

---

## SWIFTNet Client Service

The SWIFTNet Client service is responsible for sending SWIFT InterAct or FileAct messages (both requests and responses) to SWIFTNet, which are initiated by Application. The SWIFTNet Client service enables you to use InterAct or FileAct messaging with a Store and Forward option. Additionally, the SWIFTNet Client service enables you to use either synchronous or asynchronous messaging for InterAct and either put or get messaging for FileAct.

**Note:** Each instance of the SWIFTNet Client service is configured for a pair of requestor/responder DNs and the SWIFTNet Client service name.

The following table provides an overview of the SWIFTNet Client service:

System Name	SWIFTNetClientService
Graphical Process Modeler (GPM) categories)	All Services.
Description	<p>This service is responsible for sending SWIFT InterAct and FileAct messages (both requests and responses) to SWIFTNet, which are initiated by Application.</p> <p>The SWIFTNet Client service is also executed during system processing 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). You do not need to specifically configure the SWIFTNet Client service for use with CHIPS.</p>
Business usage	Use this service to send financial information based on SWIFT InterAct and FileAct messages to another participant in the SWIFTNet Central network. The business value of this service is inherent in utilizing the benefits of the SWIFTNet Central network to exchange financial messages.
Usage example	You wish to cancel a Customer Credit Transfer (MT192). The user utilizes the service to send out the Cancel Request, and wait for confirmation (MT 196). In this case, the request is sent out synchronously, with the service remaining open until the response is received.
Preconfigured?	Yes.
Requires third party files?	No third party files are required.
Platform availability	All supported Application platforms.
Related services	This service works with the SWIFTNet Server adapter, the SWIFTNet HTTP Server adapter, the Application SWIFTNet MEFG Server, and the Command Line Adapter 2.

Application requirements	The Application SWIFTNet MEFG Server must be installed and configured in order to use this service. SSL can be implemented between Application and the SWIFTNet MEFG Server. You must also configure the SWIFTNet HTTP Server adapter. <b>Note:</b> SSL is not supported on the AIX 5.2 or 5.3 operating systems for the connection between the SWIFTNet MEFG Server and the following service or adapters: SWIFTNet Client service, SWIFTNet HTTP Server adapter, and SWIFTNet Server adapter. Please note that this does not impact outbound or inbound SSL connectivity between the SWIFTAlliance Gateway (SAG) and SWIFTNet, because secure transmissions to the host are supported.
Initiates business processes?	No.
Invocation	A user who has permission to perform this activity must execute the business process that invokes this service.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> <li>◆ Fatal—non-recoverable error</li> <li>◆ Transient—recoverable error</li> <li>◆ Logic—recoverable error</li> <li>◆ Success—Success</li> <li>◆ Warning—Success with warning</li> </ul>
Restrictions	Only one Application SWIFTNet MEFG Server can be configured to talk to one SWIFTNet Server Adapter instance in Application.
Persistence level	N/A
Testing considerations	To test this adapter, run the SWIFTNet Client business process and verify that it completes successfully. Debug information for this service is located at: Operations > System > Logs > SWIFTNet

## How the SWIFTNet Client Service Works

The SWIFTNet Client service prepares the request and sends it to the Application SWIFTNet MEFG Server. The client application on the SWIFTNet MEFG Server processes this request, performs the necessary communication exchange with the SWIFTNet SAG/SNL instance, and sends the request to the SWIFTNet Central network. The SWIFTNet Client service can operate in either synchronous or asynchronous mode for InterAct. In synchronous mode, the request is sent to the SWIFTNet Central network using the SwInt:Exchange primitive. In asynchronous mode, the request is sent to the SWIFTNet Central network using the SwInt:Send primitive.

In synchronous mode, the SWIFTNet MEFG Server client application is blocked until a response is received from the responder through the SAG/SNL instance. Once a response is received, it is sent back to the Application by the client application on the SWIFTNet MEFG Server, and the response payload is placed in the primary document.

In asynchronous mode, the SWIFTNet MEFG Server client application receives a response handle from the SAG/SNL instance. Using this response handle, the SWIFTNet MEFG Server client application

periodically checks with SWIFTNet (using SwInt:Wait primitive) to determine if a response is available. Once a response is received, the response payload is placed in the primary document.

## Implementing the SWIFTNet Client Service

To implement the SWIFTNet Client service, complete the following tasks:

1. Create a configuration of the SWIFTNet Client service. See *Managing Services and Services*. For information about the fields specific to this service, see *Configuring the SWIFTNet Client Service* on page 3.

**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, SWIFTNetClient.bp or SWIFTNetClientFA.bp, to update it to use your service configuration. You do not need to create an instance of the SWIFTNet Client service for every Requestor or Responder DN; you can simply reuse the SWIFTNet Client service instance and pass the parameters that differ from the sample service through the business process.

2. Specify field settings for the service configuration in the Application Admin Console and in the GPM as necessary. See *Configuring the SWIFTNet Client Service* on page 3.

**Note:** When you create the configuration, you will configure it differently depending on whether you are using InterAct or FileAct messaging. Either can be used with or without the store-and-forward option.

## Configuring the SWIFTNet Client Service

1. Select **Deployment > Services > Configuration**.
2. Search for SWIFTNet Client service 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 Service Configuration in the Admin Console or Business Process* on page 3), or the GPM (*Setting Up the Service in the GPM* on page 6).

**Note:** Each instance of the SWIFTNet Client service is configured for a pair of requestor/responder DNs and the SWIFTNet Client service name.

5. On the Confirm page, verify that the **Enable Service for Business Processes** check box is selected.

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

Use the field definitions in the following table to create a new configuration of the SWIFTNet Client service, 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 service configuration. Required.

Field	Description
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> <li>◆ None – Do not include the configuration in a service group at this time.</li> <li>◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.)</li> <li>◆ Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list.</li> </ul> <p><b>Note:</b> See <i>Managing Services and Services</i>.</p>
SWIFTNet Interface	SWIFTNet message type. Valid values are <b>InterAct</b> or <b>FileAct</b> . Required.
Store and Forward	Indicates if the file transfer is done using the store-and-forward method. Valid values are True (use Store-and-Forward) and False (default—do not use Store-and-Forward). Required. BPML element value is <b>SnF</b> .
SWIFTNet Operation	<p>The SWIFTNet operation to send an InterAct or FileAct message. Possible values are:</p> <ul style="list-style-type: none"> <li>◆ Synchronous (default)—InterAct</li> <li>◆ Asynchronous—InterAct</li> <li>◆ Put—FileAct (default)</li> <li>◆ Get—FileAct</li> </ul> <p>Required. BPML element value is <b>sync</b> (default) or <b>async</b> for InterAct, or <b>Put</b> or <b>Get</b> for FileAct.</p>
Requestor DN	<p>Distinguished name of the requestor. Required. BPML element value is <b>requestorDN</b>.</p> <p><b>Note:</b> This DN must be registered with the SAG instance using SWIFTNet Alliance Webstation.</p>
Responder DN	<p>Distinguished name of the responder. Required. BPML element value is <b>responderDN</b>.</p> <p><b>Note:</b> This DN must be registered with the SAG instance using SWIFTNet Alliance Webstation.</p>
Service Name	<p>Name of the service to which both SWIFT correspondents have subscribed. Required. BPML element value is <b>serviceName</b>.</p> <p><b>Note:</b> This must be a SWIFTNet service to which you are subscribed.</p>
Request Type	<p>Request type supported by the message exchange. Optional for InterAct and required for FileAct in SWIFTNet 6.0. BPML element value is <b>requestType</b>.</p> <p><b>Note:</b> In SWIFTNet 6.0 FileAct the format convention is as follows:</p> <pre>&lt;business_area&gt;.&lt;type_of_syntax&gt;.&lt;detailed_syntax_and_format&gt;</pre> <p>This format starts with a four-character business area code, followed by a period (dot), followed by a three-character code that designates the type of syntax (which can be &lt;nnn&gt; , FIN, or xxx), followed by another period (dot), and then followed by a more detailed indication of syntax and format</p>
Request Reference	User reference of the request. Optional. BPML element value is <b>requestReference</b> .

Field	Description
Switch to SnF mode when real-time transmission failed	Indicates whether you want to switch to store-and-forward mode if a real-time transmission (InterAct or a FileAct Put) has failed. Possible values are True or False (default). Required. BPML element value is <b>switchToSnF</b> .
Store and Forward Service Name	The name of the store-and-forward service. Required when Switch to SnF mode when real-time transmission failed is set to True. BPML element value is <b>SnFServiceName</b> .
Non Repudiation required	Indicates whether non-repudiation is required. Possible values are True (when enabled, trading partners cannot deny that they sent a request) and False (default, indicating that non-repudiation is not required). Required. BPML element value is <b>nonRepudiation</b> .
Number of Retries	Number of retries to connect to SAG. Default value is 3. Optional. BPML element value is <b>numOfRetries</b> .
Retry Delay (in seconds)	Number of delays before the next retry. Default value is 60 (seconds). Optional. BPML element value is <b>secInRetryDelay</b> .
Trace	Trace for logging purposes in the SWIFTNet MEFG Server. Valid values are True and False (default). Required. BPML element value is <b>trace</b> .
Delivery Notification (del. Notifn)	Indicates that the sender asked the receiver to send a delivery notification. Possible values are True or False (default). Optional. BPML element value is <b>deliveryNotification</b> . <b>Note:</b> This parameter is only displayed when you select <b>True</b> for Store and Forward or are performing a FileAct Put. If you are performing a Put operation, you can request the responder to send you a delivery notification and specify a different Delivery Notification DN and Request Type of Delivery Notification, if desired. If you are performing a Get operation, the responder can request Delivery Notification from the requestor after receiving the file. That setting for delivery notification is configured through the SWIFTNet Server adapter.
Request Type of Delivery Notification	Used to request a specific delivery notification message from the remote receiving server application when it returns the delivery notification (when Delivery Notification is set to True). Optional. BPML element value is <b>requestTypeDelNotifn</b> . <b>Note:</b> This parameter is only displayed when you select <b>True</b> for Store and Forward or a FileAct Put.
Message Priority	Indicates priority handling in the queue for store-and-forward only. Valid values are Normal (default) and Urgent. Optional. BPML element value is <b>messagePriority</b> . <b>Note:</b> This value is used as a selection criterion when delivering messages from a queue, and in SWIFTNet FileAct to influence the pace of the FileAct flow.
MEFG SWIFTNet IP	The IP address for the SWIFTNet MEFG Server. Required.
MEFG SWIFTNet Port	The port for the SWIFTNet MEFG Server. Default is 80. Optional.
Response Timeout	The timeout interval (in seconds) in which a response must be received or the message operation fails. Optional. Default is 60 seconds.
Non Repudiation	Indicates whether non-repudiation is required. Possible values are True (when enabled this means that trading partners cannot deny that they sent a request) or False (default—when enabled this indicates that non-repudiation is not required). Optional. BPML element value is <b>nonRepudiation</b> .

Field	Description
Use SSL	Whether to enable Secure Socket Layer (SSL) over HTTP communication between Application and the SWIFTNet MEFG Server. Valid values are None (default) and Must. <b>Note:</b> Regardless of the value you select for <b>Use SSL</b> , you must also update the business processes associated with the SWIFTNet Client service. See <i>Upgrading the SWIFTNetClient Business Process to Use the Integrated SWIFTNet Client Service</i> on page 14 for more information.
Cipher Strength	Indicates the strength of the cipher. Possible values are ALL (default), WEAK, and STRONG. Optional.
CA Certificate	The CA certificate of the SWIFTNet MEFG Server. <b>Note:</b> This is the public key certificate that must be configured to set up the outbound SSL channel. This page is only displayed if you set <b>Use SSL</b> to <b>Must</b> . <b>Note:</b> The SWIFTNet Client service Configuration page allows you to select the same CA Certificate for SSL processing a second time, and continues to allow additional selections of the same certificate in subsequent edits. If you have already selected a CA Certificate once for a configuration of the SWIFTNet Client service, do not select the same CA Certificate again, as this will result in an error when you execute the relevant business process.

## Setting Up the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Parameter	Description
deliveryNotification	Indicates that the sender asked the receiver to send a delivery notification. Possible values are TRUE or FALSE. Optional. <b>Note:</b> This parameter is only displayed when you select <b>True</b> for SnF or are performing a FileAct Put. If you are performing a Put operation, you can request the responder to send you a delivery notification and specify a different Delivery Notification DN and Request Type of Delivery Notification, if desired. If you are performing a Get operation, the responder can request Delivery Notification from the requestor after receiving the file. That setting for delivery notification is configured through the SWIFTNet Server adapter.
deliverynotification DN	Distinguished name of the Responder of the delivery notification. Optional.
deliverynotificationRT	Request type of the delivery notification. This is used for a FileAct Get. Required.
fileDesc	User description about the file transfer. Only for FileAct Put. Optional.
fileInfo	Specify whether the file will be compressed or not. Only for FileAct Put. Optional. In SWIFTNet 6.0 FileAct, the format convention is as follows: SwCompression=<value> Valid values are SwCompression=None (default) or SwCompression=ZIP. <b>Note:</b> If you specify to use compression, you must have compressed the file before sending it to the SWIFTNet Server adapter.
interfaceMode	SWIFTNet message type. Valid values are InterAct or FileAct. The default value is InterAct. Required.

Parameter	Description
logicalFilename	<p>This name is communicated to the server application. By default, this name is the physical name without the file path. Optional. Only for FileAct.</p> <p>For a FileAct Put, this is the logical name of the file to be retrieved based on the &lt;reception_dir&gt;/&lt;responder_dn&gt;/&lt;requestor_dn&gt;.</p> <p>For a FileAct Get, this is the logical name of the file to send based on the &lt;download_dir&gt;/&lt;responder_dn&gt;/&lt;requestor_dn&gt;.</p>
messagePriority	<p>Indicates priority handling in the queue for store-and-forward only. Optional.</p> <p><b>Note:</b> This value is used as a selection criterion when delivering messages from a queue, and in SWIFTNet FileAct to influence the pace of the FileAct flow.</p>
nonRepudiation	<p>Indicates whether non-repudiation is required. Possible values are TRUE (when enabled, trading partners cannot deny that they sent a request) or FALSE (default, indicating that non-repudiation is not required). Optional.</p>
numOfRetries	<p>Number of retries to connect to SAG. Default value is 3. Optional.</p>
physicalFilename	<p>Optional. Only for FileAct.</p> <p>For a FileAct Put, this is the full path and the physical name of the file to send.</p> <p>For a FileAct Get, this is the full path and the physical name of the file to save after the Get is completed.</p>
possibleDuplicate	<p>Indicates whether to include a trailer specifying that this message may be a duplicate. This is an optional component of the envelope that indicates that this message may already have been sent. For example, if the system crashes during the delivery of a message, another copy of the message could be sent, with this trailer included to indicate that it may be a duplicate.</p> <p>Possible values are TRUE and FALSE (default).</p> <p>Optional.</p>
requestorDN	<p>Distinguished name of the requestor. Required.</p> <p><b>Note:</b> This DN must be registered with the SAG instance using SWIFTNet Alliance Webstation.</p>
requestReference	<p>User reference of the request. Optional.</p>
requestType	<p>Request type supported by the message exchange. Optional for InterAct and required for FileAct in SWIFTNet 6.0.</p> <p><b>Note:</b> In SWIFTNet 6.0 FileAct the format convention is as follows:</p> <p>&lt;business_area&gt;.&lt;type_of_syntax&gt;.&lt;detailed_syntax_and_format&gt;</p> <p>This format starts with a four-character business area code, followed by a period (dot), followed by a three-character code that designates the type of syntax (which can be &lt;nnn&gt; , FIN, or xxx), followed by another period (dot), and then followed by a more detailed indication of syntax and format</p>
requestTypeDelNotifn	<p>Used to request a specific delivery notification message from the remote receiving server application when it returns the delivery notification (when Non Repudiation required and/or Delivery Notification are set to TRUE). Optional</p>
responderDN	<p>Distinguished name of the responder. Required.</p> <p><b>Note:</b> This DN must be registered with the SAG instance using SWIFTNet Alliance Webstation.</p>

Parameter	Description
secInRetryDelay	Number of delays before the next retry. Default value is 60 (seconds). Optional.
serviceName	Name of the service to which both SWIFT correspondents have subscribed. Required. This must be a SWIFTNet service to which you have already subscribed.
SnF	Indicates if the file transfer is done using the store-and-forward method. Valid values are True (use Store-and-Forward) and False (default—do not use Store-and-Forward). Required.
swiftOp	The SWIFTNet operation to send an InterAct or FileAct message Possible values are: <ul style="list-style-type: none"> <li>◆ sync (default)—InterAct</li> <li>◆ async—InterAct</li> <li>◆ put—FileAct</li> <li>◆ get—FileAct</li> </ul> Required.
trace	Trace for logging purposes in the SWIFTNet MEFG Server. Possible values are 0 (no logging; this is the default) or 4 (logging is enabled). Optional.
transferDesc	User description about the transfer. Only for FileAct. Optional.
transferInfo	User information about the transfer. Only for FileAct. Optional.
switchToSnF	Indicates whether you want to switch to store-and-forward mode if a real-time transmission (InterAct or a FileAct Put) has failed. Possible values are True or False (default). Required.
SnFServiceName	The name of the store-and-forward service. Required when Switch to SnF mode when real-time transmission failed is set to True.
HTTPClientAdapter	HTTP Client Adapter instance that is used to communicate with the SWIFTNet MEFG Server. Optional. Default value is SWIFTNetHTTPClientAdapter.
MEFGServerHost	The IP address of the SWIFTNet MEFG Server. Required.
MEFGServerPort	The port of the SWIFTNet MEFG Server. Optional. Default value is 80.
MEFGServerResponse Timeout	Timeout period (in seconds) for the SWIFTNet MEFG Server to respond. Optional. Default value is 60.
UseSSL	Flag to indicate whether to secure communication between Application and the SWIFTNet MEFG Server with SSL. Possible values are TRUE or FALSE (default). Optional.
CipherStrength	The level of encryption to be applied on the data channel. Possible values are All (default), Weak, or Strong. Optional.
CACertId	The public key certificates for the SWIFTNet MEFG Server. Required if SSL is set to TRUE.

## Business Process Example

To construct a message you need to perform the following tasks:

- ◆ Create a configuration of the SWIFTNet Client service.
- ◆ Edit the SWIFTNetClient business process (or create a new business process) in the following manner:
  - ◆ Match the name of the business process that you create or modify.
  - ◆ If necessary, modify the SWIFTNet MEFG Server IP and port to point to your installation of the SWIFTNet MEFG Server.
  - ◆ Configure the business process for the Requestor DN/Responder DN pair and the SWIFTNet service name.
  - ◆ Specify the request type and request reference for use in SWIFTNet.
  - ◆ If required, select non-repudiation and possible duplicate (which enables the resending of the file in case of an error in transmission) parameters.
  - ◆ Specify the number of retries to the SAG connection and the retry interval.
  - ◆ Enable Document Tracking for AFT Tracking.

**Note:** You do not need to create an instance of the SWIFTNet Client service for every requestor or responder DN; you can reuse the SWIFTNet Client service instance and pass in the requestorDN, responderDN, and any other parameters that differ from the configuration of the sample service through the SWIFTNetClient business process.

This is the BPML for the example business process:

```
<operation>
<participant name="SWIFTNetClientService"/>
<output message="handleClientRequest">
<assign to="." from="*" />
<assign to="swiftOp">async</assign>
</output>
<input message="testing">
    <assign to="." from="*" />
</input>
</operation>
```

This is the complete BPML to execute the SWIFTNet Client service:

**Note:** The **bold** lines indicate information that you need to modify to match the business process you are using.

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

    <operation>
```

```

    <participant name="SWIFTNetClientService" />
    <output message="handleClientRequest">
      <assign to="." from="*"></assign>
      <assign to="interfaceMode">interact</assign>
      <assign to="swiftOp">sync</assign>
      <assign to="requestorDN">o=swiftbic,o=swift</assign>
      <assign to="responderDN">o=swiftbic,o=swift</assign>
      <assign to="serviceName">swift.generic.ia!x</assign>
      <assign to="SnF">FALSE</assign>
      <assign to="nonRepudiation">FALSE</assign>
      <assign to="possibleDuplicate">FALSE</assign>
      <assign to="deliveryNotification">FALSE</assign>
    </output>
    <input message="testing">
      <assign to="." from="*"></assign>
    </input>
  </operation>

</sequence>
</process>

```

This is the complete BPML to execute the SWIFTNet Client service for FileAct for a Put:

```

<process name="SWIFTNet-FA-Put-NonSnF-DN">
  <sequence name="SWIFTNetClientService">
    <operation name="set user token">
      <participant name="SetUserToken" />
      <output message="SetUserTokenMessage">
        <assign to="USER_TOKEN">admin</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation>
      <participant name="SWIFTNetClientService" />
      <output message="handleClientRequest">
        <assign to="." from="*"></assign>
        <assign
to="physicalFilename">/local/share/measle/swiftdata/payload.txt</assign>
        <assign to="logicalFilename">payload.txt</assign>
        <assign to="transferInfo">payload</assign>
        <assign to="transferDesc">payload</assign>
        <assign to="fileDesc">payload</assign>
        <assign to="interfaceMode">fileact</assign>
        <assign to="swiftOp">put</assign>
        <assign to="requestorDN">o=swiftbic,o=swift</assign>
        <assign to="responderDN">o=swiftbic,o=swift</assign>
        <assign to="serviceName">swift.generic.fa!x</assign>
        <assign to="requestType">Type.GIS.Server1</assign>
        <assign to="SnF">FALSE</assign>
        <assign to="nonRepudiation">FALSE</assign>
        <assign to="possibleDuplicate">FALSE</assign>
        <assign to="deliveryNotification">TRUE</assign>

```

```

    </output>
    <input message="testing">
      <assign to="." from="*"></assign>
    </input>
  </operation>

</sequence>
</process>

```

This is the complete BPML to execute the SWIFTNet Client service for FileAct for a Get:

```

<process name="SWIFTNet-FA-Get-NonSnF-DN">
  <sequence name="SWIFTNetClientService">
    <operation name="set user token">
      <participant name="SetUserToken"/>
      <output message="SetUserTokenMessage">
        <assign to="USER_TOKEN">admin</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation>
      <participant name="SWIFTNetClientService"/>
      <output message="handleClientRequest">
        <assign to="." from="*"></assign>
        <assign
to="physicalFilename">/local/share/measle/swiftdata/payload-receive.txt</assign>
        <assign to="logicalFilename">payload.txt</assign>
        <assign to="interfaceMode">fileact</assign>
        <assign to="swiftOp">get</assign>
        <assign to="requestorDN">o=swiftbic,o=swift</assign>
        <assign to="responderDN">o=swiftbic,o=swift</assign>
        <assign to="serviceName">swift.generic.fa!x</assign>
        <assign to="SnF">FALSE</assign>
        <assign to="nonRepudiation">FALSE</assign>
        <assign to="possibleDuplicate">FALSE</assign>
        <assign to="deliveryNotification">TRUE</assign>
      </output>
      <input message="testing">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>

```

## Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the SWIFTNet Client service:

Parameter	Description
messageType	SWIFTNet message type. Valid values are InterAct or FileAct. The default value is InterAct. Required.
swiftOp	The SWIFTNet operation to send an InterAct or FileAct message Possible values are: <ul style="list-style-type: none"><li>◆ sync (default)—InterAct</li><li>◆ async—InterAct</li><li>◆ put—FileAct</li><li>◆ get—FileAct</li></ul> Required.
trace	Trace for logging purposes in the SWIFTNet MEFG Server. Possible values are 0 (no logging; this is the default) or 4 (logging is enabled). Optional.
numOfRetries	Number of retries to connect to SAG. Default value is 3. Optional.
secInRetryDelay	Number of delays before the next retry. Default value is 60 (seconds). Optional.
requestorDN	Distinguished name of the requestor. Required. <b>Note:</b> This DN must be registered with the SAG instance using SWIFTNet Alliance Webstation.
responderDN	Distinguished name of the responder. Required. <b>Note:</b> This DN must be registered with the SAG instance using SWIFTNet Alliance Webstation.
serviceName	Name of the service to which both SWIFT correspondents have subscribed. Required. <b>Note:</b> This must be SWIFTNet service to which you are subscribed.
requestType	Request type supported by the message exchange. Optional.
requestReference	User reference of the request. Optional.
SnF	Indicates if the file transfer is done using the store-and-forward method. Valid values are True (use Store-and-Forward) and False (default—do not use Store-and-Forward). Required.
physicalFilename	Optional. Only for FileAct. For a FileAct Put, this is the full path and the physical name of the file to send. For a FileAct Get, this is the full path and the physical name of the file to save after the Get is completed.

Parameter	Description
logicalFilename	<p>This name is communicated to the server application. By default, this name is the physical name without the file path. Optional. Only for FileAct.</p> <p>For a FileAct Put, this is the logical name of the file to be saved based on the &lt;reception_dir&gt;/&lt;responder_dn&gt;/&lt;requestor_dn&gt;.</p> <p>For a FileAct Get, this is the logical name of the file to send based on the &lt;download_dir&gt;/&lt;responder_dn&gt;/&lt;requestor_dn&gt;.</p>
fileInfo	User information about the file transfer. Only for FileAct. Optional.
fileDesc	User description about the file transfer. Only for FileAct. Optional.
transferInfo	User information about the transfer. Only for FileAct. Optional.
transferDesc	User description about the transfer. Only for FileAct. Optional.
possibleDuplicate	<p>Indicates whether to include a trailer specifying that this message may be a duplicate. This is an optional component of the envelope that indicates that this message may already have been sent. For example, if the system crashes during the delivery of a message, another copy of the message could be sent, with this trailer included to indicate that it may be a duplicate.</p> <p>Possible values are TRUE and FALSE (default).</p> <p>Optional.</p>
messageID	Message identifier for resending a message if Possible Duplicate is set to TRUE. Optional.
deliveryNotification	<p>Indicates that the sender asked the receiver to send a delivery notification. Possible values are TRUE or FALSE. Optional.</p> <p><b>Note:</b> This parameter is only displayed when you select <b>True</b> for SnF or are performing a FileAct Put. If you are performing a Put operation, you can request the responder to send you a delivery notification and specify a different Delivery Notification DN and Request Type of Delivery Notification, if desired. If you are performing a Get operation, the responder can request Delivery Notification from the requestor after receiving the file. That setting for delivery notification is configured through the SWIFTNet Server adapter.</p>
requestTypeDelNotifn	Used to request a specific delivery notification message from the remote receiving server application when it returns the delivery notification (when Non Repudiation required and/or Delivery Notification are set to TRUE). Optional
messagePriority	<p>Indicates priority handling in the queue for store-and-forward only. Optional.</p> <p><b>Note:</b> This value is used as a selection criterion when delivering messages from a queue, and in SWIFTNet FileAct to influence the pace of the FileAct flow.</p>
nonRepudiation	Indicates whether non-repudiation is required. Possible values are TRUE (when enabled, trading partners cannot deny that they sent a request) or FALSE (default, indicating that non-repudiation is not required). Optional.

## Upgrading the SWIFTNetClient Business Process to Use the Integrated SWIFTNet Client Service

Now that the SWIFTNet Client service has been enhanced to support SSL, the SWIFTNet Client service has also been improved by integrating all the outbound services internally. To use the SWIFTNet Client service, you must upgrade the SWIFTNetClient business process. The upgraded BPML differs based on whether you are using InterAct or FileAct.

**Note:** If you previously installed an earlier version of the Application Standards Library, you do not need to upgrade the SWIFTNetClient business process again. However, you will need to reinstall the SWIFTNet MEFG Server (see *Using SWIFTNet* for more information).

### Upgrading the SWIFTNetClient Business Process for InterAct

If you are using InterAct, this is the complete BPML to execute the SWIFTNet Client service for InterAct:

```
<process name="SWIFTNetClient">
  <sequence name="SWIFTNetClientService">
    <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>
    <!-- build SWIFTNET request -->
    <operation>
      <participant name="SWIFTNetClientService"/>
      <output message="handleClientRequest">
        <assign to="." from="*" />
      </output>
      <input message="testing">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

### Upgrading the SWIFTNetClient Business Process for FileAct

If you are using FileAct, this is the complete BPML to execute the SWIFTNet Client service for FileAct:

```
<process name="SWIFTNetClientFA">
  <sequence name="SWIFTNetClientService">
    <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>
  </sequence>
</process>
```

```

    </input>
  </operation>
  <!-- build SWIFTNET request -->
  <operation>
    <participant name="SWIFTNetClientService"/>
    <output message="handleClientRequest">
      <assign to="." from="*" />
      <assign to="physicalFilename" from="'" />
      <assign to="logicalFilename" from="'" />
      <assign to="transferInfo" from="'" />
      <assign to="transferDesc" from="'" />
      <assign to="fileInfo" from="'SwCompression=None'"/>
      <assign to="fileDesc" from="'" />
    </output>
    <input message="testing">
      <assign to="." from="*" />
    </input>
  </operation>
</sequence>
</process>

```

## Enabling SWIFTNet Document Tracking

When you are creating or editing your SWIFTNet Client business process in the business process text editor, you can easily enable SWIFTNet 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.