
CHIPS Adapter

The CHIPS adapter sends CHIPS messages using MQ or SWIFTNet.

The following table provides an overview of the CHIPS adapter:

System Name	CHIPSAdapter
Graphical Process Modeler (GPM) categories	All Services.
Description	This adapter sends CHIPS messages using MQ or SWIFTNet to The Clearing House.
Business usage	Use this adapter to send CHIPS financial information using MQ or SWIFTNet to the CHIPS network. The business value of this adapter is inherent in using the benefits of the CHIPS network to exchange financial messages.
Usage example	A user needs to send CHIPS payment messages to The Clearing House and does so using the CHIPS adapter.
Preconfigured?	A default version of the CHIPS adapter is preconfigured.
Requires third party files?	MQSeries, com.ibm.mq.jar and mqji_en_US.properties must be installed to use MQ delivery.
Platform availability	All supported Application platforms.
Related services	This adapter works with the CHIPS Utility service, the Websphere MQ Suite Async Receive adapter, the SWIFTNet HTTP Server adapter, and the SWIFTNet Server adapter.
Application requirements	None.
Initiates business processes?	Yes, either a user-created business process or the sample business process provided with Application (CHIPS business process).
Invocation	A user who has permission to perform this activity must execute the business process that invokes this adapter.
Business process context considerations	The configuration parameters are picked up by the adapter in the workflow. In receiving mode, the adapter puts the incoming documents into the workflow.
Returned status values	Success, Failure.
Restrictions	Either MQ or SWIFTNet transport can be used with the CHIPS adapter.
Persistence level	N/A
Testing considerations	To test this adapter, run the CHIPS adapter business process and verify that it completes successfully. Debug information for this service is located at: Operations > System > Logs > CHIPS

How the CHIPS Adapter Works

The CHIPS adapter sends CHIPS messages to the CHIPS Central Computer, using either the SWIFTNet network (optionally using IBM Websphere MQ) or The Clearing House Frame Relay Network (a proprietary network that uses IBM Websphere MQ). The Clearing House provides a TCP/IP interface for communicating with CHIPS. All CHIPS messages include a message header and all message requests require a message acknowledgement. When the participant sends a message request, the participant expects a CHIPS acknowledgement or CHIPS Invalid Message Acknowledgement message. When CHIPS sends a message request, the participant returns a participant acknowledgement message. Acknowledgements are sent to CHIPS based on the transport mode for which the CHIPS Adapter is configured. When the CHIPS line is inactive, the CHIPS Adapter enables you to send supervisory STATUS messages to CHIPS to test the connection.

You must configure the CHIPS adapter prior to sending any messages to CHIPS. The message payload is passed through a business process that you must also create (or configure the predefined business process).

The CHIPS adapter uses five handlers to send and receive messages:

- ◆ The **Send Handler** sends outbound messages (if the communications method selected is functioning) and stores outbound messages in a local database. The Send Handler creates two mailboxes, based on the participant number, to handle the sending of outbound messages. It stores all the messages into the mailbox for auditing purposes, regardless of the status of CHIPS line. The Send Handler checks the line status in the Application database and, if the line is functioning, it sends the message based on the transport mode (either Websphere MQ or SWIFTNet). If the line is inactive, all messages are stored in a local database table until they can be processed (when the line is functioning again, the Resend Handler batches and sends the locally stored messages in the database).
- ◆ The **Resend Handler** checks for messages that have exceeded the specified resend count. If there are no messages that have exceeded the resend count, it will collect all outbound messages that were stored locally (that is, when the messages were first sent, the selected communication method was unavailable and the messages were stored in the local database). Additionally, the Resend Handler resends messages that have not been acknowledged for more than sixty seconds. If the number of messages collected is greater than the batch number, the messages are sent. If the number of messages collected is less than the batch number, the Resend Handler continues to collect messages for the CHIPS adapter instance (for which the line status is down). In case of a payment message, this handler also includes a possible duplicate tag (PSN [271]) to indicate that the message could be a duplicate. If there are messages that have exceeded the resend count, the Resend Handler will send a STATUS supervisory message to CHIPS. All the other resend messages or new incoming messages are not sent. When the acknowledgement is received, the locally stored messages will be sent in the next time interval. If it does not receive an acknowledgement, it sets the line status to inactive.

Each time a message is resent, the Resend Handler generates a new message number for the message from the database. If the resend count of a message is greater than three (that is, the Resend Handler has attempted to send a particular message three times with no success), the Resend Handler sends an event to the Heartbeat Handler to send a supervisory message to CHIPS. If this occurs, other resend messages or new incoming messages are not sent. You can configure the Resend Handler to start at specific intervals (for example, every ten minutes).

- ◆ The **Heartbeat Handler** is a “listener event”; it is initiated when an event is called by the Resend Handler or the business process that checks the status of CHIPS line. The Heartbeat Handler sends a STATUS supervisory message. It starts the sixty second timer and waits for the response. If there is no response, it updates the database with the status that the line is down. Otherwise, it updates the database with the status “Line up.”

- ◆ The **Receive Handler** receives all messages from CHIPS (through the communication method you specify, either MQ or SWIFTNet), sets the timestamp of each received messages in the appropriate database table, and returns an acknowledgement message if the incoming message is not the CHIPS Acknowledgement message. The Receive Handler stores all incoming messages except heartbeat messages in the appropriate mailbox and Application database table, and parses and validates the header of all incoming CHIPS messages. Heartbeat messages are available in FS_INBOUND database table. If a message received is a Response Acknowledgement from CHIPS, the Receive Handler sets the Acknowledged flag in the appropriate database table, notes the timestamp of the acknowledgement, and decrements the MQ counter (if you are using MQ as your transport mode). If a message received is not a Response Acknowledgement from CHIPS, the Receive Handler notifies the Acknowledgement Handler to send a participant acknowledgement response to CHIPS.
- ◆ The **Acknowledgement Handler** sends the appropriate participant acknowledgement response (transaction code 05) to CHIPS (the response is based on the incoming message). If you are using MQ transport mode, the Acknowledgement Handler uses the MQ parameters you specified for the CHIPS adapter to send the acknowledgement response. If you are using SWIFTNet transport mode, the Acknowledgement Handler encodes the acknowledgement response and sets it in the Primary Document, and this is used to return the SWIFTNet Server response.

Please note that the CHIPS adapter automatically performs the following:

- ◆ If the transport mode used is SWIFTNet, the payload *must* be base64 encoded, and the response that is received must be decoded.
- ◆ If the transport mode used is SWIFTNet, the Request Type is set to **chips.payment** if the transaction code is 10; for all other transaction codes, the Request Type is set to **chips.message**.

How the CHIPS Adapter Communicates with SWIFTNet

When the CHIPS adapter is used with the SWIFTNet network, it receives acknowledgement messages from CHIPS in the SWIFTNet Response within sixty seconds, and any incoming messages (for example, heartbeat message, resolver notification) are received by SWIFTNet Server adapter. The return acknowledgement of the incoming messages is performed by the Receive Handler and Acknowledgement Handler within the CHIPS adapter (the business process is bootstrapped using the SWIFTNet Routing Rule).

Note: The SWIFTNet adapter must be preconfigured to start up the SWIFTNet MEFG Server to listen for incoming messages.

The SWIFTNet transport process handles batches of messages as a sequential request and response process.

The SWIFTNet Client service is executed to create the SWIFTNet message header based on the configuration set in the CHIPS adapter. The request type is either chips.payment (if the transaction code is 10) or chips.message (for all transaction codes except 10).

How the CHIPS Adapter Communicates with MQ

When the CHIPS adapter is used with MQ, any acknowledgement from CHIPS and any incoming messages (for example, heartbeat message, resolver notification, and so forth) are received by the Websphere MQ Suite Async Receive adapter.

The return acknowledgement of the incoming messages is performed by the Receive Handler and Acknowledgement Handler within the CHIPS adapter (the business process is bootstrapped from the Websphere MQ Suite Async Receive adapter).

Note: A unique set of the MEFG Server IP, MEFG Server Port, Queue Manager, Channel Name, and Send Queue parameters can only be used in *one* CHIPS adapter configuration. Therefore, if a unique set of these parameters is used, it must not be used in any other CHIPS adapter configuration. Additionally, each line connection can only have one queue active.

The MQ transport process handles batches of messages as follows: open session, open queue, send multiple messages, close queue, and close session.

With MQ, a participant can have more than one connection through clustering, and each connection can be one set of a line configuration. Each line configuration represents a CHIPS adapter configuration, so each participant in this scenario can have up to two CHIPS adapter configurations when using MQ.

Implementing the CHIPS Adapter

To implement the CHIPS adapter, complete the following tasks:

1. Create a configuration of the CHIPS adapter to enable you to send a CHIPS message. For information about the fields specific to this adapter, see *Configuring the CHIPS Adapter* on page 105.

Note: If you create a new configuration, you must also create a new business process or edit a copy of the appropriate predefined business process, to update it to use your adapter configuration. You do not need to create an instance of the CHIPS adapter for every message; you can reuse the CHIPS adapter instance and pass the parameters that differ from the sample adapter through the business process.

2. Specify field settings for the adapter configuration in the Application Admin Console and in the GPM as necessary. See *Configuring the CHIPS Adapter* on page 105.
3. Perform the additional tasks necessary to use the CHIPS adapter. See *Additional Tasks Necessary to Use the CHIPS Adapter* on page 111 for more information.
4. If you are communicating through Websphere MQ Server, configure the Websphere MQ Suite Async Receive adapter. See the *Websphere MQ Suite Async Receive Adapter* documentation for more information.
5. If you are communicating through SWIFTNet, configure the following:
 - ◆ SWIFTNet Server Adapter (see the *SWIFTNet Server Adapter* documentation for more information)
 - ◆ SWIFTNet HTTP Server Adapter (see the *SWIFTNet HTTP Server Adapter* documentation for more information)
 - ◆ SWIFTNet Routing Rule S (see the *Using SWIFTNet* documentation for more information)
6. Create the necessary business process to pass the payload.
 - ◆ For a single document, you must pass the payload as an input document or indicate the document_id as a BPML parameter.
 - ◆ For multiple documents (document_list), you must indicate the document_list and document_id as BPML parameters, or, if you do not enter the document_list in the BPML, the adapter reads (from process data) the following format for processing multiple documents, which is taken from Process Data:

```

<document_list>
  <document_id>abc1</document_id>
  <document_id>abc2</document_id>
  <document_id>abc3</document_id>
</document_list>

```

Note: See *Parameters Passed From Business Process to Adapter* on page 111 for more information on creating the appropriate BPML.

- ◆ Per CHIPS, the CHIPS message input should not exceed 12,200 characters.

Configuring the CHIPS Adapter

1. Select **Deployment > Adapters > Configuration**.
2. Search for CHIPS adapter or select it from the list and click **Go!**
3. Click **Edit**.
4. Specify field settings in the Admin Console or Business Process (*Creating or Setting Up a Adapter Configuration in the Admin Console or Business Process* on page 105), or the GPM (on page 111).
5. On the Confirm page, verify that the **Enable Adapter for Business Processes** check box is selected.

Creating or Setting Up a Adapter Configuration in the Admin Console or Business Process

Use the field definitions in the following table to create a new configuration of the CHIPS adapter, or to set up the configuration provided with Application. Some fields are available in both the Admin Console and in the GPM.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – Do not include the configuration in a group at this time. ◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other adapters to the group as well.) ◆ Select Group – If adapter groups already exist for this adapter type, they are displayed in the list. Select a group from the list. <p>Note: See <i>Managing Adapters and Adapters</i>.</p>
Participant Number	Participant number assigned by CHIPS (either the Sender ID or ABA number). Required. The BPML element is participant_num .
Number of msgs in a batch	The number of messages allowed in a batch during the resend process. The default is 40 and the number should be 40 or less to allow new incoming messages to be processed. Required. The BPML element is msg_num_in_batch .

Field	Description
Resend Count	Maximum number of times that a message may be resent. If the maximum number specified is exceeded, a supervisory message is sent to CHIPS. Default is 3. Required. The BPML element is resend_count .
Workflow to bootstrap during CHIPS/Websphere MQ failure	Select the workflow to invoke if the CHIPS or MQ Server fails, to inform the administrator of the failure. Optional. The BPML element is workflow_failure . Note: This associates a customized business process that enables you to perform administrative work such as notifying an administrator through e-mail when CHIPS or MQ is down.
Environment	Whether the environment is test or production (default). Required. The BPML element is environment .
Transport Mode	The mode of transport to the CHIPS Network. Valid selections are Websphere MQ Server (default) or SWIFTNet. Required. The BPML element is transport_mode .
Websphere MQ Server Name	The host name of the MQ Server. Required. The BPML element is mq_hostname . Note: Only displayed when Transport Mode is set to Websphere MQ Server .
Websphere MQ Server Port No.	The port number of the MQ Server. Required. The BPML element is mq_port . Note: Only displayed when Transport Mode is set to Websphere MQ Server .
Websphere MQ Server User ID	The MQ user identifier used to log in to the MQ server. Optional. The BPML element is mq_userId . Note: Only displayed when Transport Mode is set to Websphere MQ Server .
Websphere MQ Server Password	The MQ password used to log in to the MQ server. Optional. The BPML element is mq_password . Note: Only displayed when Transport Mode is set to Websphere MQ Server .
Channel Name	The channel for the MQ server (server-connection type). Required. The BPML element is mq_channelName . Note: Only displayed when Transport Mode is set to Websphere MQ Server . The Channel Name, Reply-To Queue Manager, and Reply-To Queue parameters are mandatory and must be configured the same as you configure for the WebsphereMQ Async Receiver adapter. When the incoming message is received, it invokes the Receive Handler based on the three values to look up for the matched CHIPS adapter. Based on the CHIPS adapter configuration, it will then be able to send back the acknowledgement message using the same configured MQ information.
Queue Manager	The name of the queue manager. Required. The BPML element is mq_qManager . Note: Only displayed when Transport Mode is set to Websphere MQ Server .
Send Queue	The name of the send queue (Remote Definition queue). Required. The BPML element is mq_sendQueue . Note: Only displayed when Transport Mode is set to Websphere MQ Server .

Field	Description
Reply-To Queue Manager	<p>The name of the reply-to queue manager. Required. The BPML element is mq_replyToqManager.</p> <p>Note: Only displayed when Transport Mode is set to Websphere MQ Server. The Channel Name, Reply-To Queue Manager, and Reply-To Queue parameters are mandatory and must be configured the same as you configure for the WebsphereMQ Async Receiver adapter. When the incoming message is received, it invokes the Receive Handler based on the three values to look up for the matched CHIPS adapter. Based on the CHIPS adapter configuration, it will then be able to send back the acknowledgement message using the same configured MQ information.</p>
Reply-To Queue	<p>The name of the reply-to queue. Required. The BPML element is mq_qManager.</p> <p>Note: Only displayed when Transport Mode is set to Websphere MQ Server. The Channel Name, Reply-To Queue Manager, and Reply-To Queue parameters are mandatory and must be configured the same as you configure for the WebsphereMQ Async Receiver adapter. When the incoming message is received, it invokes the Receive Handler based on the three values to look up for the matched CHIPS adapter. Based on the CHIPS adapter configuration, it will then be able to send back the acknowledgement message using the same configured MQ information.</p>
Requestor DN	<p>Distinguished name of the requestor. Required. The BPML element is swift_requestorDN.</p> <p>Note: This DN must be registered with the SAG instance using SWIFTNet Alliance Webstation. Only displayed when Transport Mode is set to SWIFTNet. The Requestor DN, Responder DN, and Service Name are automatically saved in the SWIFTNet Routing Rule to allow the incoming CHIPS messages to bootstrap the Receive Handler based on the three values.</p>
Responder DN	<p>Distinguished name of the responder. Required. The BPML element is swift_responderDN.</p> <p>Note: This DN must be registered with the SAG instance using SWIFTNet Alliance Webstation. Only displayed when Transport Mode is set to SWIFTNet. The Requestor DN, Responder DN, and Service Name are automatically saved in the SWIFTNet Routing Rule to allow the incoming CHIPS messages to bootstrap the Receive Handler based on the three values.</p>
Service Name	<p>Name of the service to which both SWIFT correspondents have subscribed. Required. This must be a SWIFTNet service to which you are subscribed. The BPML element is swift_serviceName.</p> <p>Note: Only displayed when Transport Mode is set to SWIFTNet. The Requestor DN, Responder DN, and Service Name are automatically saved in the SWIFTNet Routing Rule to allow the incoming CHIPS messages to bootstrap the Receive Handler based on the three values.</p>
SWIFTNet Operation	<p>Whether the communication is handled in Synchronous (default) or Asynchronous mode. Required. The BPML element is swift_op.</p> <p>Note: Only displayed when Transport Mode is set to SWIFTNet.</p>
Request Reference	<p>User reference of the request. Optional. The BPML element is swift_requestReference. Only displayed when Transport Mode is set to SWIFTNet.</p>

Field	Description
MEFG Server IP	The IP address of the SWIFTNet MEFG Server. Required. The BPML element is mefg_server_ip . Note: Only displayed when Transport Mode is set to SWIFTNet .
MEFG Server Port	The port of the SWIFTNet MEFG Server. Required. The BPML element is mefg_server_port . Note: Only displayed when Transport Mode is set to SWIFTNet .
MEFG HTTP Response Timeout	The timeout value for SWIFT to return a response. Optional. Default is 60. The BPML element is mefg_response_timeout . Note: Only displayed when Transport Mode is set to SWIFTNet .
Use SSL	Specify whether to use SSL between the Application and MEFG Server. Valid selections are True and False (default). Required. The BPML element is useSSL . Note: Only displayed when Transport Mode is set to SWIFTNet .
Run As User	Identify the user who has permission to run the scheduled activity. Optional. You can type the user ID. Or you can click the button, select the user ID from the list, and click Save . Required. Note: The scheduler is used to start the resend business process at a specified time interval (preferably every ten minutes), so that any locally stored messages or messages that have not yet been acknowledged can be sent.
Use 24 Hour Clock Display	By default, the scheduling wizard displays times using a 12-hour clock (which designates hours as a.m. or p.m.). Use this option to display times using a 24-hour clock. Optional. Note: The scheduler is used to start the resend business process at a specified time interval (preferably every ten minutes), so that any locally stored messages or messages that have not yet been acknowledged can be sent.
Do not use schedule	If you select this option, you cannot enable the schedule in the future. You must recreate the schedule instead. Use this option only when you do not need a schedule for a service or report. Note: You must select one of the Schedule options. For this scheduling wizard, the type of schedule you select determines what options are displayed on subsequent pages. Note: The scheduler is used to start the resend business process at a specified time interval (preferably every ten minutes), so that any locally stored messages or messages that have not yet been acknowledged can be sent.
Run based on timer	Run the scheduled activity at a certain time or time interval, such as every 2 hours. Note: You must select one of the Schedule options. For this scheduling wizard, the type of schedule you select determines what options are displayed on subsequent pages. Note: The scheduler is used to start the resend business process at a specified time interval (preferably every ten minutes), so that any locally stored messages or messages that have not yet been acknowledged can be sent.

Field	Description
Run daily	<p>Run the scheduled activity one or more times every day.</p> <p>Note: You must select one of the Schedule options. For this scheduling wizard, the type of schedule you select determines what options are displayed on subsequent pages.</p> <p>Note: The scheduler is used to start the resend business process at a specified time interval (preferably every ten minutes), so that any locally stored messages or messages that have not yet been acknowledged can be sent.</p>
Run based on day(s) of the week	<p>Run the scheduled activity on certain days of the week, such as every Monday.</p> <p>Note: You must select one of the Schedule options. For this scheduling wizard, the type of schedule you select determines what options are displayed on subsequent pages.</p> <p>Note: The scheduler is used to start the resend business process at a specified time interval (preferably every ten minutes), so that any locally stored messages or messages that have not yet been acknowledged can be sent.</p>
Run based on day(s) of the month	<p>Run the scheduled activity on certain days of the month, such as the 1st or 15th of every month.</p> <p>Note: You must select one of the Schedule options. For this scheduling wizard, the type of schedule you select determines what options are displayed on subsequent pages.</p> <p>Note: The scheduler is used to start the resend business process at a specified time interval (preferably every ten minutes), so that any locally stored messages or messages that have not yet been acknowledged can be sent.</p>

Business Process Examples

The examples in this section involve the CHIPS adapter sending these four message types:

- ◆ sendCHIPSRequest
- ◆ resendCHIPSRequest
- ◆ runSupervisoryCheck

This example sends a document using the CHIPS adapter:

```
<process name="CHIPSadapter">
  <sequence name="CHIPSadapter">
    <operation name="set user token">
      <participant name="SetUserToken"/>
      <output message="SetUserTokenMessage">
        <assign to="USER_TOKEN">admin</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
    <operation>
      <participant name="CHIPSadapter"/>
      <output message="sendCHIPSRequest">
```

```

        <assign to="." from="*" />
        <assign to="document_id">test-doc-id</assign>
    </output>
    <input message="testing">
        <assign to="." from="*" />
    </input>
</operation>
</sequence>
</process>

```

This example does not need to be created. It is automatically created when the schedule is configured. When the CHIPS adapter is scheduled, the Schedule_CHIPSAdapter business process is automatically created.

```

<process name="Schedule_CHIPS_ADAPTER_MQ">
    <sequence>
        <operation name="Service">
            <participant name="CHIPSAdapter" />
            <output message="resendCHIPSRequest">
                <assign to="." from="*" />
            </output>
            <input message="Xin">
                <assign to="." from="*" />
            </input>
        </operation>
    </sequence>
</process>

```

This example sends a Supervisory STATUS message:

Note: In Application, the business process used to send the supervisory STATUS is CHIPSAdapter_SupervisoryCheck.

```

<process name="CHIPSAdapterSupervisoryCheck">
    <sequence name="CHIPSAdapter">
        <operation name="set user token">
            <participant name="SetUserToken" />
            <output message="SetUserTokenMessage">
                <assign to="USER_TOKEN">admin</assign>
                <assign to="." from="*" />
            </output>
            <input message="inmsg">
                <assign to="." from="*" />
            </input>
        </operation>
        <operation>
            <participant name="CHIPSAdapter" />
            <output message="runSupervisoryCheck">
                <assign to="." from="*" />
            </output>
            <input message="testing">
                <assign to="." from="*" />
            </input>
        </operation>
    </sequence>
</process>

```

```
</sequence>
</process>
```

Parameters Passed From Business Process to Adapter

The following table contains the parameters passed from the business process to the CHIPS adapter:

Parameter	Description
document_list	<p>The list of documents. If you do not enter the document_list in the BPML, the adapter reads (from process data) the following format for processing multiple documents:</p> <pre><document_list> <document_id>abc1</document_id> <document_id>abc2</document_id> <document_id>abc3</document_id> </document_list></pre> <p>Optional.</p> <p>Note: The payload is passed using the BPML, and you can override the CHIPS adapter configuration using these BPML parameters. For example, the payload can be sent in batches. If you want to send a single document, use the document_id parameter. You can also predefine a Process Data tag name (such as DocList) that enables you to put multiple document_id parameters into the tag so the CHIPS adapter picks up all the document_ids at once.</p>
document_id	<p>This is the document identifier if the document is already present in the Document area for single document processing. If document_list or document_id is not specified in the BPML, the document is passed in as the Primary Document. If no document is found, the BPML fails. Optional.</p>

Additional Tasks Necessary to Use the CHIPS Adapter

In addition to configuring the CHIPS adapter, you must also perform the following tasks:

- ◆ Create a mailbox routing rule to invoke the CHIPSExtractMailboxMessage business process.
- ◆ Enable the predefined MailboxEvaluateAllAutomaticRulesSubMin schedule (installed with Application).

Creating the Mailbox Routing Rule

You must create a mailbox routing rule to invoke the CHIPSExtractMailboxMessage business process, which extracts each mailbox message received by the CHIPS adapter and bootstraps the EDIDevelope business process for each extracted mailbox message.

To create the necessary mailbox routing rule:

1. From the **Deployment** menu, select **Mailboxes > Routing Rules**.
2. Next to **Create a new Routing Rule** click **Go!**
3. Specify a Name for the routing rule. This name must be unique for each routing rule. It is used to identify the routing rule in other parts of Application.

4. In the Rule Application page, select **Evaluate Manually** as the **Evaluation Mode**. This specifies that the rule must be evaluated manually or evaluated using a scheduled business process.
 5. For **Action Type**, accept the default **Business Process** selection. This specifies that the rule will notify a business process when a match is found.
 6. Click **Next**.
 7. In the Rule Pattern page, select **Filter by Name**.
 8. From the **Available Mailboxes** list, select the **mailbox that contains your sender ID**, and click the single down arrow to add the mailbox to the Selected Mailboxes list.
- Note:** All groups in the Selected Mailboxes list are searched by the routing rule.
9. For **Message Name Pattern**, type **CHIPSIN_*** and click **Next**.
- Note:** This is the message name or pattern that the routing rule searches for in the mailboxes specified.
10. In the Rule Action page, select the **CHIPSExtractMailboxMessage** business process and click **Next**.
 11. In the Run Rule as User page, select the **admin** user ID and click **Next**.
 12. In the Confirm page, verify the parameters and click **Finish**.
 13. When the system update is complete, click **Return**.

Enable the Predefined Schedule

The mailbox routing rule you created above is executed automatically when the predefined MailboxEvaluateAllAutomaticRulesSubMin schedule is enabled. This means that Application will evaluate all mailbox routing rules on an automatic basis.

To enable the MailboxEvaluateAllAutomaticRulesSubMin schedule:

1. From the **Deployment** menu, select **Schedules**.
2. In the Search section, type **Mailbox** and click **Go!**.
3. Locate the **MailboxEvaluateAllAutomaticRulesSubMin** schedule in this list and select the check box in the Enable column.
4. Click **Return**.

Enabling CHIPS Document Tracking

When you are creating or editing your CHIPS business process in the business process text editor, you can enable CHIPS document tracking in Application by selecting the **Document Tracking** check box on the Process Levels page. Set the following options as needed and leave the rest of the business process parameters as the defaults:

- ◆ On the **Deadline Settings** page, set the deadline and notification options, if necessary.
- ◆ On the **Life Span** page, set the life span, if necessary.