Using the WebSphere MQ Driver

V4.2
Document ID: TMGN-16-4.2-01
Date Published: 3.5.04

edocs, Inc., One Apple Hill Dr., Natick, MA 01760

Printed in the USA.

Preface

In This Section

Using this Manual

Welcome to Using the WebSphere MQ Driver for Telco Service & Analytics Manager.

This manual covers using the WebSphere MQ Driver in Connectors.

Before You Get Started

You should be familiar with the following:

- Your application architecture
- Programming in Java
- Designing or working with databases
- eXtended Markup Language (XML)
- Your OSS

Who Should Read this Manual

This manual is for developers and project managers who are responsible for configuring and developing connectors which use the IBM WebSphere MQ driver.

However, there are other topics covered in this manual that may interest other members of the project development team.

- Administrators

  You will find information about running and managing connectors using this driver.
- Developers

  You will find everything you need to know in order to build, generate and run connectors using this driver. This manual contains instructions on how to use the Integration Logic Studio you use the driver in connectors. It also shows you how to generate the runtime files then launch the connector.
- Project Architect

  You can use the information in this manual to learn about connectors and how they use this driver.
- Project Manager

  You will find information about connectors and the different tools used to create and run them. You will find information about how your solution uses connectors and this driver.

How this Manual is Organized

This manual contains the following chapters:

- **About WebSphere MQ and the SmartLink (ISF)**

  This chapter introduces the IBM WebSphere MQ driver and its role in the SmartLink (ISF).

  It contains information about:

  - SmartLink (ISF)
  - SmartLink (ISF) Connectors
  - This driver and the SmartLink (ISF) Connectors

- **Installing the WebSphere MQ Driver**

  This chapter covers the installation of the driver.

  It contains information about:

  - Required software
  - Installing this driver

- **Configuring Your Connector**

  This chapter covers the configuration the Connector to use the driver.

  It contains information about:

  - Installing the ILS driver plugin
  - Configuring the connector to use this driver

- **Generating Your Connector**

  This chapter covers generating the runtime files of your connector.

  It contains information about:

  - Using the ILS to generate the runtime files
  - Using the command line tool to generate runtime files

- **Running Your Connector**

  This chapter covers running the connector.

  It contains information about:

  - Starting and stopping connectors

    - Administrating connectors

- **Tool Reference**

  This appendix is a tool reference guide. It covers the location, configuration and use of the administration tools.

  The administration tools covered include:

    - Synchronizer connector tools
    - OSS connector tools
    - Connector Generator

What Typographical Changes and Symbols Mean

This manual uses the following conventions:

TYPEFACE	MEANING	EXAMPLE
Italics	Manuals, topics or other important items	Refer to *Developing Connectors*.
Small Capitals	Software and Component names	Your application uses a database called the CID.
Fixed Width	File names, commands, paths, and on screen commands	Go to `//home/my file`

Obtaining edocs Software and Documentation

You can download edocs software and documentation directly from Customer Central at https://support.edocs.com. After you log in, click on the Downloads button on the left. When the next page appears, you will see a table displaying all of the available downloads. To search for specific items, select the Version and/or Category and click the Search Downloads button. If you download software, an email from edocs Technical Support will automatically be sent to you (the registered owner) with your license key information.

If you received an edocs product installation CD, load it on your system and navigate from its root directory to the folder where the software installer resides for your operating system. You can run the installer from that location, or you can copy it to your file system and run it from there. The product documentation included with your CD is in the Documentation folder located in the root directory. The license key information for the products on the CD is included with the package materials shipped with the CD.

Finding the Information You Need

This driver is an integral part of solutions based on Telco Service & Analytics Manager.

If you are new to using drivers and working with the SmartLink (ISF), you should start by reading *Building Connectors*. This manual helps you understand the integration architecture and shows you how to build connectors to connect to today's market-leading OSS software. You can also read *Using the Loopback Connector* for information about a connector built for development purposes.

You should always read the release bulletin for late-breaking information.

You should refer to the documentation set which came with your version of Telco Service & Analytics Manager for more detailed information.

If You Need Help

Technical support is available to customers who have valid maintenance and support contracts with edocs. Technical support engineers can help you install, configure, and maintain your edocs application.

edocs provides global Technical Support services from the following Support Centers:

US Support Center

Natick, MA
Mon-Fri 8:30am – 8:00pm US EST
Telephone: 508-652-8400

Europe Support Center

London, United Kingdom
Mon-Fri 9:00am – 5:00 GMT
Telephone: +44 20 8956 2673

Asia Pac Rim Support Center

Melbourne, Australia
Mon-Fri 9:00am – 5:00pm AU
Telephone: +61 3 9909 7301

Customer Central

https://support.edocs.com

Email Support

mailto:support@edocs.com

When you report a problem, please be prepared to provide us the following information:

- What is your name and role in your organization?
- What is your company's name?
- What is your phone number and best times to call you?
- What is your e-mail address?
- In which edocs product did a problem occur?
- What is your Operating System version?
- What were you doing when the problem occurred?
- How did the system respond to the error?
- If the system generated a screen message, please send us that screen message.
- If the system wrote information to a log file, please send us that log file.

If the system crashed or hung, please tell us.

Contents

CHAPTER 1

About WebSphere MQ and the SmartLink (ISF)

SmartLink (ISF) drivers are communication drivers for Telco Service Manager (TSM).

This driver is for Telco Service Manager (TSM) that communicate with OSS using IBM's WebSphere MQ middleware.

In This Section

Overview of the SmartLink (ISF) Connectors

The SmartLink (ISF) is not only a set of application integration services, but is a framework in which the providers of the services work together.

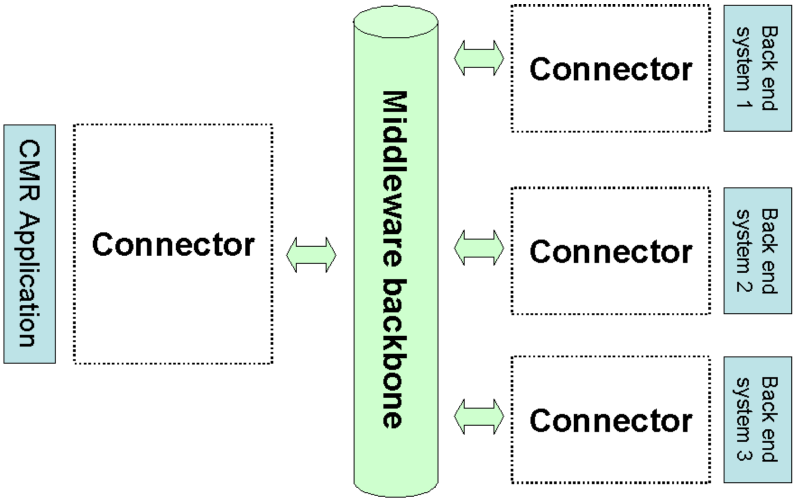
In order to integrate an application, these service providers, or SmartLink (ISF) components, are arranged into dataflows. This means that they are connected together and organized into a step-by-step sequence. A connector is part of the SmartLink (ISF) that is responsible for moving data between applications and transport layers.

There are two types of connectors:

- Synchronizers

  They are responsible for data exchanged between your Telco Service Manager (TSM) and the transport layer
- OSS Connectors

  They are responsible for data exchanged between the BSS/OSS applications and the transport layer



For example, the Synchronizer connector is responsible for moving data between the Telco Service Manager (TSM) and the transport layer. This connector uses different connector components to carry out the following tasks:

- **Send data to the middleware**
  1. Extract the requests from the request queue located in the CID
  2. Use these requests to generate XML messages
  3. Send this XML message to the middleware backbone

- **Receive data from the middleware**

4. Receive the XML message from the middleware
5. Use these messages to create CID objects
6. Update the CID by running scripts which use the CID objects

About the SmartLink

The SmartLink (ISF) enables the Telco Service Manager (TSM) to communicate, or integrate with the core CSP technology infrastructure, for example billing platforms, Customer Relationship Management (CRM) software, and other business and operational support systems (BSS/OSS). This in turn allows the CID to maintain a synchronized cache of semi-static customer and service data, as well as passing requests and responses between the various systems.

The SmartLink (ISF) can be considered to comprise:

- Connectors that move data between Telco Service Manager (TSM) and the transport layer. These connectors are called Synchronizers.
- Connectors that move data between BSS/OSS applications and the transport layer. These connectors are called BSS/OSS Connectors.
- A repository of message definitions to build and process messages for all common transactions. This is the Message Schema Reference Repository.
- Associated tools to design and configure Connectors along with administration tools to run and manage them.

Using the IBM WebSphere MQ Driver

Connectors are responsible for moving data between applications and transport layers. This driver allows your connectors to use WebSphere MQ to exchange messages.

Before you get started, you must create and configure the WebSphere MQ Message Queues. You use the WebSphere MQ tools to create the queue manager and queues for your connector. When you are finished, you are ready to begin using the driver.

Using this driver involves:

- Installing the Connector

  You install the required runtime files so the connector can communicate with WebSphere MQ.

- Configuring the Integration Logic Studio

  You use the WebSphere MQ add-in to import the inbound and outbound WebSphere MQ message queues.

- Configure your Connector to use these queues

  Using the Integration Logic Studio, you modify Integration Processes to use the queues. Depending on the type of connector, your Integration Process may use the inbound queue to receive messages or the outbound queue to send messages.

- Generating and Running the Connector

  You then have to configure the queues for your environment. Once you configure your connectors to use these queues, you generate the connector runtime files. You can then run the connector.

For more information about Connectors and using the Integration Logic Studio, refer to Developing Connectors

CHAPTER 2

Installing the WebSphere MQ Driver

In This Section

Before You Install

Before you install this driver, you must install at least one of the following:

- Telco Service & Analytics Manager 4.2 Framework including the Integration Logic Studio
- Telco Service & Analytics Manager 4.2 Loopback Connector
- Telco Service & Analytics Manager 4.2 Connector Template

Installing the WebSphere MQ Driver

Insert the Telco Service & Analytics Manager WebSphere MQ Driver CD-ROM and open `Contents.html`. This page contains a link to the installer page. On the installer page, click the link to the driver to run the installer.

If you install the WebSphere MQ Driver on a computer running Windows2000, you must use the Windows Install/Remove Programs utility to run the installer you download. You can find this utility in the Control Panel. Go to *Start*>*Settings*>*Control Panel*.

To install the WebSphere MQ driver

1 Run the installer application for your platform that you downloaded from edocs Customer Central or that is on the Telco Service & Analytics Manager CD-ROM. The Installer appears.

- AIX: *wsmq_aix.bin*
- HPUX: *wsmq _hpux.bin*
- Solaris: *wsmq _solaris.bin*
- Windows: *wsmq _windows.exe*

2 Click *Next*. The License Agreement window appears.

3 Read and accept the license agreement then click *Next*. The Location window appears

4 Select the component to install:

- **ISF Driver** to install the driver runtime files.
- **Integration Logic Studio Add-in** to install the add-in.
- **Full Installation** to install all of the components.

5 Follow the onscreen instructions to enter the different paths of installed software. Depending on your installation type, you may have to enter:

- The home directory of Telco Service Manager
- The home directory of the Telco Service Manager Composer
- The location of the WebSphere MQ Client

6 Click *Next*. The Pre-Installation Summary window appears. Review the installation information. If necessary, you can click *Previous* to modify the installation configuration.

7 Click *Install* to install the driver.

When finished, the installer displays a confirmation message.

CHAPTER 3

Configuring Your Connector

In This Section

About Configuring Connectors

After you install the WebSphere MQ driver, you have configure your environment to use the driver.

Configuring your environment involves:

- Installing the WebSphere MQ Addin
- Configuring the connectors to use the queues
- Configuring User Exits
- Setting environment variables

Installing the Integration Logic Studio Add-in

In order for your connectors to use the WebSphere MQ driver, you need to install the addin for the ILS.

This addin installs the following:

- Inbound MQSeries Queue

  This queue is for inbound messages coming from WebSphere MQ
- Outbound MQSeries Queue

  This queue is for outbound messages going to WebSphere MQ

To install add-ins

1. Run the ILS
2. Select *Tools* > *Addins*. The directory selection dialog box appears.
3. Select the location of the addin install files.

   By default, the WebSphere MQ install files are in `<ils_home>/addins/ils/mqseries`
4. Select *OK*. The ILS imports the addin. When finished, the ILS displays a message.
5. Restart the ILS.

Configuring the Connector to use WebSphere MQ

After installing the WebSphere MQ addin, you can open and configure your connector to use the WebSphere MQ queues.

Configuring the connector involves:

- Opening the connector and Integration Processes to modify
- Create the Message Queue processors
- Configure the Message Queue processors
- Connect the Message Queue processors
- Validate the connector

To open connectors

1. Choose *File* > *Open*. The *Open Connector* dialog box opens.
2. Select the connector file to open.
3. Choose *Open*. The connector opens in the Connector Explorer.

To open Integration Processes

1. Open the connector.
2. In the Connector Explorer, expand *Integration Processes.*
3. Double-click the Integration Process. The Integration Process appears in the Workspace.

To create the queues

1. Open the Integration Process.
2. Right-click *Processors* then select *New* > *Processor*. The *Choose a Processor Type* dialog box appears.
3. Expand the *Message Queue* node.
4. Select one of the following:
    - Inbound MQSeries Queue
    - Outbound MQSeries Queue

5 Enter the name of the Processor.

6 Select *Confirm*. The new processor appears in the Connector Explorer and the Integration Process displayed in the Workspace.

To specify the properties of outbound queues

1 Open the *Integration Process*.

2 Expand the *Processors* node.

3 Select the Inbound MQ Series Processor to modify.

4 In the Properties pane, enter the following:

PROPERTY	DESCRIPTION
Queue Manager Name	The queue manager name
Queue Name	The queue name
Host Name	The host name
Port Number	The host port number
Channel Name	The channel name
User Name	User name to connect to WebSphere MQ
User Password	Associated password
Napping time	Time between polling (in milliseconds)
User Exits	Class containing user exit code to execute
Number of threads	Number of threads to allocate
Java priority	JVM priority of the threads
Time To Live	Optional parameter Integer containing the lifespan of the message in the queue (tenths of seconds) This value corresponds to the EXPIRY property of the MQSeries message.

1 Add MQ Options as required.

1. Select Options node under the WebSphere MQ Outbound Message Queue. The properties appear in the *Properties* pane. The *Add* button on the Properties pane is active.
2. Choose *Add* to add a new property. A new row is created in the *Properties* pane.
3. In the new row, enter the name of the MQ Option.

For instance, for outbound queues, you can add the following options:

- `MQOO_OUTPUT`
- `MQOO_SET_IDENTITY_CONTEXT`
- `MQOO_BIND_NOT_FIXED`

For more information about the available MQ Options, refer to your WebSphere MQ documentation.

To specify the properties of inbound queues

1 Open the *Integration Process*.

2 Expand the *Processors* node.

3 Select the Outbound MQSeries Queue Processor to modify.

4 In the Properties pane, enter the following:

PROPERTY	DESCRIPTION
Queue Manager Name	The queue manager name
Queue Name	The queue name
Host Name	The host name
Port Number	The host port number
Channel Name	The channel name
User Name	User name to connect to WebSphere MQ
User Password	Associated password
Time To Live	Optional parameter Integer containing the lifespan of the message in the queue (tenths of seconds) This value corresponds to the EXPIRY property of the MQSeries message.
User Exits	Class containing user exit code to execute

1 Add MQ Options as required.

1. Select Options node under the WebSphere MQ Inbound Message Queue. The properties appear in the *Properties* pane. The *Add* button on the Properties pane is active.
2. Choose *Add* to add a new property. A new row is created in the *Properties* pane.
3. In the new row, enter the name of the MQ Option.

For instance, for inbound queues, you can add the following options:

- `MQOO_OUTPUT`

- `MQOO_SET_IDENTITY_CONTEXT`
- `MQOO_INPUT_AS_Q_DEF`

For more information about the available MQ Options, refer to your WebSphere MQ documentation.

To connect the queues

1. Open the Integration Process in the Workspace.
2. Find the Processor you want to connect.
3. Click the Processor then drag the pointer to the destination Processor. Depending on the type of destination Processor, the following may occur:
   - An arrow appears connecting the two Processors.
   - A menu appears asking you to select the type of connection.
   - A dialog box appears informing you that you cannot connect the Processors.

To save connectors

1. Do one of the following:
   - Choose *File* > *Save* to save the connector.
   - Choose *File* > *Save As* to save the connector with a different name.

To validate connectors

Choose *Connector* > *Check connector*. The ILS validates the connector.

After verification, the ILS displays the result in the message pane. If your connector is invalid or if a problem occured during verification, the ILS displays messages to help you resolve the problem.

Programming User Exits

If you need to customize the processing of messages in the WebSphere MQ inbound and outbound queue processors, you can specify a User Exit. A User Exit is a call to custom methods to process messages in a queue.

If your solution requires specific processing of messages before getting or sending messages using WebSphere MQ, you can use the `MQCustomIF` User Exit class that comes with this driver.

Programming User Exits involves:

- Implementing the custom User Exit class
- Specifying the User Exit class

To implement the default User Exit class

The WebSphere MQ driver comes with class for programming user exits. This class is `MQCustomIF` and is located in the `drv_mqseries.jar` file. This file contains the following methods:

- `beforePut`

  The code in this method is executed before the the Put API.
- `afterPut`

  The code in this method is executed after calling the Put API.
- `beforeGet`

  The code in this method is executed before calling the Get API.
- `afterGet`

  The code in this method is executed after calling the Get API.

To create your User Exit class, implement this class in your class containing the custom code to run before and after calling WebSphere MQ PUT and GET API.

The MQCustomIF Interface

```
/*
 * Determines the requirements for a WebSphere MQ Driver User Exit.
 * <p>
 * This interface has a set of methods to execute before and after calling MQSeries PUT and GET API
 */
public interface MQCustomIF {

       /**
        * Contains the code to run before calling the PUT API.
        * @param eaiMsg the ISF EAI message
        * @param queue the MQ queue
        * @param mmsg the MQ message
        * @param pmo MQ message options
        * @throws Exception if an error occurs
        */

void beforePut(EaiMessage eaiMsg, MQQueue queue,MQMessage mmsg, MQPutMessageOptions pmo) throws Exception;

       /**
        * Contains the code to run after calling the PUT API.
        * @param eaiMsg the ISF EAI message
        * @param mmsg the MQ message
        * @throws Exception if an error occurs
        */

void afterPut(EaiMessage eaiMsg, MQMessage mmsg) throws Exception;

       /**
        * Contains the code to run before calling the GET API.
        * @param mmsg the the MQ message
        * @param queue the MQ queue
        * @param gmo MQ message options
        * @throws Exception if an error occurs
        */

void beforeGet(MQMessage mmsg, MQQueue queue, MQGetMessageOptions gmo) throws Exception;

       /**
        * Contains the code to run after calling the GET API.
        * @param eaiMsg the ISF EAI message
        * @param mmsg the MQ message
        * @throws Exception if an error occurs
        */

void afterGet(EaiMessage eaiMsg,  MQMessage mmsg) throws Exception;

}
```

To specify the User Exit class

1 Open the *Integration Process*.

2 Expand the *Processors* node.

3 Select the MQ Series Queue Processor to modify*.*

4 In the Properties pane, enter the following:

PROPERTY	DESCRIPTION
User Exits	Class containing user exit code to execute

1 Save your changes.

Setting Environment Variables

Depending on your system and environment, you may need to modify or set some dedicated environment variables.

The values for this driver are declared in the `mqseries.env` environment setting file. This file is located in:

- `<home_dir>/bin/env.cmd/isf` for Windows
- `<home_dir>/bin/env.cmd/isf` for UNIX

The default values of these variables comes from the values you entered during installation.

For more information about `.env` environment setting files and how they work, refer to *Configuring Environment Variables* in *Installing Telco Service Manager (TSM).*

CHAPTER 4

Generating Your Connector

In This Section

About Generating Your Connector

The ILS uses the Integration Logic Connector Design file (`<connector_name>.ilcd`) to store the design and configuration of your connector. This file is for configuring and designing connectors. In order to run your connector, you need to generate the runtime files.

The generated connector runtime files are:

- Integration Logic Connector Configuration file (`<connector_name>.ilcc`)

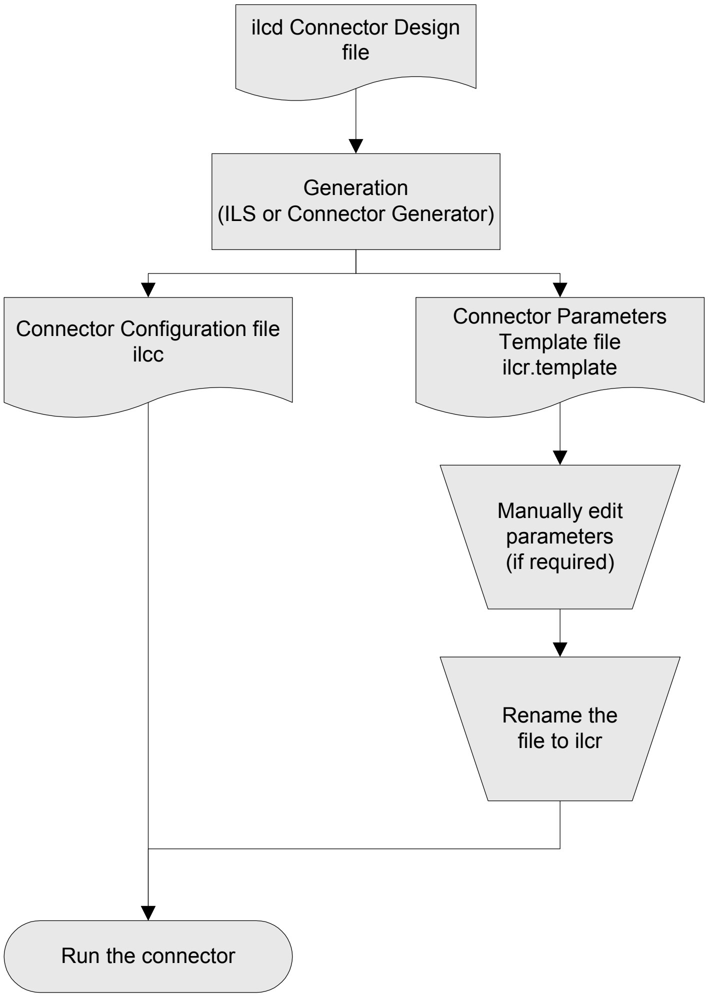
  This file is an internal configuration file used by the connector. It contains the configuration of your connector.

  This file is automatically generated and should not be edited. Use the ILS to change your configuration.

- Integration Logic Connector run time configuration file template (`<connector_name>.ilcr.template`)

  This run time configuration file contains the parameters defined for your connector. You edit this file manually and change its name.

The Generation process is as follows:


ilcd Connector Design file
Generation (ILS or Connector Generator)
Connector Configuration file ilcc
Connector Parameters Template file ilcr.template
Manually edit parameters (if required)
Rename the file to ilcr
Run the connector

To generate the runtime files, you can:

- Use the ILS
- Use the Connector Generator command line tool

To generate connectors

1. Choose *Connector* > *Generate*. The Generate Connector Configuration dialog box appears.
2. If you want the ILS to validate the connector before generation, select *Perform Validation before Generation.*

3 In Output directory, enter the location of the generated connector files.

4 Choose *Generate*. The ILS generates the following connector runtime files:

- `<connector_name>.ilcc`
- `<connector_name>.ilcr.template`

5 When finished, the ILS displays a message in the message pane.

6 If you need to change the runtime parameters before running your connector, open `<connector_name>.ilcr.template` and edit the parameters.

7 Change the file name to `<connector_name>.ilcr`.

To generate connectors using the Connector Generator

1 Go to `<ils_dir>/bin`.

2 Run the `ilccGenerator` tool. Use the syntax:

```
ilccGenerator  <connector_name>
```

where `<connector_name>` is the full path of the Integration Logic Connector Design `.ilcd` file

The tool generates the the following runtime files in the same directory as the Integration Logic Connector Design `.ilcd` file.

- `<connector_name>.ilcc`
- `<connector_name>.ilcr.template`

3 If you need to change the runtime parameters before running your connector, open `<connector_name>.ilcr.template` and edit the parameters.

4 Change the file name to `<connector_name>.ilcr`.

CHAPTER 5

Running Your Connector

In This Section

Starting and Stopping the Synchronizer Connector

You use a set of administration tools to start and stop the Synchronizer connector. The administration tools are:

- `agentstart`
- `agentadm`

These administration tools are located in `<home_dir>/bin`.

About Synchronizer Run Modes

The Synchronizer connector works in one of the following modes:

MODE	DESCRIPTION
NORMAL	THE DEFAULT MODE IN WHICH THE SYNCHRONIZER SCANS THE CID AT A REGULAR, FIXED TIME INTERVALS AND EXTRACTS THE REQUESTS
PAUSED	THE SYNCHRONIZER CONNECTOR STOPS SCANNING THE CID FOR REQUESTS, AND WAITS FOR A RESTART COMMAND BEFORE IT RESUMES
RECOVER	THE SYNCHRONIZER CONNECTOR MAKES A FINAL ATTEMPT TO SEND REQUESTS (WITH THE STATUS 'TRANSPORTFAILED') THAT HAVE NOT BEEN SENT DUE TO A TRANSPORT PROBLEM. WHEN FINISHED, IT GOES BACK TO THE NORMAL MODE.

To start the Synchronizer connector

1. Go to `<home_dir>/bin`.
2. Run `agentstart`. Use the syntax:

```
agentstart <connector_name>
```

The connector is loaded and it starts the processes it needs.

To stop the Synchronizer connector

1. Go to `<home_dir>/bin`.
2. Run `agentadm`. Use the syntax:

```
agentadm <host> <port> shutdown
```

When finished, it displays a message.

Starting and Stopping the OSS Connector

You use a set of administration tools to start and stop the OSS connector. The administration tools are:

- `ossstart`
- `ossadm`

These administration tools are located in `<home_dir>/bin`.

To start the OSS connector

1. Go to `<home_dir>/bin`.
2. Run `ossstart`. Use the syntax:

   ```
   ossstart <connector_name>
   ```

   The connector is loaded and it starts the processes it needs.

To stop the OSS connector

1. Go to <home_dir>/bin.
2. Run `ossadm`. Use the syntax:

   ```
   ossadm <host> <port> <connector> shutdown
   ```

   When finished, it displays a message.

Administrating the Connector

You can stop, start and manage the connector interactively.

You use a set of administration tools to start, stop and manage the Synchronizer and OSS Connector.

About Administrating Connectors

To administer the connectors, you have a complete set of commands you can use to start, stop, or change the settings of the connector while it runs.

Whenever you send a command, the connector responds confirming the execution of your command. For example, you may see:

```
AGENTADM LOCALHOST 3000 STOP

-200AGENT PAUSED

AGENTADM LOCALHOST 3000 START

-200AGENT RESUMING TO NORMAL MODE

AGENTADM LOCALHOST 3000 POSE

-220UNKNOWN COMMAND POSE
```

To manage the Synchronizer connector

1 Log into the computer

2 Do one of the following:

- If the connector is running, run `agentadm`
- If the connector is not running, run `agentstart <synchronizer_name>`, then `agentadm`

3 Use one of the commands below to manage the Synchronizer connector:

COMMAND	DESCRIPTION	PARAMETERS	RETURN VALUES
start	Resumes the execution of the connector. Use this command to restart the connector after a stop command.	none	-200 AGENT START
stop	Stops the connector. The connector no longer processes inbound and outbound messages	none	-120 AGENT STOP
shutdown	Terminates the connector	none	-100 AGENT SHUTDOWN
setparameter	Changes the settings of the remotely. Use the following syntax: setparameter <parameter>=<value>	AGENTNAPPING AGENTLOOPING ONE_EXECUTION NBRROW QFILL MAXNUMBERRETRY RETRYDELAY	-510 AGENT SET<parameter name>=<parameter value>
getparameter	Restarts the current value of the parameter	AGENTNAPPING AGENTLOOPING ONE_EXECUTION NBRROW QFILL MAXNUMBERRETRY RETRYDELAY Any parameter defined in the properties files	-610 AGENT GET<parameter name>=<parameter value>
getmode	Displays the current running mode	none	-210 AGENT NORMAL -220 AGENT REDUCE -230 AGENT PAUSE
getstatus	Displays the current state	none	-130 AGENT NORMAL -140 AGENT REDUCE -150 AGENT PAUSE

force	Forces the connector to run in the specified mode	NORMAL REDUCE RECOVER	-410 AGENT SET MODE=NORMAL -410 AGENT SET MODE=REDUCE -410 AGENT SET MODE=RECOVER" For MODE=RECOVER, an additional parameter must be provided: FILTER <FilterName>
info	Displays information about the connector	none	[AGENTNAME:xxx,NBTHREADS:xxx ,FILL:xxx,NBROW:xxx,NAPPING: xxx,LOOPING:xxx,VERSION: xxx, BUILD:xxx,OS:xxx]"
kill	Forces shutdown	none	none
list	Returns all the profiling information	none	none
purge	Purges the profiling information	none	none
stat	Returns the profiling information of the last profiling element	none	none
version	Returns the version of the connector	none	none

To manage the OSS connector

1. Log into the computer
2. Do one of the following:
   - If the connector is running, run `ossadm`
   - If the connector is not running, run `ossstart <synchronizer_name>`, then `ossadm`
3. Use one of the commands below to manage the OSS connector:

COMMAND	DESCRIPTION	PARAMETERS	RETURN VALUES
START	Resumes the execution of the connector. Use this command to restart the connector after a stop command.	none	-200 AGENT START
stop	Stops the connector. The connector no longer processes inbound and outbound messages	none	-120 AGENT STOP
shutdown	Terminates the connector.	none	-100 AGENT SHUTDOWN
kill	Forces shutdown.	none	none

APPENDIX A

Tool Reference

In This Section

Synchronizer Connector Administration Tools

You use the following administration tools to administrate the Synchronizer Connector:

- `agentstart`
- `agentadm`

These tools are located in `<home_dir>/bin`.

agentstart Syntax

```
agentstart <connector_name>
```

PARAMETERS	DESCRIPTION
<connector_name>	Name of the synchronizer connector to start

agentadm Syntax

```
agentadm help | <host> <port> <command> [<parameters>]
```

PARAMETERS	DESCRIPTION
help	Displays help for the tool
<host>	Specifies the agent host
<port>	Specifies the agent administration port
<command>	Administration command

COMMAND	DESCRIPTION	PARAMETERS	RETURN VALUES
start	Resumes the execution of the connector. Use this command to restart the connector after a stop command.	none	-200 AGENT START
stop	Stops the connector. The connector no longer processes inbound and outbound messages	none	-120 AGENT STOP
shutdown	Terminates the connector	none	-100 AGENT SHUTDOWN
setparameter	Changes the settings of the connector remotely. Use the following syntax: setparameter <parameter>=<value>	AGENTNAPPING AGENTLOOPING ONE_EXECUTION NBRROW QFILL MAXNUMBERRETRY RETRYDELAY	-510 AGENT SET<parameter name>=<parameter value>

getparameter	Restarts the current value of the parameter	AGENTNAPPING AGENTLOOPING ONE_EXECUTION NBRROW QFILL MAXNUMBERRETRY RETRYDELAY Any parameter defined in the properties files	-610 AGENT GET<parameter name>=<parameter value>
getmode	Displays the current running mode	none	-210 AGENT NORMAL -220 AGENT REDUCE -230 AGENT PAUSE
getstatus	Displays the current status	none	130 AGENT NORMAL -140 AGENT REDUCE -150 AGENT PAUSE
force	Forces the connector to run in the specified mode	NORMAL REDUCE RECOVER	-410 AGENT SET MODE=NORMAL -410 AGENT SET MODE=REDUCE -410 AGENT SET MODE=RECOVER For MODE=RECOVER, an additional parameter must be provided: FILTER <FilterName>
info	Displays information about the connector	none	[AGENTNAME:xxx,NBTHREADS:xxx,FILL:xxx,NBROW:xxx,NAPPING:xxx,LOOPING:xxx,VERSION:xxx,BUILD:xxx,OS:xxx]
kill	Forces shutdown	none	none
list	Returns all the profiling information	none	none
purge	Purges the profiling information	none	none

stat	Returns the profiling information of the last profiling element	none	For each message type, displays the following information: - name of message type - min processing time - max processing time - total processing time - mean processing time - deviation processing time - nb number of messages - probability the percentage of successful messages processed - first timestamp of the first message processed - last timestamp of the last message processed - throughput number of messages processed per minute
version	Returns the version of the connector	none	none

OSS Connector Administration Tools

The tools you use to administrate the OSS connector are:

- `ossstart`
- `ossadm`

ossstart Syntax

```
ossstart help | <connector_name>
```

PARAMETERS	DESCRIPTION
`<CONNECTOR_NAME>`	**NAME OF THE OSS CONNECTOR TO START**

ossadm Syntax

```
ossadm help | <host> <port> <command>
```

PARAMETERS	DESCRIPTION
HELP	**DISPLAYS HELP FOR THE TOOL**
`<HOST>`	Specifies the connector host
`<port>`	Specifies the connector port
`<command>`	Administration command

COMMAND	DESCRIPTION	PARAMETERS	RETURN VALUES
`SHUTDOWN`	**TERMINATES THE CONNECTOR**	**NONE**	**NONE**
`KILL`	**FORCES SHUTDOWN**	**NONE**	**NONE**

Connector Generator Tool

This tool generates the following connector runtime files from the Integration Logic Connector Design (`<connector_name>.ilcd`) file:

- Integration Logic Connector Configuration file (`<connector_name>.ilcc`)
- Integration Logic Connector Run time configuration file (`<connector_name>.ilcr`)

ilccGenerator Syntax

```
ilccGenerator -help | [-check] [-verbose] [-dir <output
directory>] <ILCD file>
```

PARAMETERS	DESCRIPTION
help	Displays help for the tool
check	Checks the validity of the ilcd file before generating
verbose	Displays all generation messages
dir <output directory>	Specifies the destination directory if other than the location of the ilcd file
<connector_name>	The ilcd file to use to generate the connector runtime files

Index