

Sterling File Gateway

EBICS Integration Architect Help

Version 2.0



Contents

- Communities.....3**
 - About Communities.....3
 - Create a Community.....3
 - Edit a Community4
 - Delete a Community4
- Groups.....6**
 - About Partner Groups.....6
 - Create a Group.....6
 - Add Partners to Groups.....7
 - Add EBICS Partners to the All Partners Group.....7
 - View Groups a Partner Belongs To.....7
 - View Partners in a Group.....7
 - Delete a Partner from a Group.....8
 - Delete a Group.....8
- Routing Channel Templates.....9**
 - About Routing Channel Templates.....9
 - About Facts.....10
 - About Provisioning Facts.....12
 - Use Provisioning Facts.....13
 - Use Provisioning Facts with EBICS.....13
 - About Regular Expressions.....13
 - About Producer File Structures.....14
 - About Delivery Channels.....16
 - About Consumer File Structures.....16
 - Example EBICS Scenarios.....18
 - Create a Routing Channel Template20
 - List Routing Channel Templates22
 - Edit a Routing Channel Template22
 - Copy and Modify an Existing Routing Channel Template.....23
 - Delete a Routing Channel Template23
 - Ignore Routing of Temporary Files.....24
 - Route Determination.....24

Communities

About Communities

A community is required to create a partner. The community must exist before the partner is created. Integration Architects manage communities.

A community represents a way to organize partners for purposes of onboarding. In Sterling File Gateway, communities are used to limit or widen the selection of protocols available when creating partners. They are also used to enable listening or initiating modes of connection. For example, you may have the following community categories:

- FTP only
- SFTP only
- HTTP only
- Connect:Direct only
- Listening only
- Initiating only

Partner groups are another way of organizing Partners.

Create a Community

Integration Architects create, edit, and delete communities.

To create a community:

1. From the main menu, select **Participants > Communities**.
2. Click **add**.
3. Complete the information in the wizard for the new community.

Field	Description
Community Name	A meaningful name to describe the community. Required. Cannot be a name previously used in Sterling Integrator. Do not use spaces, tabs, or the following special characters:

Field	Description
	! @ # % ^ * () + ? , < > [] { } / ' \ " ;
Secret key for PGP signing	Select from list of keys assigned to AFTPGPPProfile. Required if any of the consuming partners belonging to this community require PGP signed data from the Router. See <i>Prepare to Use PGP</i> .
Secret key for PGP decryption	Select from list of keys assigned to AFTPGPPProfile. Required if any of the producing partners belonging to this community send PGP encrypted data to the Router. This secret key may be the same or different from the one for PGP signing. See <i>Prepare to Use PGP</i> .
Partner Initiates Protocol Connections to Mailbox	Select to make this option available when creating Partners belonging to this community. A unique mailbox is created for Partners that initiate connections. Depending on other selections, submailboxes are created for Partners to enable them to drop files off for routing, or pick up files routed to them.
Partner Listens for Protocol Connections	Select to enable protocols available to listening Partners belonging to this community. When selected, the following choices are available when creating Partners belonging to this community: <ul style="list-style-type: none"> • FTP or FTPS • Connect:Direct • SSH/SFTP If the System Administrator adds other protocols to the AFTEExtensionsCustomer.xml file, they are also provided as choices here.
Should member partners receive notifications that they are subscribed to?	This selection has no effect in Sterling File Gateway.

4. Confirm the information and save.
5. Close the Community window to return to Sterling File Gateway.

Route Provisioners can now create partners using the community.

Edit a Community

To edit a community:

1. From the main menu, select **Participants > Communities**.
2. Click **edit** next to the community in the list.
3. Modify the information as desired, click **Save**, and click **Return**.
4. Close the Community window to return to Sterling File Gateway.

Delete a Community

To delete a community:

1. From the main menu, select **Participants > Communities**.
2. Click **Delete** next to the community in the list.

Note: You cannot delete a community that has partner members. Delete the partners first.

3. Confirm the deletion.
4. Close the Community window to return to Sterling File Gateway.

Groups

About Partner Groups

Groups can contain multiple Partners and are used to constrain the use of routing channel templates to specific groups of Partners. Integration Architects create and delete Partner groups.

A group represents a way to organize partners for purposes of applying templates that govern file transfer policies. Templates are limited to the producer and consumer groups configured to use them. Assigning partners to groups allows them to participate in transfers using templates that specify their group. For example, you may have the following types of group names:

- Western division
- Northern division
- Southern division
- Eastern division

Upon installation of Sterling File Gateway, a group named All Partners is created. Every Partner is automatically associated with the All Partners group when they are created in Sterling File Gateway. The All Partners group cannot be deleted.

Sterling File Gateway Partner groups are separate and distinct from the groups managed in Sterling Integrator.

Plan your groups based on the Partners that will exchange files, recognizing characteristics they have in common. For example, you may have groups based on using any or all of the same:

- Protocols
- Encryption requirements
- File formats

You can delete groups you create, unless they are currently being used in a routing channel template. First delete the routing channel template, and then delete the group.

Create a Group

Create a group to constrain Partners to certain routing channel templates.

To create a group:

1. From the main menu, select **Participants > Groups**.
2. Click **Create**.
3. Type a **Partner Group Name**. Only alphanumeric, underscore, and <space> characters are allowed in the group name. The maximum numbers of characters for the group name is 255.
4. Click **Save**.

Add Partners to Groups

Add Partners to groups to enable them to use the appropriate routing channel templates. Partners cannot participate in file transfers until they are associated with at least one group.

Partners created directly in Sterling File Gateway are automatically placed in the All Partners group.

To add a Partner to a group:

1. From the main menu, select **Participants > Groups**.
2. Click **Add Partners**.
3. Highlight one or more **Identities** in the left pane. Highlight one or more groups in the right pane. Click **Execute**.

The highlighted Identities are added as Partners to the highlighted groups.

Add EBICS Partners to the All Partners Group

An EBICS Partner/Subscriber onboarded through the Sterling Integrator EBICS Subscription Manager and stored as an Identity is automatically associated with the File Gateway All Partners group to be eligible for Sterling File Gateway operations.

View Groups a Partner Belongs To

To view the groups a Partner belongs to:

1. From the main menu, select **Participants > Partners**.
2. A list of Partners is displayed in the left pane. To view the groups a Partner belongs to, select one from the list. The groups are displayed in the right pane.
3. A filter box is available on the list. Type characters in the box and click on the **Filter** icon or press **Enter**, to narrow down the list to items containing these characters. Scroll to select an item from the list. Backspace to remove the characters and press **Enter** to return to the full list.

View Partners in a Group

To view group membership:

1. From the main menu, select **Participants > Groups**.

2. A list of groups is displayed in the left pane. To view the Partners in a group, select one from the list. The Partners are displayed in the right pane.
3. A filter box is available on the list. Type characters in the box and click on the **Filter** icon or press **Enter**, to narrow down the list to items containing these characters. Scroll to select an item from the list. Backspace to remove the characters and press **Enter** to return to the full list.

Delete a Partner from a Group

To remove a Partner from a group, delete the Partner or delete and recreate the group.

Delete a Group

Delete groups that are no longer needed.

To delete a group:

1. From the main menu, select **Participants > Groups**.
2. Select the group to delete in the list. To select from the lists, type characters and **Enter** in the box to narrow the list to items containing these characters. Scroll to select an item from the list. Backspace to remove the characters and **Enter** to return to the full list.
3. Click **Delete**.

Note: You cannot delete a group that is currently being used in an RCT. First delete the RCT, then delete the group. You cannot delete the All Partners group.

Routing Channel Templates

About Routing Channel Templates

The routing channel template (RCT) defines the structure through which routing occurs. The RCT specifies producer and consumer mailbox structures and file structures. The RCT functions as a policy that mandates which partners can participate in various file transfer scenarios and which file formats they must use.

An RCT is required to create a routing channel, which establishes the producer-consumer relationship for file transfers. Integration architects create routing channel templates.

Routing channel templates are comprised of:

- Template type - static or dynamic, with a method of determining if dynamic
- Rules for handling special characters in the producer file name
- One or more producer groups
- One or more consumer groups
- One producer mailbox name pattern
- One or more producer file structures
- One or more delivery channels, each of which specifies one consumer file structure and one consumer mailbox name pattern

Template Types

Routing channel templates can be static or dynamic. A static routing channel template requires that the corresponding routing channels specify a single consumer and does not allow for any variability in that consumer.

A dynamic routing channel template specifies one of the following methods to determine the consumer for each file transfer:

- Specify the ConsumerName or ConsumerCode system facts in the producer file structure. For example, to dynamically identify the consumer based on file name, use parenthesis to group a portion of the file layer regular expression that represents the consumer and assign that grouping the ConsumerName or ConsumerCode system fact. If the following is entered:

```
(.+)_(.+)[.](.+)
ConsumerName,myFilenameBase,myFilenameExtension
```

The characters preceding the underscore in the file name determine the consumer that the file will route to. The consumer must be identified in the outer layer, or the first layer of an outer zip layer.

- Specify a business process and ProcessData element name. With this method, you create and deploy a business process that returns a ProcessData xml element that holds the consumer name.

About Facts

Facts are name-value-pairs that contain information about routing channels, active routes and the files being routed.

Some facts are specified during route provision, others are intrinsic to the route, and still others are determined from the producer file name. A fact is a named piece of known information for or about a routing channel, for example:

- A part of a file name
- The time a message arrived from the producer
- The name of the producer
- A user who is associated with the producer
- The name of the consumer
- A user who is associated with the consumer

When a fact becomes known it is frozen. During route execution, facts never lose their value or change values. A different route can have different values for the same facts.

Sterling File Gateway provides system facts and supports the specification of file name facts during routing channel template creation. Most system facts can be used to build out the pattern for producer and consumer mailboxes and to define the producer and consumer file structures in the routing channel template. System facts are given their value at the point of routing channel creation or during route execution. File name facts are given their values during route execution.

Use system facts as follows:

Fact Name	Description	When Derived	Where Used
ConsumerName	The name of the consuming partner	Derived: <ul style="list-style-type: none"> • During routing channel identification • As the result of a consumer identification business process • When associated to a file name regular expression parenthetic grouping 	Producer Mailbox, Consumer Mailbox, Producer File Structure, Consumer File Structure
ConsumerCode	The code of the consuming partner	Derived: <ul style="list-style-type: none"> • During routing channel identification • As the result of a consumer identification business process 	Producer Mailbox, Consumer Mailbox, Producer File Structure, Consumer File Structure

Fact Name	Description	When Derived	Where Used
		<ul style="list-style-type: none"> When associated to a file name regular expression parenthetic grouping 	
ProducerFilename	The name of the file that the producer sent	During early routing channel identification	Consumer Mailbox, Consumer File Structure
ProducerName	The name of the producing partner	During early routing channel identification	Producer Mailbox, Consumer Mailbox, Producer File Structure, Consumer File Structure
ProducerCode	The code of the producing partner	During early routing channel identification	Producer Mailbox, Consumer Mailbox, Producer File Structure, Consumer File Structure
RoutingTimestamp	When the routing began (when the route was identified)	Producer file arrival time	Consumer Mailbox, Consumer File Structure
ConsumerPgpExtension	<p>The extension of a file based on the consumer's PGP preferences.</p> <p>Value=</p> <ul style="list-style-type: none"> .pgp - consumer is getting a PGP file .asc - consumer is getting an ASCII encoded pgp file (blank) - consumer is not getting a PGP file <p>Only available for the PGP file layer type.</p>	PGP layer packaging time	Consumer File Structure

When using system facts in the producer and consumer mailbox or in the consumer file structure, certain notations are required:

Notation	Description
<code>\${factname}</code>	Replaced with the value of factname
<code>`\${fmt:factname}`</code>	Replaced with the value of factname, formatted according to fmt. The fmt string can be any format string supported by java.util.Formatter or a Timestamp value.
<code>`\${x:y:z}`</code>	Any use of more than one colon is an error and is rejected at RCT creation time.
<code>`\${factname`</code>	An unclosed reference is an error and is rejected.
<code>`\${`</code>	Produces a literal `\${`
<code>{factname}</code>	Is not an error, but produces a literal {factname} and is not replaced with the value of factname.

Example formats and results are:

- No format - For most facts this will just result in their value. For RoutingTimestamp this will produce an 8-digit string like "20100821" <year, month, and date> for August 21, 2010.

- s - The value of the fact
- S - The value of the fact mapped to all upper case
- .5s - the value of the fact right truncated to 5 characters

Formats for RoutingTimestamp use the Java Formatter class formats with an enhancement that allows multiple formats to be specified for one value. Essentially, this is either a 't' or 'T' followed by multiple time format specifiers. 'T' forces the results to upper case. In addition pieces of literal text may be included in ' characters.

Examples of RoutingTimestamp are:

- tYmd - An 8-digit string consisting of a 4-digit year, 2-digit month, and 2-digit day of the month. For example, 20080201 (default).
- tHMS - A 6-digit Hour, Minute, Second value using a 24-hour clock. For example, 053027.
- tMSp - A 5 or 6-digit Hour, Minutes, Seconds value using a 12-hour clock. For example, 53027am.
- TMS' 'p - Like above, but in upper case with an embedded space. For example, 53027 AM.
- Ta' -'b' -'d' -'Y - Textual date, in upper case. For example, FRI-APR-21-2008.
- tYmdHMSL - A 17-digit string consisting of a 4-digit year, 2-digit month, 2-digit day, 3-digit minute, 3-digit second, and 3-digit millisecond. For example, 20090813142111920.

Note: For more information about the Java Formatter class formats, see <http://java.sun.com/j2se/1.5.0/docs/api/java/util/Formatter.html>.

In addition to using system facts, you can create your own facts in the producer file structure. User-created facts must be prefixed with the string "my" to clearly delineate between custom and system facts. Custom facts may be used in the consumer mailbox pattern (for mailboxes created on demand), and in the consumer file structure.

System facts and custom facts are searchable on the Advanced Search page. Facts can be used to enforce policy where the fact values must match when used in more than one layer.

About Provisioning Facts

Provisioning facts provide a controlled way to customize particular routing channels (RCs) within the context of a particular routing channel template (RCT). For example, if you need to create a mailbox of the form

```
/PartnerName/UserId/
```

within an RCT, the UserId can be specified as a parameter for the RC using the provisioning fact. At provision time, while selecting an RCT, producer, and consumer, an additional prompt is made to specify the UserId. The value of UserId is then bound to the new RC and can be used for creating mailboxes (at either provision time or routing time) as well as creating file names (at routing time). Provisioning facts are available for static and dynamic templates.

You are not required to define any provisioning facts when creating an RCT.

You can edit provisioning facts as part of RCT edit. However, provisioning fact values cannot be edited after being set in a routing channel.

You can add a maximum of ten provisioning facts to an RCT.

The same provisioning fact can be used in more than one template. For example, both Template 1 and Template 2 can have a "myUserID" provisioning fact.

Use Provisioning Facts

To use provisioning facts:

1. When you create a routing channel template, use the Provisioning Facts tab to create a provisioning fact. The provisioning fact name must start with "my". Valid characters are alphanumeric in English only and the hyphen.
2. Refer to the fact created in step 1 using the “\${myfactname}” notation in the producer mailbox path, consumer mailbox path, or the “File name format” field within the layers of the consumer file structure.
3. During creation of a routing channel, the user is prompted for the value to be assigned to the fact.
4. The value of the fact is substituted for the references used in the producer mailbox path, consumer mailbox path, or the consumer file structure. It also becomes a fact attached to the executing route and can be used like a discovered fact, for example as input to a custom layer.

Use Provisioning Facts with EBICS

For EBICS operations, you add a provisioning fact to the routing channel template to ensure that mailbox path structures match what was previously setup during EBICS Partner/Subscriber creation. If you do not use provisioning facts, mailboxes may not be appropriately set up to support EBICS operations. Generally, a mailbox path structure of the format /User/Partner/Inbox or /User/Partner/Outbox is needed to support EBICS transactions.

To define a mailbox path appropriate for EBICS operations:

1. In the Routing Channel Template wizard, click on the Provisioning Facts tab, and specify the following:
 - Display Label - User ID
 - Fact name - myUserId
 - Description - Enter the User ID for the consumer mailbox.
2. Click Save to save the provisioning fact definition.
3. In the Routing Channel wizard, select a template of the form created in step 1.
4. Select the producer and consumer.
5. Save the routing channel.

About Regular Expressions

Regular expressions are a standardized pattern matching language. Some suggested references for learning more about regular expressions are:

<http://www.regular-expressions.info/>

http://en.wikipedia.org/wiki/Regular_expression

<http://java.sun.com/docs/books/tutorial/essential/regex/index.html>

<http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html>

About Producer File Structures

A producer file structure contains a description of the basic content structure and naming conventions for files the producer sends. The Integration Architect specifies a pattern for the producer mailbox and a producer file structure when creating a routing channel template.

The default pattern for the producer mailbox path is:

```
/${ProducerName}
```

The pattern can be specified using the following:

- System facts (for example, `/${ProducerName}/Inbox`) where the actual name would be determined dynamically
- Explicit strings (for example, `/${ProducerName}/myInbox`)
- A combination of system facts and explicit strings. The mailbox pattern must always start with `/${ProducerName}`

Note: If creating a dynamic routing channel template, do not specify `ProducerName` or `ProducerCode` in the producer mailbox path. In the New Delivery Channel window, check the box, *If checked, mailboxes matching this pattern may be created on demand.*

A producer file structure specifies the kind of file that producers are expected to send to Sterling File Gateway. A file structure is an ordered list of file layers, from outside to inside. A producer file structure may have one or more file layers. The layers are of a certain type, have a specific regular expression pattern, and may specify facts to be defined when a file matching the pattern is received.

There are two types of layers:

- Container layers- contain other container or non-container layers. For example, a PGP layer can contain a ZIP layer and a ZIP layer can contain a TEXT layer. A zip file can only contain multiple files if it is the outermost layer. All layers except the innermost layer must be one of:
 - ZIP
 - GZIP
 - PGP
- Non-container layers - or primitive layers, do not contain any nested layers. The innermost layer must be a primitive layer of type:
 - Text
 - Unknown

There is no limit to the number of layers for a file structure. Certain processing operations, such as the special character handling feature, only apply to the outermost layer. Also, route identification occurs at the outermost layer, except in the case of an outer ZIP file layer. With an outer ZIP layer, the second layer is also used to perform route identification.

If the layer type of the consumer file structure is text, the layer types of every producer file structure must be text and must not be unknown. If the layer type of the consumer file structure is unknown, the producer file structure can include either unknown or text.

By using regular expressions, a powerful standardized pattern matching language, the Sterling File Gateway application can match file names and gather facts related to file names. In defining facts for a particular regular expression, you have the flexibility to use the following system facts:

Fact Name	Description
ConsumerFilename	The name of the ultimate consumer file.
ConsumerName	The name of the consuming partner
ConsumerCode	The code of the consuming partner Note: The ConsumerCode may or may not be the same as the ConsumerName. The ConsumerCode is derived from the consuming partner name and characters may be truncated (if the partner name exceeds 24 characters), replaced (if the partner name includes characters that are not alphanumeric), or removed (if the partner name includes one or more spaces).
ProducerFilename	The name of the file the producer sent
ProducerName	The name of the producing partner
ProducerCode	The code of the producing partner
RoutingTimestamp	Date when routing began (when the route was identified)

Or, you can create your own facts. User-created facts must be prefixed with the string "my" to clearly delineate between custom and system facts.

Facts are useful for describing the meaning of the regular expression language. The regular expression carries in it the definition of the file expected from the producer. Because a file name may carry different categories of information, it is helpful to divide those categories of information into groups. In the regular expression, each group of information is contained within parentheses "()". Each group within the regular expression must have a corresponding fact, either system or custom. Consider the following example:

File Format in Regular Expression

```
(\p{Alnum}+[-_]?(\d{4})_(\d{2})_(\d{2})\.zip)
```

where:

- the full expression, `(\p{Alnum}+[-_]?(\d{4})_(\d{2})_(\d{2})\.zip)`, = capturing group 1 with fact = "mySanitizedFilename"
- `(\d{4})` = capturing group 2 with fact = "myFileYear"
- `(\d{2})` = capturing group 3 with fact = "myFileMonth"
- `(\d{2})` = capturing group 4 with fact = "myFileDay"

The corresponding file name pattern group fact name entry for the above expression is:

```
mySanitizedFilename,myFileYear,myFileMonth,myFileDay
```

You can use the same fact in different layers of a producer file structure or in multiple file structures for the same routing channel template. During execution the value of this fact must not change. If it does change, Sterling File Gateway generates an error.

The simplest pattern that matches any name, and defines no facts, is:

```
.+
```

The File name pattern as regular expression field is case insensitive when matching file names.

After you specify a file layer type and click **Next**, you cannot change the file layer type. You can click **Edit** to change the fact names or regular expressions. To change the file layer type, click **Delete**, then click **Add** to create a new file layer.

About Delivery Channels

The delivery channel defines what will be delivered to a consumer.

A routing channel template contains one or many delivery channels. Each delivery channel consists of:

- One consumer mailbox name pattern
- One consumer file structure

The mailbox name pattern can be specified using the following:

- System facts (for example, `/${ConsumerName}/Inbox`) where the actual name would be determined dynamically
- Explicit strings (for example, `/${ConsumerName}/myInbox`)
- A combination of system facts and explicit strings. The mailbox pattern must always start with `/${ConsumerName}`

The Integration Architect creates a delivery channel as part of the process of creating a routing channel template.

About Consumer File Structures

A consumer file structure contains a description of a file's basic content structure and naming conventions.

A consumer file structure specifies the kind of file that consumers expect from Sterling File Gateway. A file structure is an ordered list of file layers, from outside to inside. The layers are of a certain type and they may have a specific format pattern and may include unique parameters.

The Integration Architect must create a consumer file structure when creating a delivery channel.

To create a file structure with multiple file layers, the outside layer must be a container layer of one of the following types:

- ZIP
- GZIP
- PGP

The innermost layer of a producer file structure must be a non-container layer of one of the following types :

- TEXT
- Unknown

If the layer type of the consumer file structure is text, the layer types of every producer file structure must be text and must not be unknown. If the layer type of the consumer file structure is unknown, the producer file structure can be either unknown or text.

For each layer, the file name format pattern is used to generate the new file name. The file name format pattern uses system or custom facts to define how the consumer file should be named. The following system facts may be used in building the file name format pattern:

Fact Name	Description	When Derived
ConsumerFilename	The name of the ultimate consumer file	During routing channel identification or as the result of a consumer identification business process
ConsumerName	The name of the consuming partner	During routing channel identification or as the result of a consumer identification business process
ConsumerCode	<p>The code of the consuming partner</p> <hr/> <p>Note: The ConsumerCode may or may not be the same as the ConsumerName. The ConsumerCode is derived from the consuming partner name, and characters may be truncated (if the partner name exceeds 24 characters), replaced (if the partner name includes characters that are not alphanumeric), or removed (if the partner name includes one or more spaces).</p>	During routing channel identification or as the result of a consumer identification business process
ProducerFilename	The name of the file that the producer sent	During early routing channel identification
ProducerName	The name of the producing partner	During early routing channel identification
ProducerCode	<p>The code of the producing partner</p> <hr/> <p>Note: The ProducerCode may or may not be the same as the ProducerName. The ProducerCode is derived from the producing partner name, and characters may be truncated (if the partner name exceeds 24 characters), replaced (if the partner name includes characters that are not alphanumeric), or removed (if the partner name includes one or more spaces).</p>	During early routing channel identification
RoutingTimestamp	<p>When the routing began (when the route was identified)</p> <hr/> <p>Note: The RoutingTimestamp format for file names and mailboxes is restricted so that the generated file name is legal. Characters from literal text, text fact values, and formatted timestamp fact values must be valid.</p> <p>For example, the format string: <code>\${tH}:'M:RoutingTimestamp}</code> is a legal format, but it will produce an invalid</p>	Producer file arrival time

Fact Name	Description	When Derived
	name, because it produces a name like 05:23. Colons are not allowed.	
ConsumerPgpExtension	<p>The extension of a file based on the consumer's PGP preferences. Only available for the PGP file layer type.</p> <p>Value=</p> <ul style="list-style-type: none"> • .pgp - consumer prefers a PGP file • .asc - consumer prefers an ascii encoded pgp file • (nothing) - consumer does not prefer a PGP file 	PGP layer packaging time

If custom facts are used, they must exactly match the custom facts specified in the producer file structure.

In the following example, the file name format pattern uses both system and custom facts:

```
${mySanitizedFilename}_${RoutingTimestamp}.zip
```

In this example, "mySanitizedFilename" was defined as a custom fact in the producer file structure, and "RoutingTimestamp" is a system fact. When Sterling File Gateway receives the producer's file, it will derive the facts and then use the facts to determine the values to place in the consumer file name.

Example EBICS Scenarios

An example inbound EBICS scenario is:

1. An EBICS order is uploaded from an EBICS client to an EBICS server in a file upload (FUL) function.
2. The EBICS Order Data Processor (ODP) unpackages the payload to its raw inner .DAT format.
3. The EBICS ODP deposits the file into a mailbox of the structure User/Partner/Inbox (according to the EBICS specification).
4. Sterling File Gateway routes the file from the mailbox for downstream processing and ultimate delivery to the consumer according to the routing channel (RC), which specifies the template to use for the route. The template uses provisioning facts that established the mailbox path and user ID that were determined for the routing channel. The details of an example template suitable for an inbound EBICS scenario are:

Field	Value
Template Name	EBICSFUL
In the Type tab	
Type	Static
In the Special Characters tab	
Special Characters	None
In the Groups tab	
Groups	All Partners for producers and consumers
In the Provisioning Facts tab	

Field	Value
Fact Name	myUserId
Display Label	User ID
Description	The user ID of the producer
In the Producer tab	
Producer Mailbox Path	/\${myUserId}/\${ProducerName}/Inbox
Producer File Structure	Click Add
Producer File Type	Unknown
File name pattern as regular expression	{{[^_]+}_([^_]+)_.(+)([.]DAT)}
File name pattern group fact names, comma delimited	myHostId,myFileFormat,mySuffix,myExtension
In the Consumer tab	
Delivery Channel Description	Click Add
Pattern for Consumer Mailbox Path	/\${ConsumerName}/Inbox
If checked, mailboxes matching this pattern may be created on demand	Do not check the box
Consumer File Structures	Click Add
Consumer File Type	Unknown
Consumer file name format	\${myHostId}ara_\${mySuffix}\${myExtension}

An example outbound EBICS scenario is:

1. Sterling File Gateway routes an incoming file.
2. The file is packaged according to the EBICS configuration.
3. The file is deposited as a message in an EBICS partner's mailbox in the structure /User/Partner/Outbox based on a routing channel, which specifies the template to use for the route. The template uses provisioning facts that establish the mailbox path and user ID for the routing channel. The details of an example template suitable for an outbound EBICS scenario are:

Field	Value
Template Name	EBICSFDL
In the Type tab	
Type	Static
In the Special Characters tab	
Special Characters	None
In the Groups tab	
Producer Group	All Partners
Consumer Group	All Partners
In the Provisioning Facts tab	
Fact Name	myUserId
Display Label	User ID

Field	Value
Description	The user ID of the consumer partner
In the Producer tab	
Producer Mailbox Path	/\${ProducerName}
Producer File Structure	Click Add
Producer File Type	Text
File name pattern as regular expression	.+
File name pattern group fact names, comma delimited	
In the Consumer tab	
Delivery Channel Description	Click Add
Pattern for Consumer Mailbox Path	/\${myUserId}/\${ConsumerName}/Outbox
If checked, mailboxes matching this pattern may be created on demand	Check the box
Consumer File Structures	Click Add
Consumer File Type	Text
Consumer file name format	\${ProducerFilename}

4. The EBICS partner, using an EBICS client, initiates a file download (FDL) request to the EBICS server.
5. The file in the consumer mailbox is picked up by the EBICS server.
6. The packaged file is transferred from the EBICS server to the EBICS client.

Create a Routing Channel Template

To create a routing channel template:

1. From the main menu, select **Routes > Templates**.
2. Click **Create**.
3. On the **Type** tab, enter a descriptive name for the template. Only alphanumeric, underscores, and <space> characters are allowed in the template name.

The following are system-designated prefixes and must not be used for user-created templates:

- AFT
 - FileGateway
 - FG
4. Select **Static** or **Dynamic**. If **Dynamic**, select how the consumer is determined, whether by facts or by business process. If by facts, specify the **ConsumerName** or **ConsumerCode** facts in every producer file structure. If by business process, specify the business process name and the element name in **ProcessData** that identifies the consumer.
 5. On the **Special Characters** tab, select how special characters in producer file names should be handled. Use this when the file name the producer gives a file is incompatible with the naming convention the consumer uses. Select from the following:

Selection	Other Input and Results
None	Do not change any special characters.
Substitute characters individually	<p>Enter which characters to substitute with what new characters. The number of characters specified in the search for and replace fields must match. The characters are case-sensitive.</p> <p>For example, Search for character sequence ABC, Replace with 123 would translate a producer file name of:</p> <ul style="list-style-type: none"> • Abbot.txt to a consumer file name of 1bbot.txt • ABBOT.TXT to a consumer file name of 122OT.TXT <p>Search for abcdef, Replace with ABCDEF would translate producer a file name of:</p> <ul style="list-style-type: none"> • abbot.txt to a consumer file name of ABBot.txt
Replace characters, then omit consecutive replacements	<p>Enter which sequence to substitute with what new sequence.</p> <p>For example, Search for [*%\$] Replace with -, would translate all of the following producer file names:</p> <ul style="list-style-type: none"> • A*B.txt • A***B.txt • A%%%%B.txt • A*B.txt • A\$\$**%B.txt • A*\$%\$%*\$*%B.txt <p>to a consumer file name of A-B.txt</p>
Remove characters	Enter the characters to remove from the producer file name to create the consumer file name.
Remove Windows invalid characters \.*?"<> ;%	Characters that are invalid or cause problems in the Windows operating system are removed from producer file names prior to delivering to consumers.
Remove UNIX invalid characters ;\!#\$%*"?'<>)	Characters that are invalid or cause problems in the UNIX operating system are removed from producer file names prior to delivering to consumers.
Remove all characters, except alphanumeric, dash, and period	All special characters are removed from producer file names prior to delivering to consumers.

6. On the **Groups** tab, click **Add** beneath each group box. Select from the drop down lists the producer groups and consumer groups that can use the template. You can select one or more producer groups and consumer groups by clicking **Add** and selecting additional groups from the lists.

Note: Select the **All Partners** group in the producer and consumer lists to enable all partners present in the application to use the RCT.

7. On the **Producer** tab, specify the producer mailbox name pattern and producer file structure. Mailbox paths are case insensitive and should therefore be unique without regard to case.
8. On the **Consumer** tab, specify the consumer mailbox pattern. Check the following:
 - If checked, mailboxes matching this pattern may be created on demand. For a dynamic template, this mailbox creation at run time is required.

Note: The root mailbox for the user must already exist; only a submailbox for the partner is created by Sterling File Gateway.

- If checked, the delivery channel will attempt to deliver to a listening protocol if available.
-

Note: If you have two delivery channels and you only want to send to that listening protocol once, uncheck this option.

9. Specify the delivery channel, including the consumer file structure. The innermost layer of the consumer file structure must match the innermost layer of the producer file structure, or be Unknown. The consumer file structure defines the file name format delivered to the consumer. To pass the file name through unchanged, enter: `${ProducerFilename}`
-

Note: There can only be one consumer file structure for each delivery channel. If you click **Add** and specify another file structure, it will overwrite the previously defined file structure.

10. Click **Save** to complete the routing channel template. Sterling File Gateway validates the routing channel template and displays errors for invalid criteria. You will get a confirmation when the template is successfully validated and saved.

List Routing Channel Templates

To list the routing channel templates available in Sterling File Gateway and view the details of each template:

1. From the main menu, select **Routes > Templates**.
2. A list of routing channel templates is displayed in the left pane. To view details of a template, select one from the list. The details are displayed in a tree form in the right pane.
3. A filter box is available on the list. Type characters in the box and click on the **Filter** icon or press **Enter**, to narrow down the list to items containing these characters. Scroll to select an item from the list. Backspace to remove the characters and press **Enter** to return to the full list.

Edit a Routing Channel Template

You should only edit routing channel templates during a time of no activity, such as a maintenance window. You can disable routing for maintenance by disabling the routing rule (FileGatewayRoutingRule) or by disabling the routing schedules (MailboxEvaluateAllAutomaticRules and MailboxEvaluateAllAutomaticRulesSubMin.)

There are some constraints for editing routing channel templates:

- You cannot edit a template currently in use by a routing channel.
- You cannot change the name of the template you are editing.
- You cannot edit templates with names prefixed with AFT, FG, or FileGateway.

Integration Architects can edit routing channel templates within these constraints.

To edit a routing channel template:

1. From the main menu, select **Routes > Templates**.

2. Select one of the listed routing channel templates. To select from the lists, type characters and **Enter** in the box to narrow the list to items containing these characters. Scroll to select an item from the list. Backspace to remove the characters and **Enter** to return to the full list.
3. Click **Edit**.
4. If the selected template does not have any dependent routing channels, the edit template wizard opens.
5. Make the desired changes to the template.
6. Click **Save**.

Note: When a template is edited, the original instance of the template is no longer available and attempts to link to the template instance from event details will result in an error message.

Copy and Modify an Existing Routing Channel Template

To create new routing channel templates with similar characteristics, you can copy an existing template.

To copy a routing channel template:

1. From the main menu, select **Routes > Templates**.
2. Select from the list of templates the one you want to copy.
3. Click **Copy**.
4. Enter a new name for the new template.
5. Modify the items you want to change.
6. Click **Save** to complete the routing channel template. Sterling File Gateway validates the routing channel template and displays errors for invalid criteria. You will get a confirmation when the template is successfully validated and saved.

After you save the template, you cannot edit it.

Delete a Routing Channel Template

Delete RCTs that are no longer needed. If you need to change an RCT, delete it and create a new one with the new parameters.

To delete an RCT:

1. From the main menu, select **Routes > Templates**.
2. Select the routing channel template to delete, and click **Delete**.

Note: You cannot delete an RCT that is currently being used in an RC. First delete any RCs using the template, then delete the RCT.

3. Confirm that the correct RCT will be deleted.

The RCT is deleted, but information about the operations that used it remain until purged.

Ignore Routing of Temporary Files

Some organizations configure their FTP environment to place a file in a mailbox with a temporary name, then rename the file, with the intention of routing only the renamed file. Sterling File Gateway provides the capability to assign a special status, Ignored, to these temporary files so they are not listed as Failed.

To configure Sterling File Gateway to ignore routing of temporary files:

1. Open the following file: `<install_dir>\properties\customer_overrides.properties`
2. Add the following parameter to the file: `filegateway.ignoreFilename=TEMP(.*)[.]txt`
where `TEMP(.*)[.]txt` is the regular expression pattern that should be ignored. This pattern is checked before any patterns in a routing channel template. The value for `ignoreFilename` is limited to 255 characters.
3. Run `./setupfiles.sh` (UNIX) or `setupfiles.cmd` (Windows).
4. Restart Sterling File Gateway.

Route Determination

To perform day-to-day operations, it is not required that you understand how Sterling File Gateway handles file transfer activities. The following information is provided as background only.

When a producer sends or uploads a file, that file always lands in a producer mailbox. In preset intervals, Sterling File Gateway checks all the producer mailboxes that it monitors and initiates routing on any newly arrived files. Sterling File Gateway will use the producer partner name and producer mailbox to determine all routing channels that match. For each routing channel match, the producer file structure is compared to the outer layer file name (for ZIP files, comparisons are also made on inner (one layer down) file names). If there is a mismatch between the file the producer sent and the file that was expected by the template, the file is not routed.

At times, multiple routing channels may seem appropriate for a transfer because the particular producer and producer mailbox may be used more than once in Sterling File Gateway routing channels. All routing channels that match are considered as candidates. The routing channel candidates are then grouped for each file. For each file, the winning routing channel is the candidate with the highest precedence. The precedence is determined based on the format layering and filename pattern in the template.

The following steps are involved in determining which routing channel wins:

1. Determine routing channel candidates. For example, if Producer Mailbox is `/Producer1`, identify the routing channels that use `/Producer1`.
2. Match the producer file structure. For example, if the producer file structure is `ZIP/Unknown`, identify the routing channel candidates that use a template with the `ZIP/Unknown` producer file structure.
3. Determine precedence. Of the routing channel finalists from step 2, find the one with the longest regular expression for producer file name. For example, a regular expression of `(\p{Alnum}+[-_]?(\d{4})_(\d{2})_(\d{2})\zip)` is longer than a regular expression of `(.+)`.
4. Select the winner. In the previous example, the regular expression of `(\p{Alnum}+[-_]?(\d{4})_(\d{2})_(\d{2})\zip)` is the winner.
5. Route to the consumer. As configured in the winning routing channel, according to format specified in the consumer file structure.

After this process, Sterling File Gateway knows the consumer to send the file to and the required file name and layering format for that consumer. It uses this information to transform the file and deliver it to the consumer. It is possible that more than one delivery can be made to a consumer, depending on template configuration. If the template specifies more than one delivery channel, Sterling File Gateway delivers files according to the requirements of each.

At each stage of processing, Sterling File Gateway generates events that provide detail about the arrival of the file, the partners involved in the transfer, including pickup and drop off mailboxes, the template that governs the transfer, and the file format and name transformations. These event details also link to lower level objects – for example, business processes, data flows, and communication sessions – that allow in-depth analysis and troubleshooting if necessary.

Several factors can impact expected Sterling File Gateway operations, including configuration changes, missing configurations resulting from out-of-band partner setup, template design, and discrepancies in expected and received files. With detailed error tracking and customizable notifications, these processing errors can be manually checked, and rerouted after required corrections.