or another full URL. Optional.
errorTo	Location where error documents are sent. Predefined port name or another full URL. Optional.

4. Click *Finish* when you have supplied the values on the Add Port dialog box.

   The port that you added is displayed beneath the Ports node.

Procedure: How to Create a Port for the MSMQ Disposition

1. Connect to the Events node in iWay Explorer and expand the node for the application system (for example, *ConnectDirect*).
2. Right-click the *Ports* node, and click *Add Port* from the menu.

   The Add Port dialog box opens.
3. Supply the values for the fields on the dialog box as follows.
   a. In the Name field, type a name for the port.
   b. In the Description field, optionally type a brief description.
   c. From the Protocol drop-down list, select *MSMQ*.
   d. In the URL field, type an MSMQ destination in the following format:

   ```
   msmq://[machineName]/private$/[qName];errorTo=[pre-defined port
   name or another disposition url]
   ```

The following table describes the parameters for the MSMQ disposition.

Parameter	Description
machineName	Name of the machine on which the Microsoft Message Queuing system is running.
queue type	For private queues, type *Private$*. Private queues are queues that are not published in the Active Directory. They appear only on the local computer that contains them. Private queues are accessible only by Microsoft Message Queuing applications that recognize the full path name or format name of the queue.
qName	Name of the private queue where messages are placed.
errorTo	Location where error documents are sent. Predefined port name or another full URL. Optional.

4. Click *Finish* when you have supplied the values on the Add Port dialog box.

   The port that you added is displayed beneath the Ports node.

Procedure: How to Create a Port for the JMSQ Disposition

1. Connect to the Events node in iWay Explorer and expand the node for the application system (for example, *ConnectDirect*).
2. Right-click the *Ports* node, and click *Add Port* from the menu.

   The Add Port dialog box opens.
3. Supply the values for the fields on the dialog box as follows.
   a. In the Name field, type a name for the port.
   b. In the Description field, optionally type a brief description.
   c. From the Protocol drop-down list, select *JMSQ*.
   d. In the URL field, type a Java Message Service (JMS) destination.

      When pointing the Events node in iWay Explorer to an iBSP deployment, use the following format:

      ```
      jmsq:[myQueueName]@[myQueueFac];jndiurl=[myurl];jndifactory=
      [myfactory];user=[user];password=[xxx];errorTo=[pre-defined port
      name or another disposition url]
      ```

      The following table describes the parameters for the JMSQ disposition.

Parameter	Description
myQueueName or jmsqueue	Name of a queue to which events are emitted.
myQueueFac or jmsfactory	A resource that contains information about the JMS Server.
jndiurl	The URL used to contact the JNDI provider. The syntax of this URL depends on the JNDI provider that is used. This value corresponds to the following standard JNDI property: `java.naming.provider.url`
jndifactory	Is JNDI context.INITIAL_CONTEXT_FACTORY, which is provided by the JNDI service provider.
user	User ID associated with this queue.
password	Password for the user ID.
errorTo	Location where error logs are sent. Optional. Predefined port name or another disposition URL. The URL must be complete, including the protocol.

4. Click *Finish* when you have supplied the values on the Add Port dialog box.

   The port that you added is displayed beneath the Ports node.

Procedure: How to Create a Port for the SOAP Disposition

1. Connect to the Events node in iWay Explorer and expand the node for the application system (for example, *ConnectDirect*).
2. Right-click the *Ports* node, and click *Add Port* from the menu.

   The Add Port dialog box opens.
3. Supply the values for the fields on the dialog box as follows.
   a. In the Name field, type a name for the port.
   b. In the Description field, optionally type a brief description.
   c. From the Protocol drop-down list, select *SOAP*.
   d. In the URL field, type a SOAP destination, using the following format:

```
soap:[wsdl-url];soapaction=[myaction];method=[web service
method];namespace=[namespace];responseTo=[pre-defined port name or
another disposition url];errorTo=[pre-defined port name or another
disposition url]
```

The following table describes the parameters for the SOAP disposition.

Parameter	Description
wsdl-url	The URL to the WSDL file that is required to create the SOAP message, for example, `http://localhost:7001/ibsp/IBSPServlet/test/`*`webservice`*`.ibs?wsdl` where: *`webservice`* Is the name of the web service that you created using Business Service Explorer. To find this value, navigate to the Business Service Explorer node and open the *Service Description* hyperlink in a new window. The WSDL URL appears in the Address field. You can also open the WSDL file in a third-party XML editor (for example, Altova XMLSpy®) and view the SOAP request settings to find this value.

Parameter	Description
soapaction	The method called by the SOAP disposition, for example, *webservice.method@test@@* where: *webservice* Is the name of the web service that you created using Business Service Explorer. *method* Is the method being used. *test* Is the license that is being used by the web service. To find this value, navigate to the Business Service Explorer node and open the *Service Description* hyperlink in a new window. Perform a search for *soapAction*.
method	The web service method that you are using. You can find this value in the WSDL file.
namespace	The XML namespace that you are using. You can find this value in the WSDL file.
responseTo	The location to which responses are posted, which can be a predefined port name or another URL. Optional.
errorTo	The location to which error logs are posted, which can be a predefined port name or another URL. Optional.

4. Click *Finish* when you have supplied the values on the Add Port dialog box.

   The port that you added is displayed beneath the Ports node.

Procedure: How to Create a Port for the HTTP Disposition With an IBSP Deployment

1. Connect to the Events node in iWay Explorer and expand the node for the application system (for example, *ConnectDirect*).
2. Right-click the *Ports* node, and click *Add Port* from the menu.

   The Add Port dialog box opens.

3. Supply the values for the fields on the dialog box as follows.
   a. In the Name field, type a name for the port.
   b. In the Description field, optionally type a brief description.
   c. From the Protocol drop-down list, select *HTTP*.
   d. In the URL field, type an HTTP destination, using the following format

```
http://[myurl];responseTo=[pre-defined port name or another
disposition url]
```

   where:

   `myurl`

   Is the URL target for the post operation, for example,

```
http://myhost:1234/docroot
```

   `responseTo`

   Is the location to which responses are posted, if desired.

4. Click *Finish* when you have supplied the values on the Add Port dialog box.

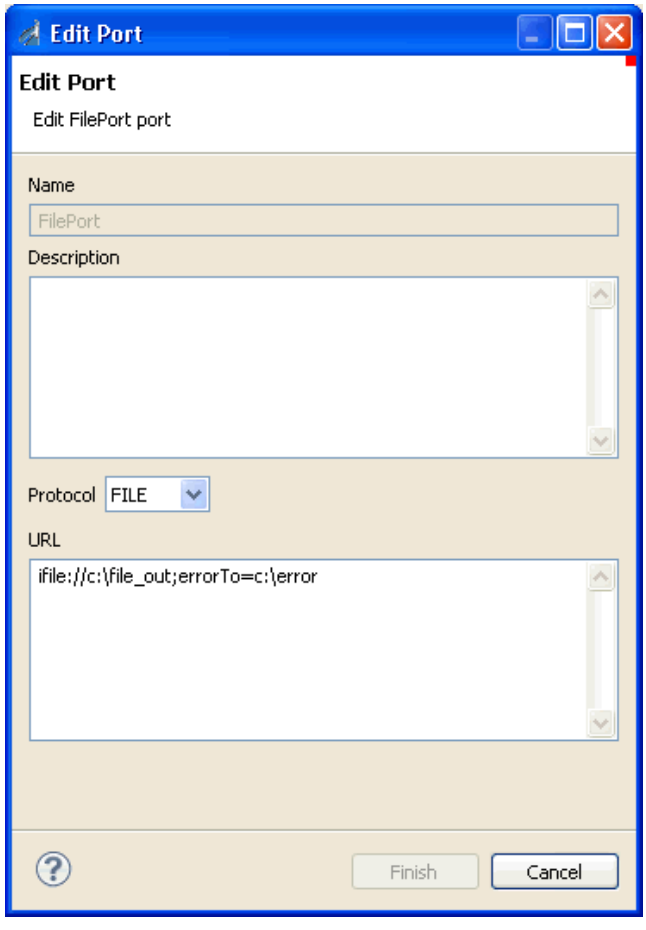
   The port that you added is displayed beneath the Ports node.

Procedure: How to Edit a Port

After you create a port, you can edit the information that you provided during the creation procedure.

1. Expand the *Ports* node in the Events area of iWay Explorer to locate the name of the port that you want to edit, for example, *FilePort*.
2. Right-click the port, and click *Edit* from the menu.

The Edit Port dialog box opens. It displays the values that you supplied when you created the port.



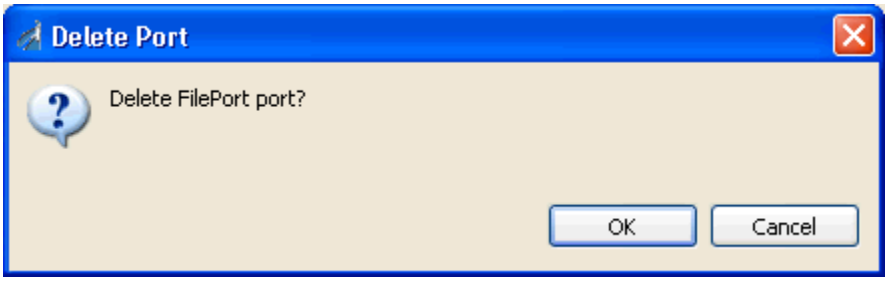
3. Use the fields on the dialog box to modify the properties as desired. You cannot change the name of the port.
4. Click *Finish* when you have completed your edits.

   The modified properties are applied to the port.

Procedure: How to Delete a Port

1. Expand the *Ports* node in the Events area of iWay Explorer to locate the name of the port that you want to delete, for example, *FilePort*.
2. Right-click the port, and click *Delete* from the menu.

iWay Explorer displays a prompt, asking you to confirm the deletion of the selected port, as shown in the following image.



3. Click *OK* to proceed with the deletion.

Using the Default Port

When using iWay Explorer to connect to an application system and listen for events, a default port is available.

You can use the default port for testing purposes or when you do not want to route event data to a specific port that you have configured. The default port is enabled when you start a channel that does not have a specific port assigned to it.

The default event data is a file disposition that writes to an out.xml file in the following output directory:

```
ifile://./eventOut/out.xml
```

Procedure: How to Modify the Default Port Output Directory

1. In the ibse\WEB-INF\lib directory, open the ibse.jar file, and locate the dispositioninfo.xml file.
2. Using a text editor, locate the following lines in the dispositioninfo.xml file:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <dispositioninfo pref="built-in" defaultPortURL="ifile://./eventOut/
   out.xml">
   ```

3. Change the default output directory to a new location of your choice.

   You can also change the name and type of the default output file.

4. Save your changes, and redeploy iBSP.

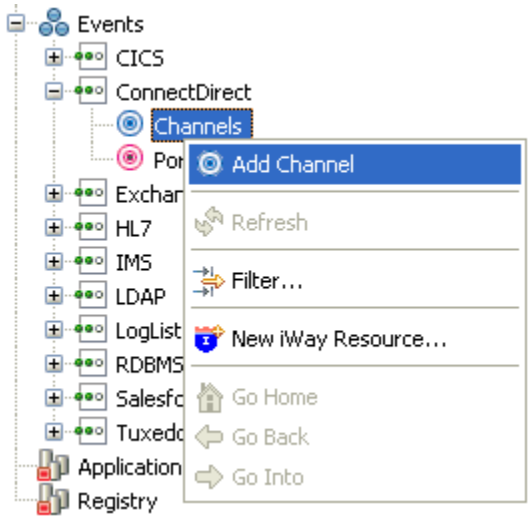
Creating a Channel With the Adapter for Direct:Connect

A channel binds a port to a listener managed by an iWay adapter. A defined port must be associated with a channel in order for you to listen for events that take place in an application system.

The procedures in this topic describe how to create, edit, or delete a channel using the Events node in iWay Explorer.

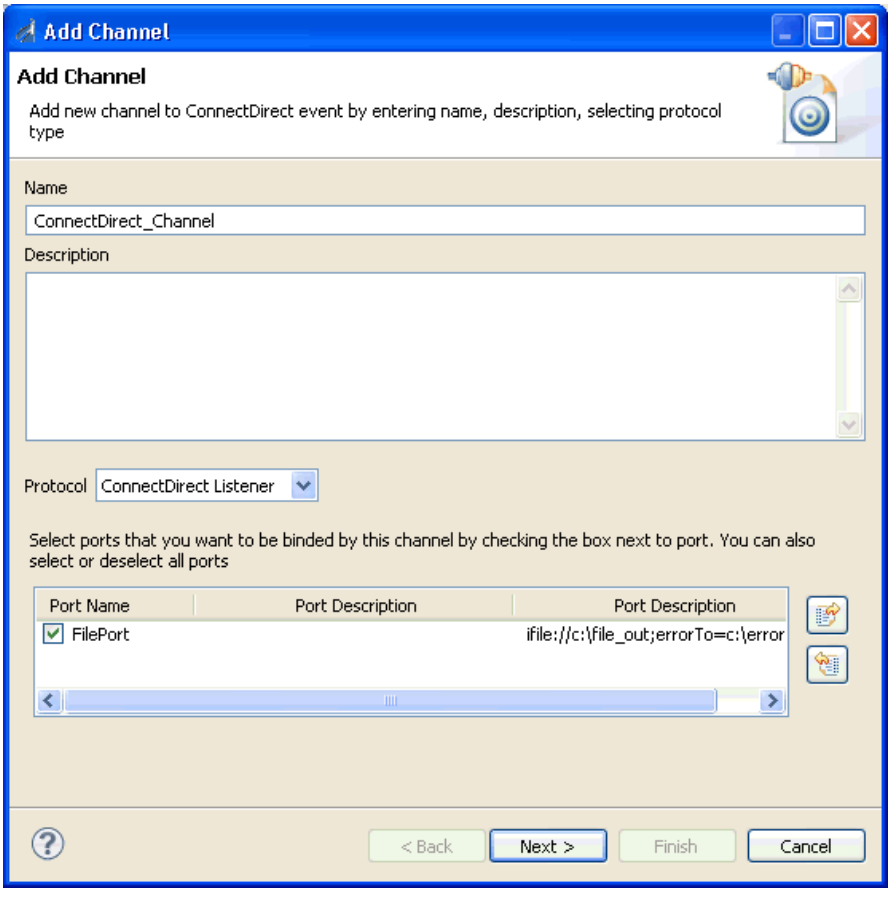
Procedure: How to Create a Channel

1. Connect to the Events node in iWay Explorer and expand the node for the application system (for example, *ConnectDirect*).



2. Right-click the *Channels* node, and click *Add Channel* from the menu.

The following image shows the Add Channel dialog box that opens, where you supply information about the channel.



a. In the Name field, type a name for the channel, for example, ConnectDirect_Channel.

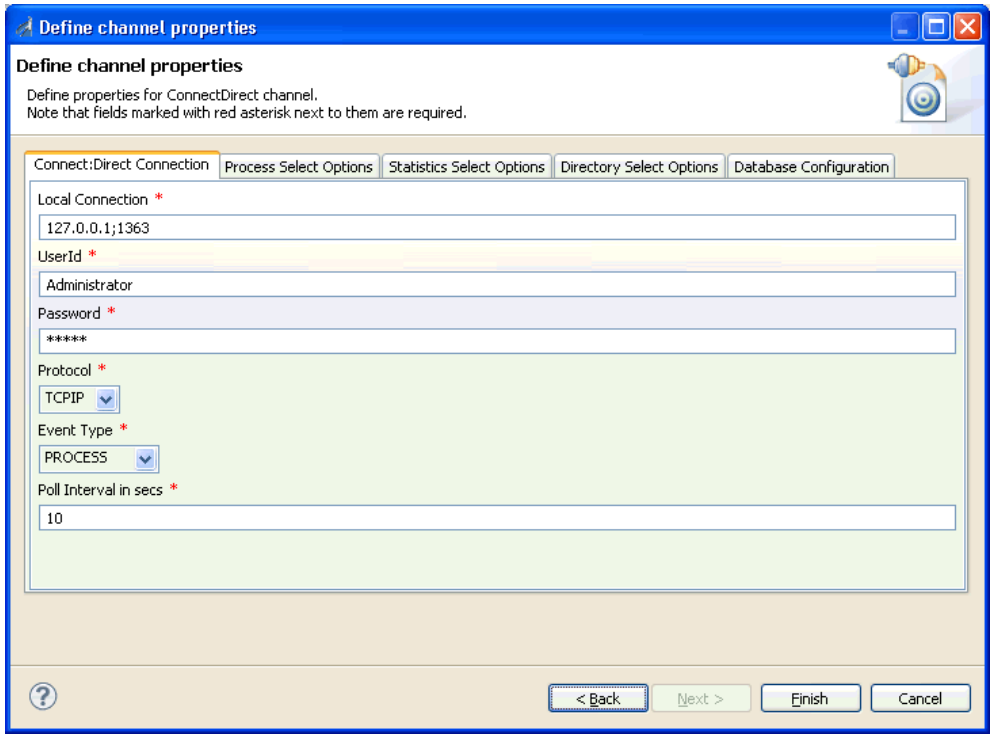
b. In the Description field, optionally type a brief description.

c. From the Protocol drop-down list, select *ConnectDirect Listener*.

d. Under Port Name, select the check box for each port that this channel will bind to a listener.

3. Click *Next* to open the Define Channel Properties dialog box. The Connect:Direct Connection tab is displayed by default, as shown in the following image.

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



The following tabs are available:

- **Connect:Direct Connection tab** (displayed by default)
- **Process Select Options tab**

  For more information, see *Configuring the Process Select Options Tab* on page 79.
- **Statistics Select Options tab**

  For more information, see *Configuring the Statistics Select Options Tab* on page 82.
- **Directory Select Options tab**

  For more information, see *Configuring the Directory Select Options Tab* on page 84.
- **Database Configuration tab**

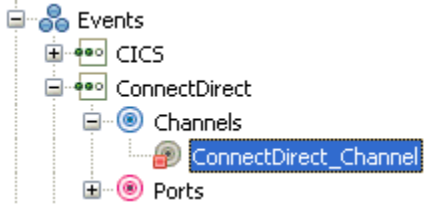
  For more information, see *Configuring the Database Configuration Tab* on page 85.

4. Supply the values for the parameters on the Connect:Direct Connection tab as listed and described in the following table.

Parameter	Description
Local Connection	The name of the machine or IP address (and port) where the Connect:Direct server is being hosted.
UserId	The user name used to connect to the Connect:Direct server.
Password	The password that is associated with the user name.
Protocol	The type of protocol to use for the connection between the adapter and the server. The default protocol is TCP/IP.
Event Type	Select one of the following event types (command) that is expected to be handled by the channel: ❏ PROCESS ❏ STATISTICS ❏ LSDIR
Poll Interval in secs	Enter the desired poll interval (in seconds). This poll interval is the frequency at which the adapter will invoke the selected commands to the Connect:Direct server.

5. Click *Finish* when you are done.

   The following image shows the channel that you added, which appears beneath the Channels node in the Events area of iWay Explorer. In this example, the channel is named ConnectDirect_Channel.



   You are ready to start the channel to listen for events.

6. Right-click the channel, for example, *ConnectDirect_Channel*, and click *Start* from the menu.

The channel is now active and will poll the Connect:Direct system with the configured commands to generate events.

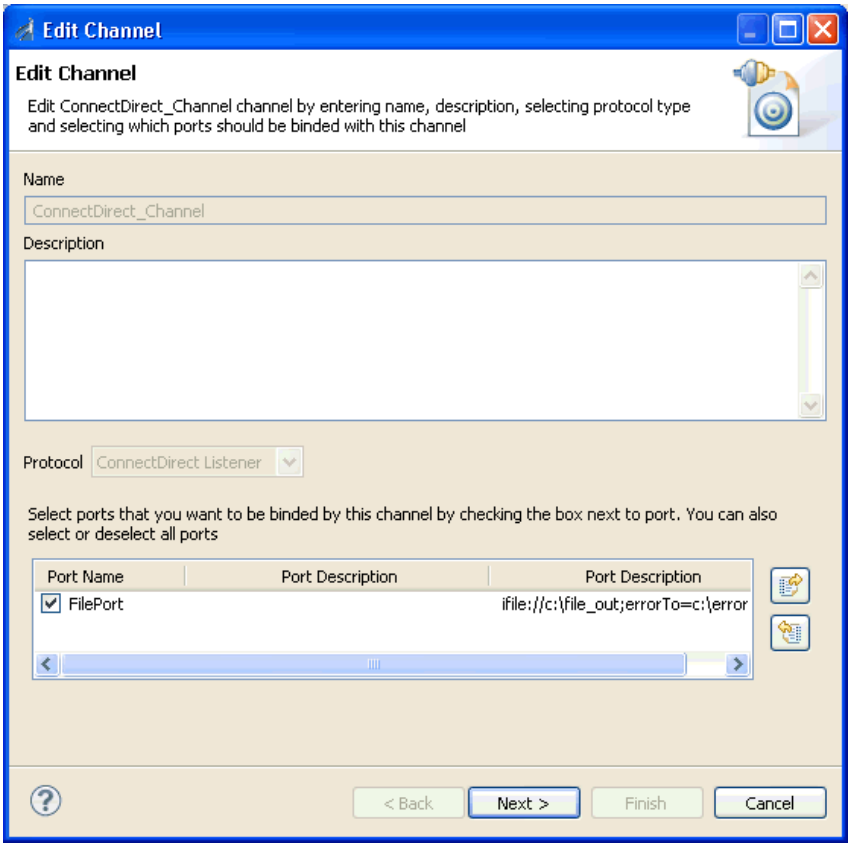
7. To stop the channel at any time, right-click the channel, and click *Stop* from the menu.

Procedure: How to Edit a Channel

After you create a channel, you can edit the information that you provided during the creation procedure.

1. In the Events node of iWay Explorer, locate the name of the channel that you want to edit, for example, *ConnectDirect_Channel*.
2. Right-click the channel, and click *Edit* from the menu.

The Edit Channel dialog box opens. It displays the values that you supplied when you created the channel.



3. Use the fields on the dialog box to modify the properties as desired. You cannot change the name of the channel or its protocol.
4. Click *Next* on the Edit Channel dialog box to open the Define Channel Properties dialog box.
5. Use the tabs and fields on this dialog box to modify the properties as desired.
6. Click *Finish* on the Define Channel Properties dialog box when you have made your edits.

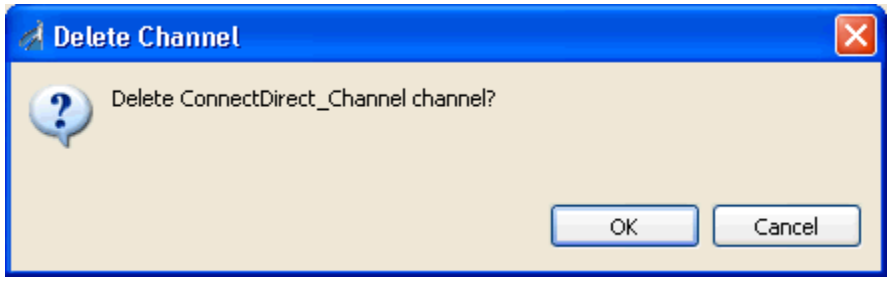
   The modified properties are applied to the channel.

Procedure: How to Delete a Channel

1. Expand the *Channels* node in the Events area of iWay Explorer to locate the name of the channel that you want to delete, for example, *ConnectDirect_Channel*.

2. Right-click the channel, and click *Delete* from the menu.

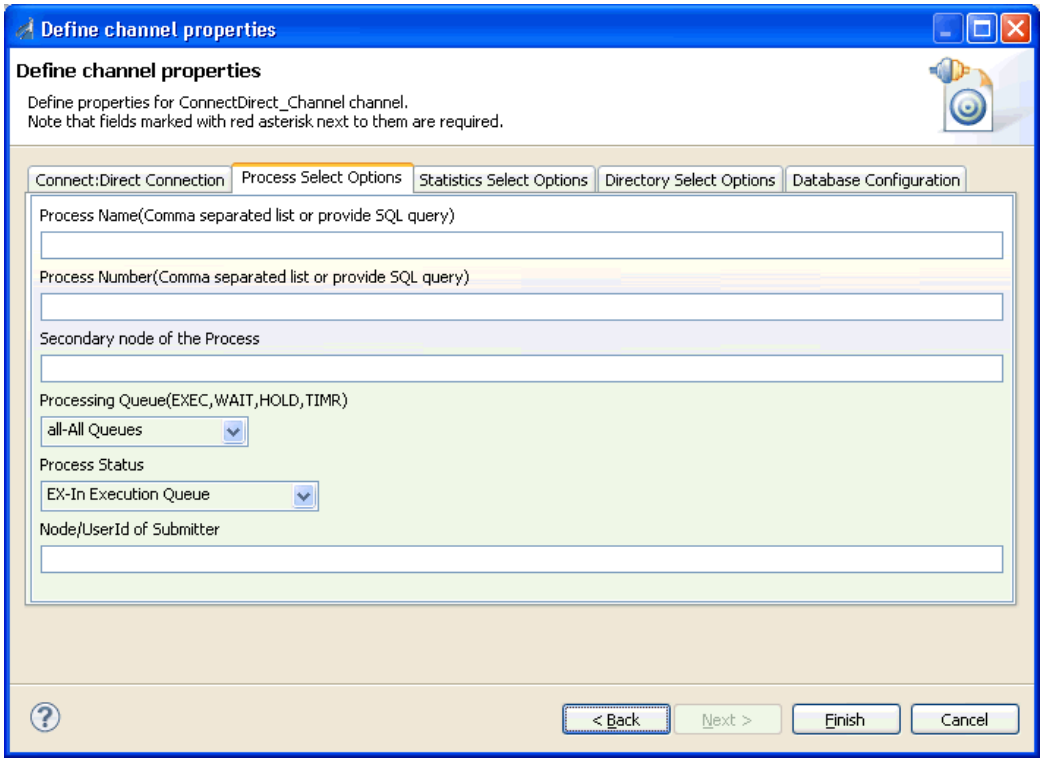
   iWay Explorer displays a prompt, asking you to confirm the deletion of the selected channel, as shown in the following image.



3. Click *OK* to proceed with the deletion.

Configuring the Process Select Options Tab

This section describes how to configure the Process Select Options tab during the channel configuration process, which is shown in the following image.



The parameters available in this tab are similar to the parameters for the Select Process API command. This configuration enables the adapter to poll the Connect:Direct system for the status of submitted processes based on the specified criteria listed and described in the following table:

Parameter	Description
Process Name	The name of the process. A single name, comma separated list of names, or an SQL query can be defined.
Process Number	The number of the process. A single number, comma separated list of numbers, or an SQL query can be defined.
Secondary node of the Process	This is the secondary node of the process.
Processing Queue	The type of processing queue where status is checked. Select one of the following queue types from the drop-down list: ❑ all - All Queues ❑ EXEC - Execution Queue ❑ WAIT - Wait Queue ❑ HOLD - Hold Queue ❑ TIMR - Timer Queue

Parameter	Description
Process Status	The status of the process. Select one of the following status types from the drop down list: ❑ EX - In Execution Queue ❑ PE - Submitted Processes ❑ WC - Awaiting Connection ❑ WR - Retry Status ❑ WA - Eligible for Execution ❑ HC - Submitted with Hold Call ❑ HI - Submitted with Hold Yes ❑ HE - Held due to Connection Error ❑ HO - Held by Change Process ❑ HR - Submitted with Retain Yes ❑ HS - Suspended by Delete Process ❑ RE - Waiting for Restart ❑ WS - Waiting in Timer Queue
Node/UserID of Submitter	The node or user ID of the submitter.

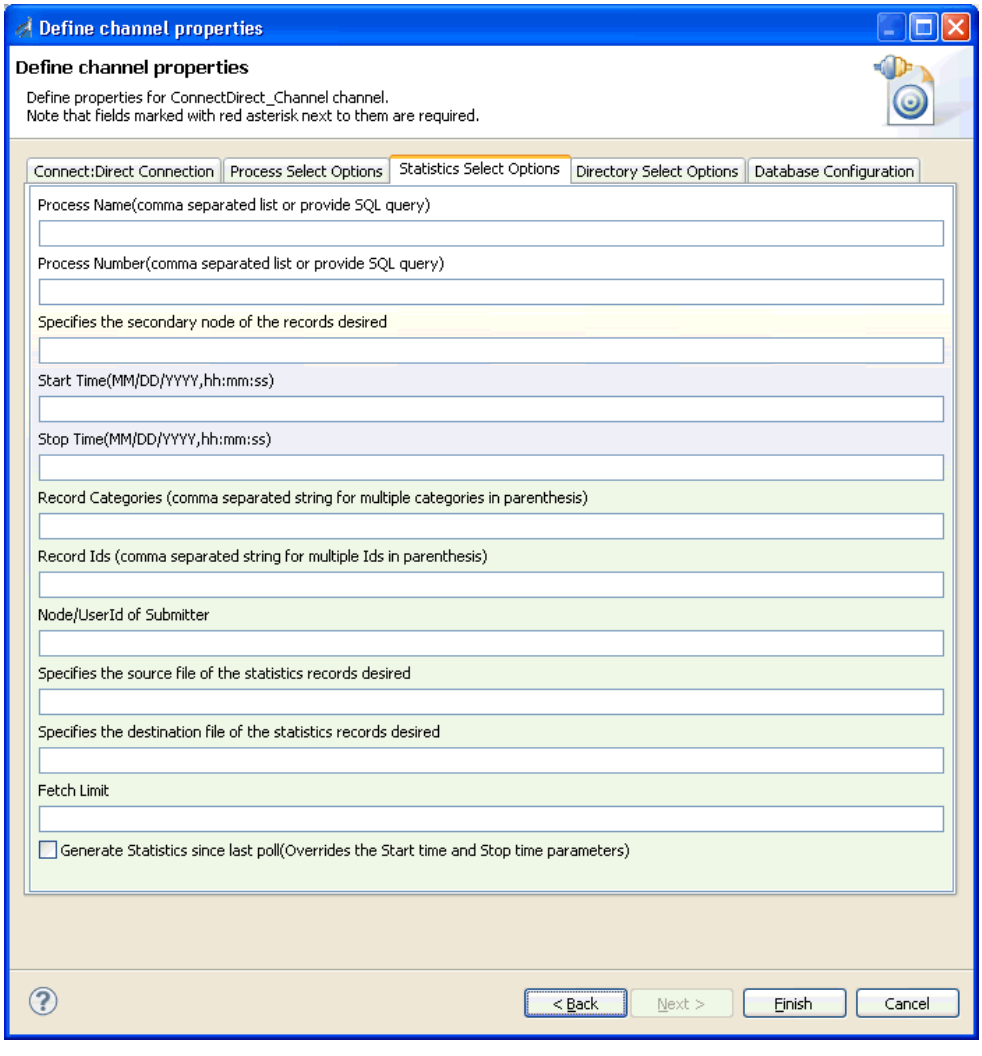
The Process Name and Process Number parameters can be retrieved from a database table by using an SQL query. For example:

```
Select process_name from mytableSelect process_num from mytable
```

These queries are resolved by the adapter during every poll cycle.

Configuring the Statistics Select Options Tab

This section describes how to configure the Statistics Select Options tab during the channel configuration process, which is shown in the following image.



The Statistics Select Options tab enables the adapter to poll the Connect:Direct system for server activity based on the supported criteria listed and described in the following table:

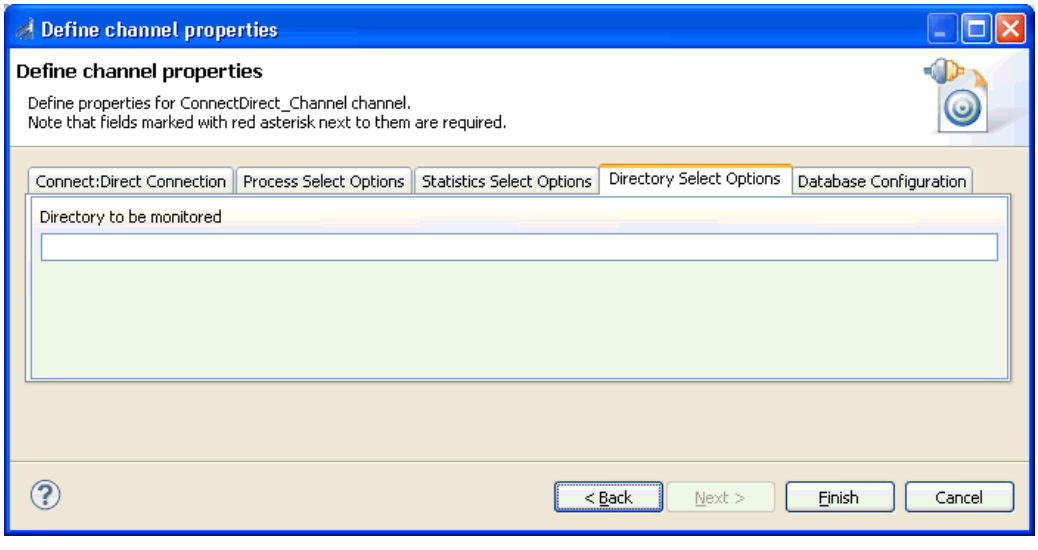
Parameter	Description
Process Name	The name of the process. A single name, comma separated list of names, or an SQL query can be defined.
Process Number	The number of the process. A single number, comma separated list of numbers, or an SQL query can be defined.
Specifies the secondary node of the records desired	This is the secondary node of the records.
Start Time	Start time of the activity specified using the following format: `MM/DD/YYYY,hh:mm:ss`
Stop Time	Stop time of the activity specified using the following format: `MM/DD/YYYY,hh:mm:ss`
Record Categories	The record categories, which can be specified as a comma-separated string for multiple categories in parenthesis.
Record IDs	The record IDs, which can be specified as a comma-separated string for multiple IDs in parenthesis.
Node/UserID of Submitter	The node or user ID of the submitter.
Specifies the source file of the statistics records desired	The source file name for the desired statistics records.
Specifies the destination file of the statistics records desired	The destination file name for the desired statistics records.

Parameter	Description
Fetch Limit	The maximum number of records to be retrieved.

The *Generate Statistics since last poll* check box enables the adapter to only retrieve statistics in the time frame since the last poll cycle. If set, this option overrides the Start Time and Stop Time parameters.

Configuring the Directory Select Options Tab

This section describes how to configure the Directory Select Options tab during the channel configuration process, which is shown in the following image.

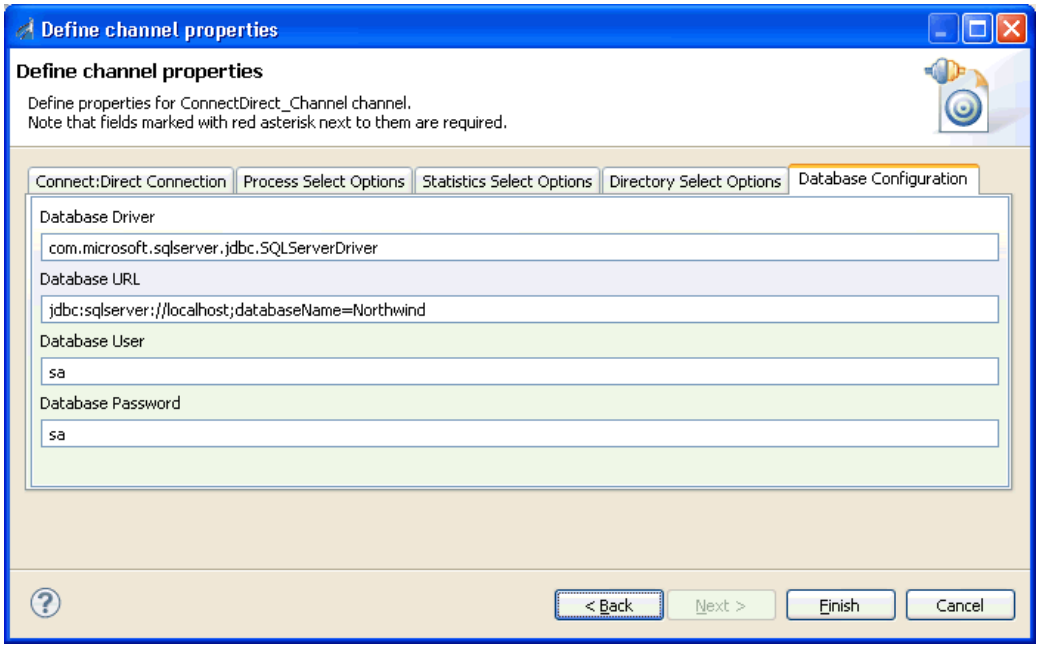


The Directory Select Options tab internally uses the LSDIR command to monitor remote directories for the existence of files.

The Directory to be monitored parameter is used to point to the target directory.

Configuring the Database Configuration Tab

This section describes how to configure the Database Configuration tab during the channel configuration process, which is shown in the following image.



The database configuration is defined during the adapter activation. Ensure that the database drivers must be included in the iway7 classpath.

The following table lists and describes the parameters in the Database Configuration tab.

Parameter	Description
Database Driver	The driver of the database. For example: `com.microsoft.sqlserver.jdbc.SQLServerDriver`
Database URL	The URL of the database. `jdbc:sqlserver://localhost;databaseNameDBName`
Database User	A valid user name to access the database.
Database Password	A valid password to access the database.

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.*

Information Builders

iWay

iWay Protocol Adapter for Connect:Direct User's Guide

Version 7.0.x and Higher

DN3502283.0418

Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898