INTERCOPE

BOX
MESSAGING HUB

Version 3 Release 23 (V3R23)

FileAct

Administration and Operation

Table of Contents

1 Introduction

This document describes the **BOX** FileAct functionality.

The document includes an overview of the architecture and of the concept, descriptions of how to perform administrative and operational tasks and some configuration help.

The reader of this document must be familiar with the SWIFT architecture and terminology.

We also recommend having at least the following SWIFT documents at hand:

- SWIFTNet FileAct Implementation Guide
- SWIFTNet 7.0 Service Design Guide

1.1 Important Release Note

As from Version 3 Release 20 (V3R20) there is a new Configuration Management (CM) implementation for BOX.

This new CM concerns the configuration of all workflow-related objects (Instruction Patterns, IPS, Analysis and Content Processing modules, Address Books, etc.) and the UPM configuration, i.e. practically all aspects of configuration except for message objects (Messages, Message History, Archives) and SWIFT Reference Data, such as BIC Directory.

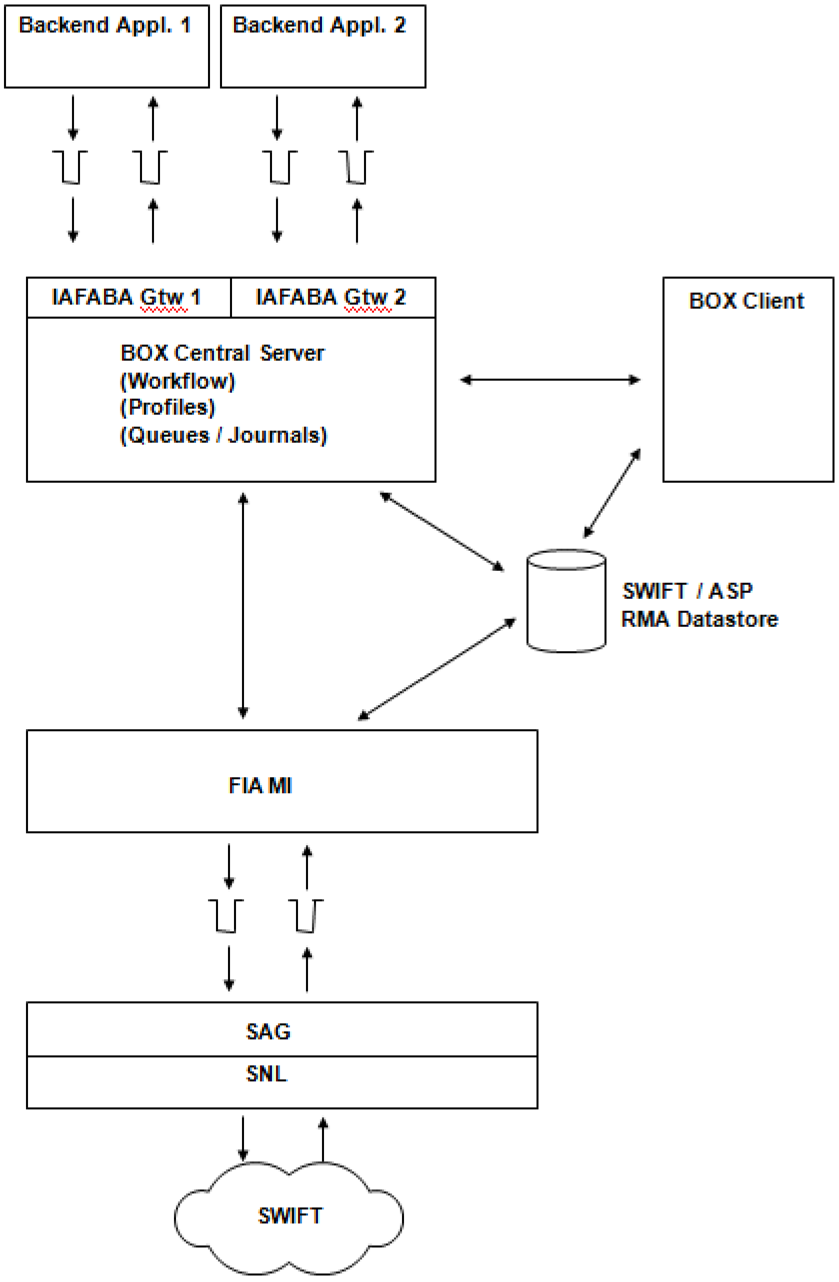
In this document we have assumed that all configuration objects are in an editable mode (**‘Yellow Mode’**) even if the figures do not always reflect this.

In order to understand the Concept of the Configuration Management and how to handle the new implementation, you must read the chapter "Configuration Management" in the document **Box Administration Guide** (box_admig_<releasenumber>.pdf).

2 Overview

BOX FileAct allows sending files to and receiving files from SWIFTNet via a SWIFT Alliance Gateway, without requiring any additional SWIFT connectivity solution.

The BOX FileAct functionality uses the **BOX Messaging Interface** (FIA MI) that connects BOX FileAct channels for SWIFT messages to SAG and a **BOX InterAct/FileAct Backend Application Gateway** (IAFABA Gtw) to backend applications in order to receive SWIFT Input messages and to route SWIFT Output messages.



2.1 Resources

Messaging Interface Module Local Channel Groups (MI LCG), **Submission Profiles** and the **SWIFT ASP** are called **resources**.

Resources are created and maintained on Enterprise level so that a Service Provider can administrate the resources.

MI LCGs and Submission Profiles always get **2 owners** assigned at creation time.

The **SWIFT ASP** resource is downloaded and imported on Enterprise level but is used by **all Company Level objects**!

2.1.1 Messaging Interface LCGs

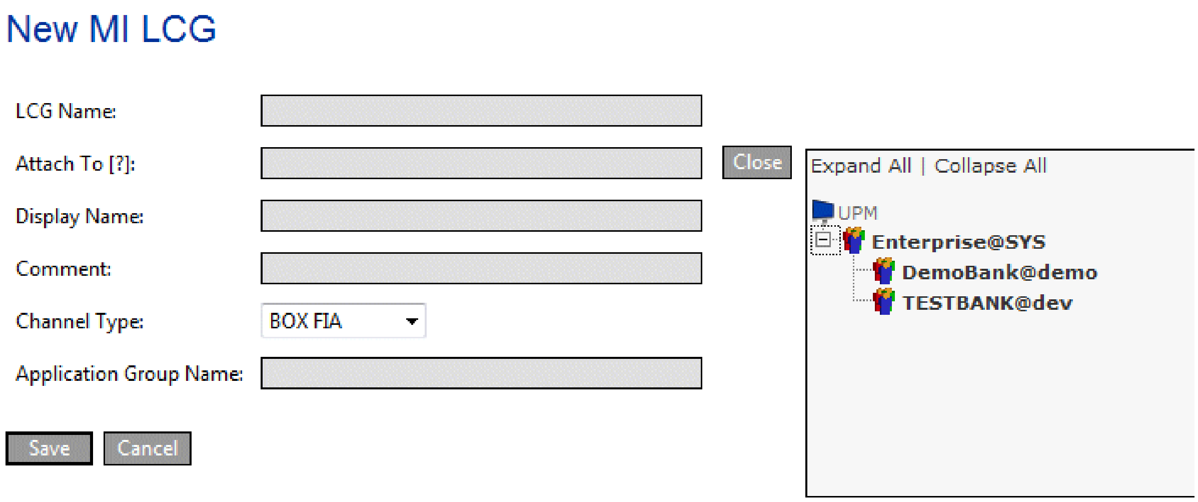
MI LCGs are created on **Enterprise level** but each MI LCG must be assigned an **operator owner** on Company level so that the Service Provider can administrate these resources for a given company.

This means that each MI LCG **always has 2 owners**:

- A **system owner** (referred to as “owner”) which is always on **Enterprise** level.
  The Enterprise user maintains the resource (MI LCG) according to the needs of the company for which the resource was created.
- An **operator owner**. which is always on **Company** Level

When creating an MI LCG on Enterprise level (see 4.5), you have to assign an operator owner to it by attaching it to a Company level node in the existing UPM Tree.

Press **Select Owner**.



New MI LCG

LCG Name:

Attach To [?]:

Display Name:

Comment:

Channel Type: BOX FIA

Application Group Name:

Select the Company level object to which the LCG shall be attached.

Press **Close**.

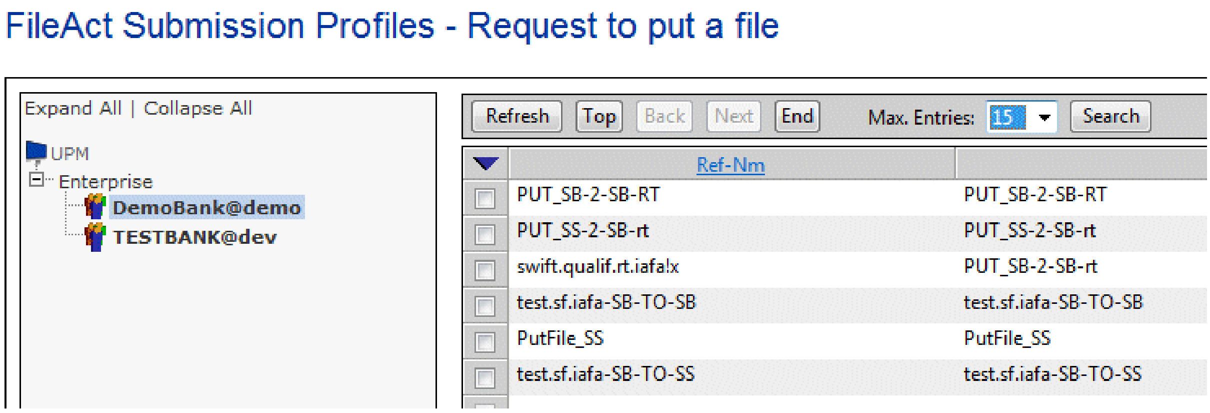
This will allow the operator to use the resource but not to modify it.

For more information how to create MI LCGs see chapter 4.5.

2.1.2 Submission Profiles

When creating a new Submission Profile on Enterprise level, you have to select an “operator owner” on Company level from the UPM tree on the left hand side.

This Profile can be used only by this company.



FileAct Submission Profiles - Request to put a file

	Ref-Nm	
□	PUT_SB-2-SB-RT	PUT_SB-2-SB-RT
□	PUT_SS-2-SB-rt	PUT_SS-2-SB-rt
□	swift.qualif.rt.iafa!x	PUT_SB-2-SB-rt
□	test.sf.iafa-SB-TO-SB	test.sf.iafa-SB-TO-SB
□	PutFile_SS	PutFile_SS
□	test.sf.iafa-SB-TO-SS	test.sf.iafa-SB-TO-SS

A Submission Profile can also be created on Company level. In this case the Company automatically is the “operator owner”.

For more information regarding Submission Profiles see **chapter 4.11**

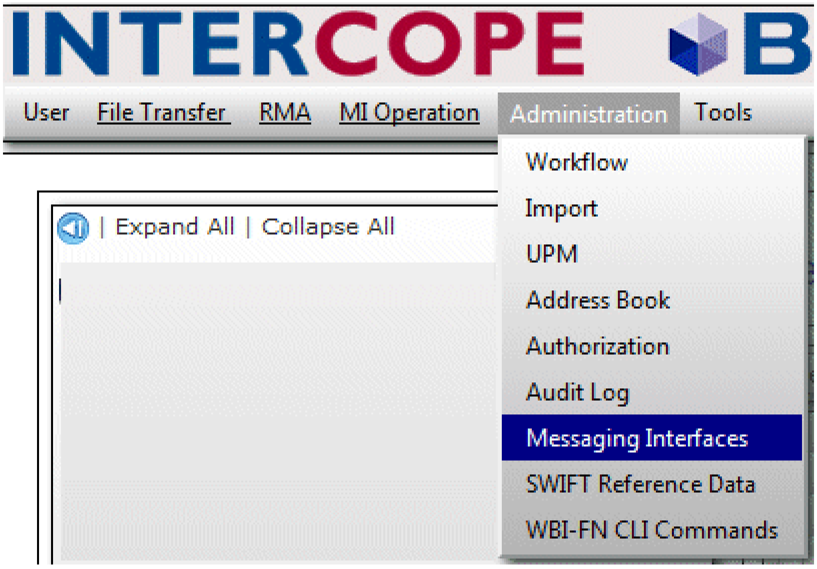
2.1.3 SWIFT ASP

Application Service Profiles (ASP) were introduced by SWIFT for SWIFTNET 7.

ASPs can be downloaded from the SWIFT web site and then imported to BOX.

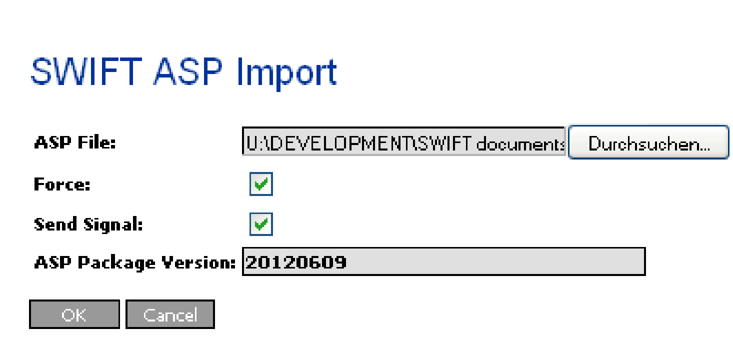
This is done on Enterprise level.

To import an ASP select **Administration / Messaging Interfaces** from the main menu.



Select **SWIFT ASP Importer** from the tree-view on the left hand side of the page.

The **SWIFT ASP Import** page is displayed.



SWIFT ASP Import

ASP File: U:\DEVELOPMENT\SWIFT document:

Force: ☑

Send Signal: ☑

ASP Package Version: 20120609

Browse for the ASP file you want to import.

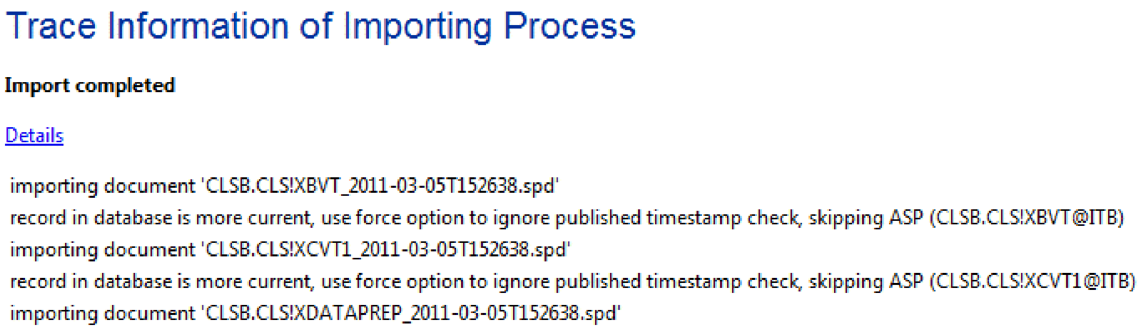
Activate the **Force** box in order to overwrite existing entries.

Send an update signal to the server by activating the **Send Signal** box.

Optionally enter the ASP version of the current ASP file.

Press **OK**

If you press the **Details** link on the Trace Information page, you will obtain details of the importing process:



Trace Information of Importing Process

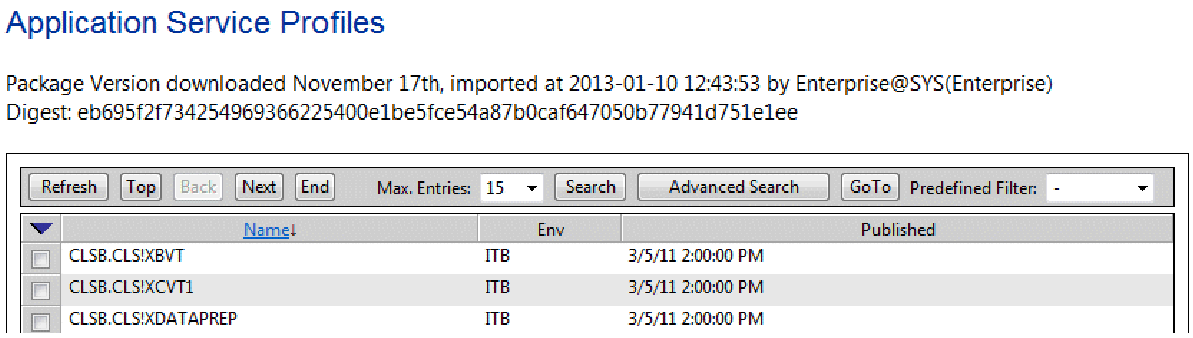
Import completed

Details

```
importing document 'CLSB.CLS!XBVT_2011-03-05T152638.spd'
record in database is more current, use force option to ignore published timestamp check, skipping ASP (CLSB.CLS!XBVT@ITB)
importing document 'CLSB.CLS!XCVT1_2011-03-05T152638.spd'
record in database is more current, use force option to ignore published timestamp check, skipping ASP (CLSB.CLS!XCVT1@ITB)
importing document 'CLSB.CLS!XDATAPREP_2011-03-05T152638.spd'
```

By scrolling down to the bottom of the page you will find a summary of the importing process.

The version information entered into the **ASP Package Version** field, the time of the import and the user that performed the import will be displayed in the **Application Service Profiles** list.



Application Service Profiles

Package Version downloaded November 17th, imported at 2013-01-10 12:43:53 by Enterprise@SYS(Enterprise)
Digest: eb695f2f734254969366225400e1be5fce54a87b0caf647050b77941d751e1ee

Max. Entries: 15 Predefined Filter: -

	Name↓	Env	Published
□	CLSB.CLS!XBVT	ITB	3/5/11 2:00:00 PM
□	CLSB.CLS!XCVT1	ITB	3/5/11 2:00:00 PM
□	CLSB.CLS!XDATAPREP	ITB	3/5/11 2:00:00 PM

2.1.4 The SWIFT ASP Importer Tool

The SWIFT Application Service Profile Importer (ASP Importer) is a tool to import Application Service Profile and FINCopy Profile files into the BOX database.

This tool is described in the document **BOX Administration Guide**.

3 Required UPM Settings

3.1 View / Modify Messaging Interface Configuration

For viewing / modifying the configuration for BOX FileAct the access rights in the UPM must be set accordingly. The configuration is done on Enterprise level.

Select **Administration / UPM** from the main menu.

Select the user (or node or role) for which the access rights shall be specified.

From the tree view, select **Access Rights / Admin Access**.

Set the following access right(s) **View Server Config**, (view only, can be set for both Enterprise and Company level objects), **Modify Server Config**. (view and modify, can be set only for Enterprise level objects) to **YES**.

3.2 Messaging Interface Access

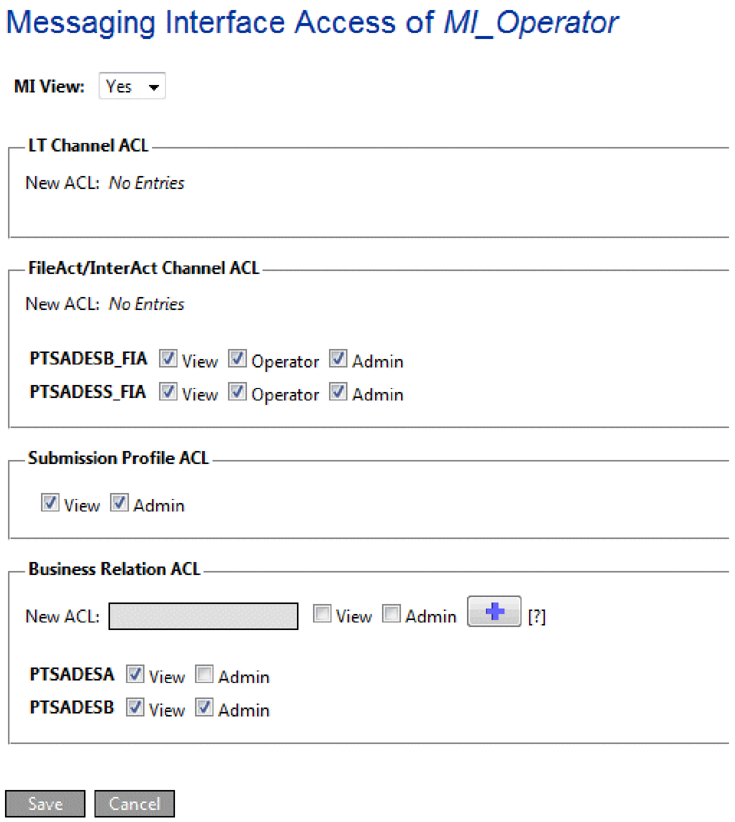
Pre-requisite for accessing, administering BOX FileAct and for working with it is that the access control in the UPM is correctly set for the logged-in user. This must be done by a user on Enterprise level.

To specify the required access for using FileAct, select **Administration / UPM** from the main menu. Select the user (or node or role) for which the access rights shall be specified.

From the three view, select **Access Rights / Messaging Interface Access**.

The **Messaging Interface Access** page of the selected UPM object is displayed.

Press **Edit**.



Messaging Interface Access of *MI_Operator*

MI View: Yes

LT Channel ACL

New ACL: *No Entries*

FileAct/InterAct Channel ACL

New ACL: *No Entries*

PTSADESB_FIA ☑ View ☑ Operator ☑ Admin

PTSADESS_FIA ☑ View ☑ Operator ☑ Admin

Submission Profile ACL

☑ View ☑ Admin

Business Relation ACL

New ACL: ☐ View ☐ Admin [?]

PTSADESA ☑ View ☐ Admin

PTSADESB ☑ View ☑ Admin

On this page you can set additional Messaging Interface-relevant access rights. These will be described below.

The LT Session ACL section refers to access control that relates to CBT and LT Sessions and is described in the CBT documentation.

3.2.1 General Access

In the general section specify the access rights **MI View** and **MI Admin** by selecting **Yes** or **No** from the pull-down menu.

The access right **MI View** allows viewing MI Modules, MI Local Channel Groups, FINCopy Profiles and ASP Profiles. This access right can be set for Enterprise, Company, Division, Department and Group level objects.

The access right **MI Admin** allows adding, deleting and modifying MI Modules and the MI Local Channel Groups. This access right can be set for Enterprise level objects only.

3.2.2 FileAct Channel Access Control List

The **FileAct/InterAct Channel ACL** (Access control list) section represents an additional access control for FileAct channels. This access control must be specified for each FileAct channel separately:

Note: A FileAct ACL **must** be created for each FileAct Channel before the channel itself can be created. Only channels listed in the section FileAct/InterAct Channel ACL will be displayed in the **Channel Name** selection box when creating new FileAct Channels (see 4.6)!

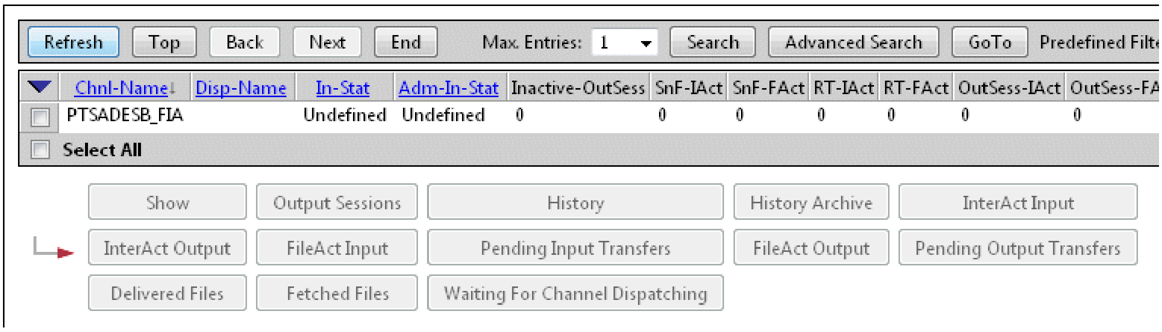
There are three FileAct Channel Access Control Lists: **View**, **Operator** and **Admin**.

Below you can find a list of enabled actions for each access control list:

View ACL (Channels):

- Show
- Output Sessions
- History
- History Archive
- InterAct Input
- InterAct Output
- FileAct Input
- Pending Input Transfers
- FileAct Output
- Pending Output Transfers
- Delivered Files
- Fetched Files
- Waiting For Channel Dispatching

FileAct/InterAct Channels



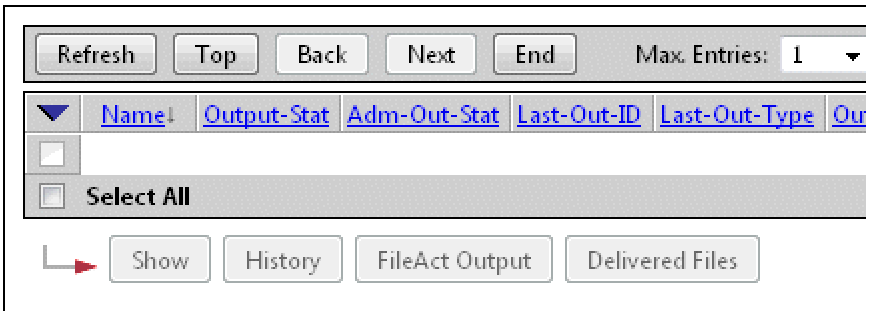
Max. Entries: 1

Chnl-Name	Disp-Name	In-Stat	Adm-In-Stat	Inactive-OutSess	SnF-IAct	SnF-FAct	RT-IAct	RT-FAct	OutSess-IAct	OutSess-FA
PTSADESB_FIA		Undefined	Undefined	0	0	0	0	0	0	0

View ACL (Output Sessions):

- Show
- History
- FileAct Output
- Delivered Files

FIA Output Sessions *(PTSADESB_FIA)*



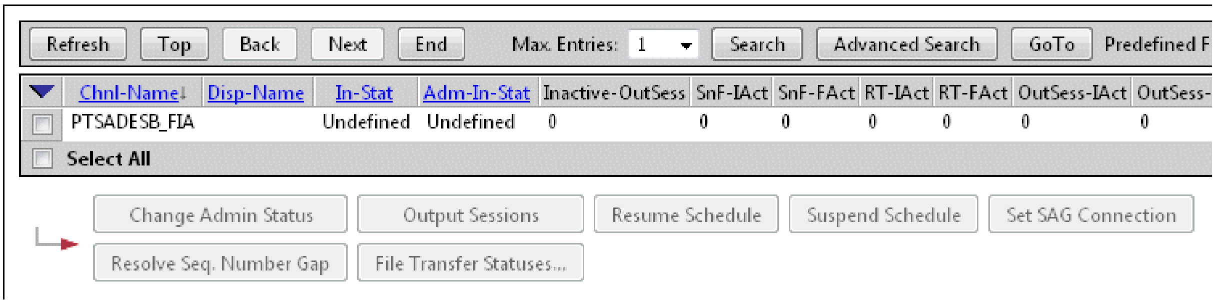
Max. Entries: 1

Name	Output-Stat	Adm-Out-Stat	Last-Out-ID	Last-Out-Type	Ou

Operator ACL (Channels):

- Change Admin Status
- Output Sessions
- Resume Schedule
- Suspend Schedule
- Set SAG Connection
- Resolve Seq. Number Gap
- File Transfer Statuses

FileAct/InterAct Channels

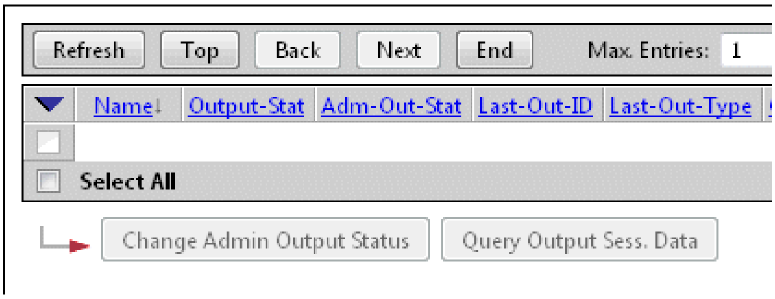


Chnl-Name	Disp-Name	In-Stat	Adm-In-Stat	Inactive-OutSess	SnF-IAct	SnF-FAct	RT-IAct	RT-FAct	OutSess-IAct	OutSess-
PTSADESB_FIA		Undefined	Undefined	0	0	0	0	0	0	0

Operator ACL (Output Sessions):

- Change Admin Output Status
- Query Output Sess. Data

FIA Output Sessions *(PTSADESB_FIA)*

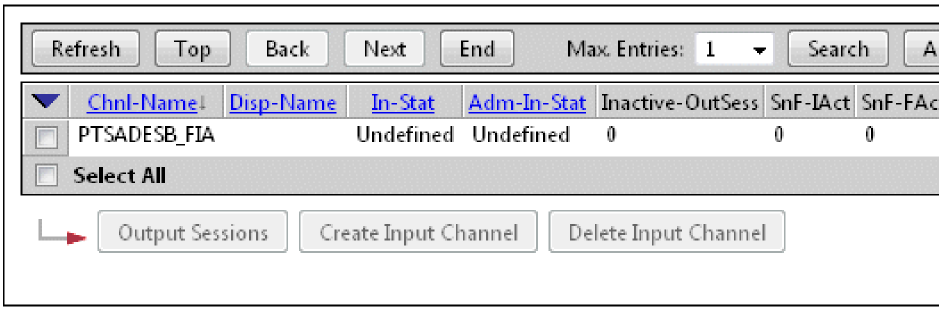


Name	Output-Stat	Adm-Out-Stat	Last-Out-ID	Last-Out-Type

Admin ACL (Channels):

- Create Input Channel
- Delete Input Channel

FileAct/InterAct Channels



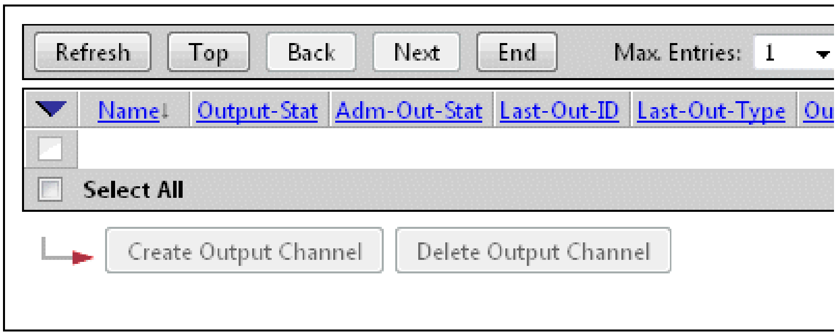
Refresh | Top | Back | Next | End | Max. Entries: 1 | Search | A

	Chnl-Name	Disp-Name	In-Stat	Adm-In-Stat	Inactive-OutSess	SnF-IAct	SnF-FAc
□	PTSADESB_FIA		Undefined	Undefined	0	0	0
□	Select All						

Admin ACL (Output Sessions):

- Create Output Channel
- Delete Output Channel

FIA Output Sessions *(PTSADESB_FIA)*



Refresh | Top | Back | Next | End | Max. Entries: 1

	Name	Output-Stat	Adm-Out-Stat	Last-Out-ID	Last-Out-Type	Ou
□						
□	Select All					

Enter the name of the FileAct channel for which access control shall be specified.

Select the access control by checking the respective box(es). At least one box must be checked.

Press the + button.

Repeat the steps for each FileAct channel for which access control shall be specified.

To delete a FileAct channel ACL from the list you must deactivate all checkboxes.

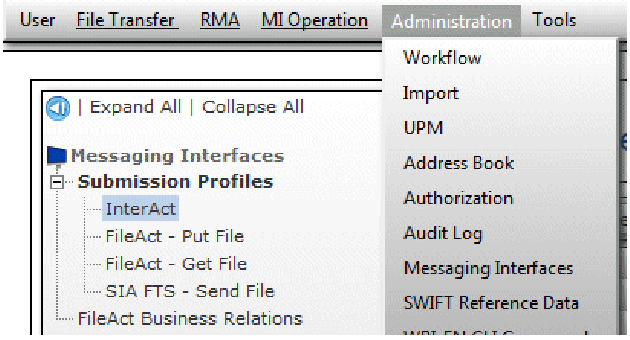
Press **Save** to have the settings become effective.

3.2.3 Submission Profile ACL

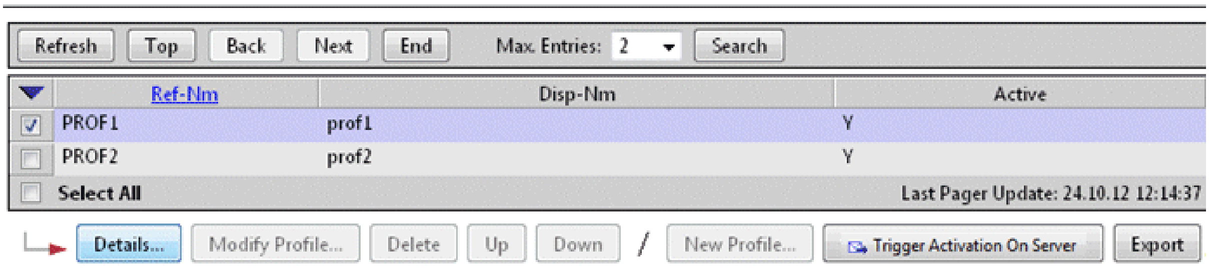
The **Submission Profile ACL** section refers to access control regarding Submission Profiles

Activating **View**:

- allows viewing the menu item **Submission Profiles** and sub-menus:



- enables the task buttons **Details…**, **Trigger Activation On Server** and **Export**:



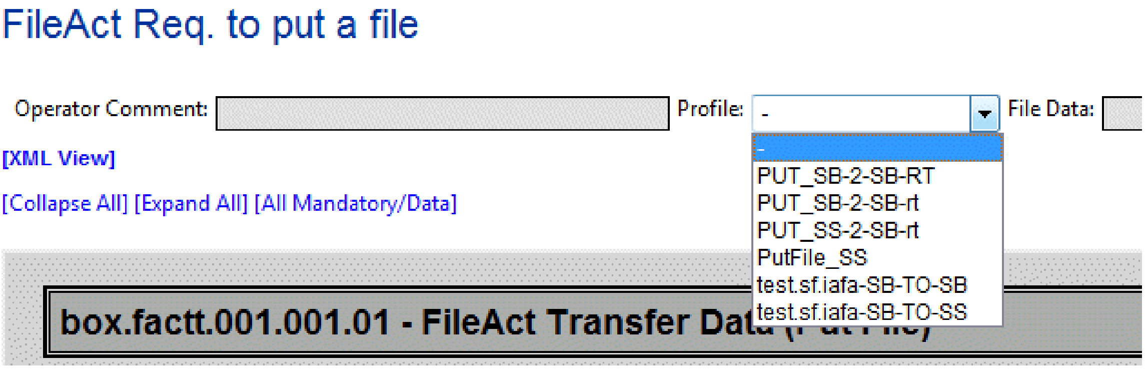
Max. Entries: 2

	Ref-Nm	Disp-Nm	Active
☑	PROF1	prof1	Y
□	PROF2	prof2	Y

□ Select All

Last Pager Update: 24.10.12 12:14:37

- allows selecting the Profile in the FileAct Message Entry form:


FileAct Req. to put a file

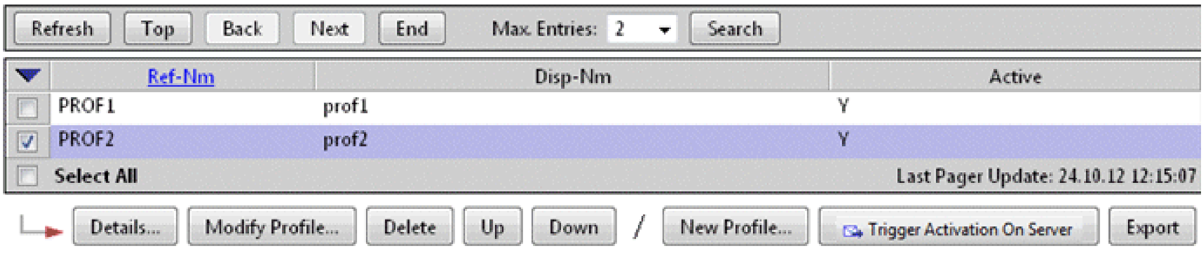
Operator Comment: Profile: - File Data:

-
PUT_SB-2-SB-RT
PUT_SB-2-SB-rt
PUT_SS-2-SB-rt
PutFile_SS
test.sf.iafa-SB-TO-SB
test.sf.iafa-SB-TO-SS

[XML View]
[Collapse All] [Expand All] [All Mandatory/Data]
box.factt.001.001.01 - FileAct Transfer Data

Activating **Admin**:

- additionally enables the task buttons **Modify Profile…**, **Delete**, **Up**, **Down** and **New Profile**:



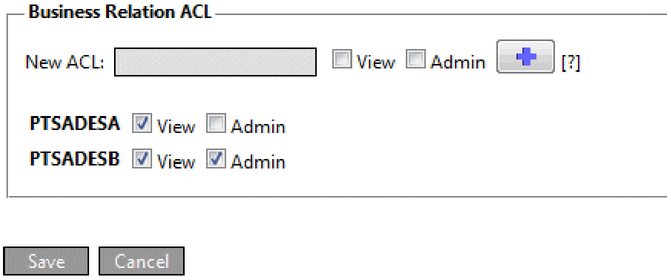
	Ref-Nm	Disp-Nm	Active
□	PROF1	prof1	Y
☑	PROF2	prof2	Y
□	Select All		Last Pager Update: 24.10.12 12:15:07

3.2.4 Business Relation ACL

Business Relations allow specifying criteria for accepting or rejecting received FileAct messages in order to control which correspondents are allowed to send FileAct messages to your organization / company.

A Business Relation is configured separately for each Responder BIC (= Own BIC).

If a user shall be able to access (view or create/modify/delete) a FileAct Business Relation for a specific Responder BIC, he must have a respective entry in the **Business Relation ACL** (Business Relation Access List) configured in the UPM.



Business Relation ACL

New ACL: □ View □ Admin + [?]

PTSADESA ☑ View □ Admin

PTSADESB ☑ View ☑ Admin

Specify the Responder BIC8 for which you want to create an entry into the **New ACL** field.

You can assign **View** ACL (view only) or **Admin** ACL (create/modify/delete).

Press the + button.

The BIC 8 is displayed in the list.

If additional ACLs are required for other Responder BICs, repeat the procedure for each Responder BIC.

Press **Save**.

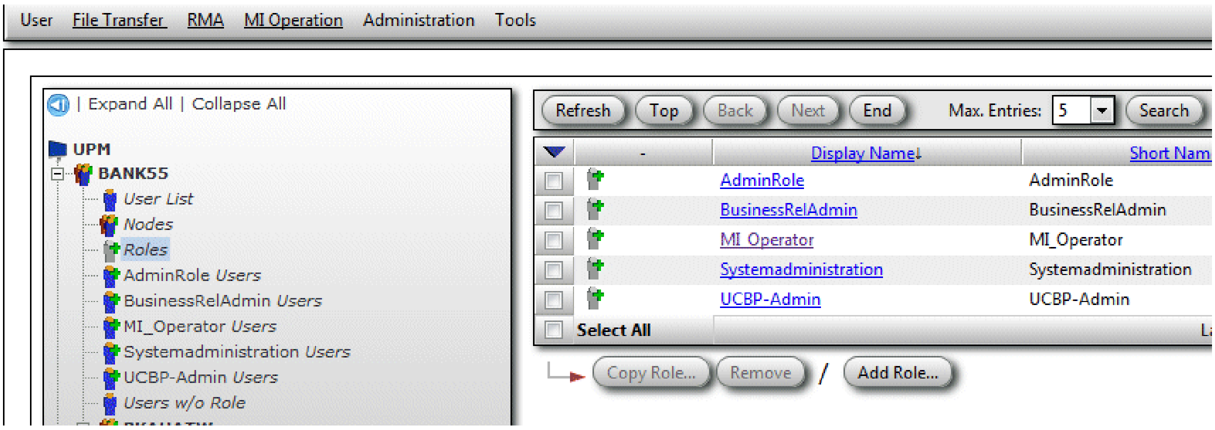
How to create Business Relation Sets is described in section 4.12 of this document.

3.3 UPM Settings for Operator Actions

Any user or application that wants to handle FileAct messages must have the corresponding access rights specified in the BOX User Profile Management (UPM). The Access rights can be specified either for UPM Users, for UPM Nodes or for UPM Roles.

Select **Administration / UPM /** from the main menu.

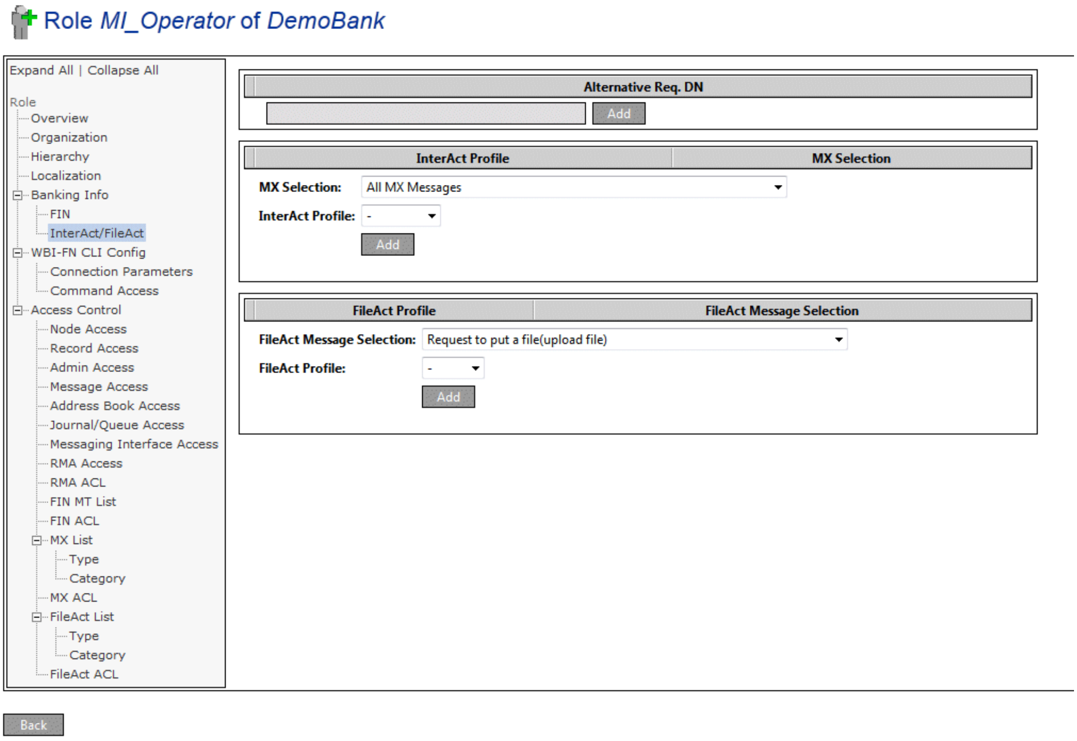
Then select the UPM object (user, role, node) for which the access rights shall be set.



	-	Display Name↓	Short Nam
□		AdminRole	AdminRole
□		BusinessRelAdmin	BusinessRelAdmin
□		MI Operator	MI_Operator
□		Systemadministration	Systemadministration
□		UCBP-Admin	UCBP-Admin
□	Select All		L

3.3.1 Banking Info

Go to **Banking Info / InterAct/FileAct** in the tree view on the left hand side.


Role *MI_Operator* of DemoBank

MX Selection: All MX Messages

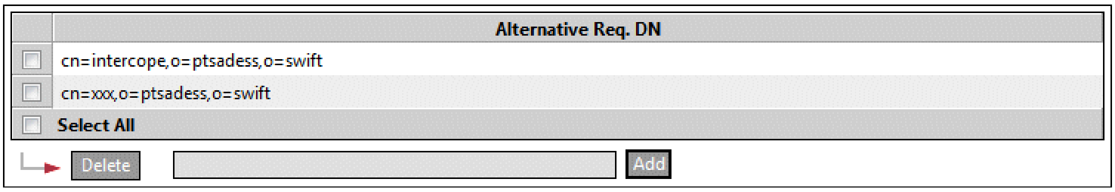
InterAct Profile: -

FileAct Message Selection: Request to put a file(upload file)

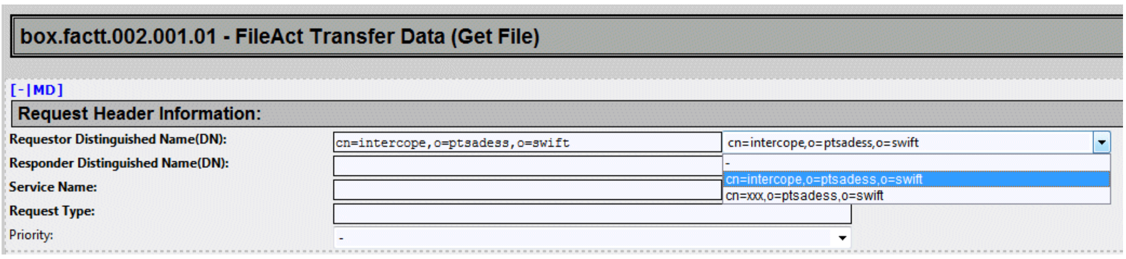
FileAct Profile: -

On this page you specify which FileAct Profiles the user (attached to this UPM Node or with this UPM Role) can access.

In the **Alternative Req. DN** field you can specify alternative Requestor DNs which then can be selected to be used as Requestor DN when entering the **Request Header Information** in (message) forms (see figures below).



	Alternative Req. DN
□	cn=intercope,o=ptsadess,o=swift
□	cn=xxx,o=ptsadess,o=swift
□	Select All



box.factt.002.001.01 - FileAct Transfer Data (Get File)

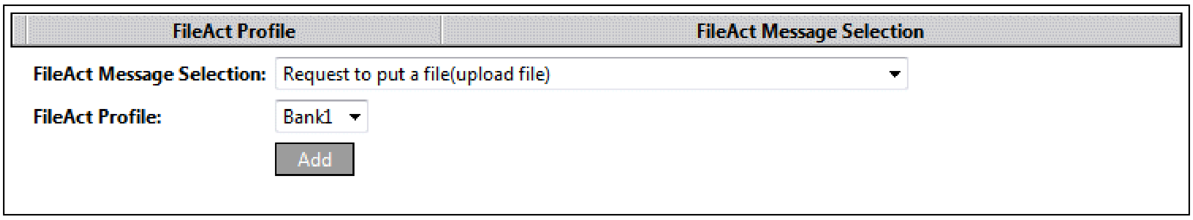
Request Header Information:

Requestor Distinguished Name(DN):	cn=intercope,o=ptsadess,o=swift	cn=intercope,o=ptsadess,o=swift
		-
		cn=intercope,o=ptsadess,o=swift
		cn=xxx,o=ptsadess,o=swift

Responder Distinguished Name(DN):		
Service Name:		
Request Type:		
Priority:	-	

To specify a Profile that the user may access, select the **FileAct Profile** from the list and press **Add**.

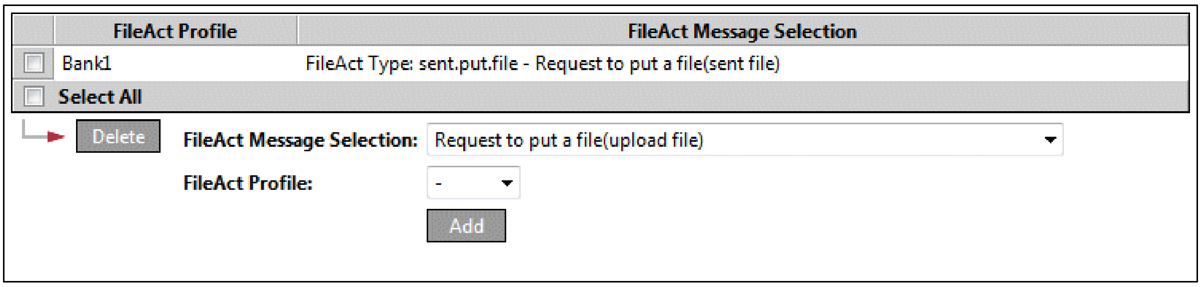
The list of entries is filtered by the **FileAct Message Selection** above, i.e. if you select the FileAct Message Selection “Request to put a file (file upload)”, the list of **FileAct Profiles** will show only the Submission Profiles defined in **Administration / Messaging Interfaces /Submission Profiles / FileAct – Put File**.



FileAct Profile	FileAct Message Selection

FileAct Message Selection: Request to put a file(upload file)

FileAct Profile: Bank1



	FileAct Profile	FileAct Message Selection
□	Bank1	FileAct Type: sent.put.file - Request to put a file(sent file)
□	Select All	

FileAct Message Selection: Request to put a file(upload file)

FileAct Profile: -

Select additional FileAct Profiles if required and add them to the list of FileAct Profiles that shall be accessible for the user (attached to this UPM Node or with this UPM Role).

3.3.2 FileAct List

On this page you specify which FileAct messages the user (attached to this UPM Node or with this UPM Role) can access.

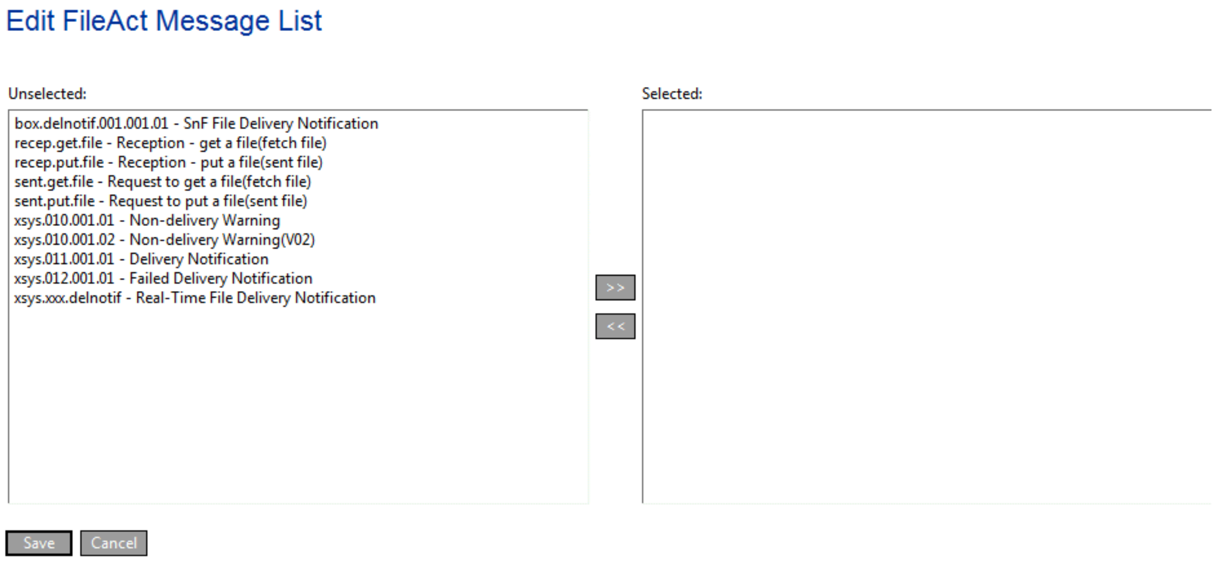
The access rights can be granted either separately for each type of message or by category of messages.

Select **Administration / UPM** from the main menu.

Select the UPM object (user, role, node) for which the access rights shall be set.

To grant access for types of FileAct messages, go to **FileTransfer List / Type** in the tree view on the left hand side.

Press **Edit**.



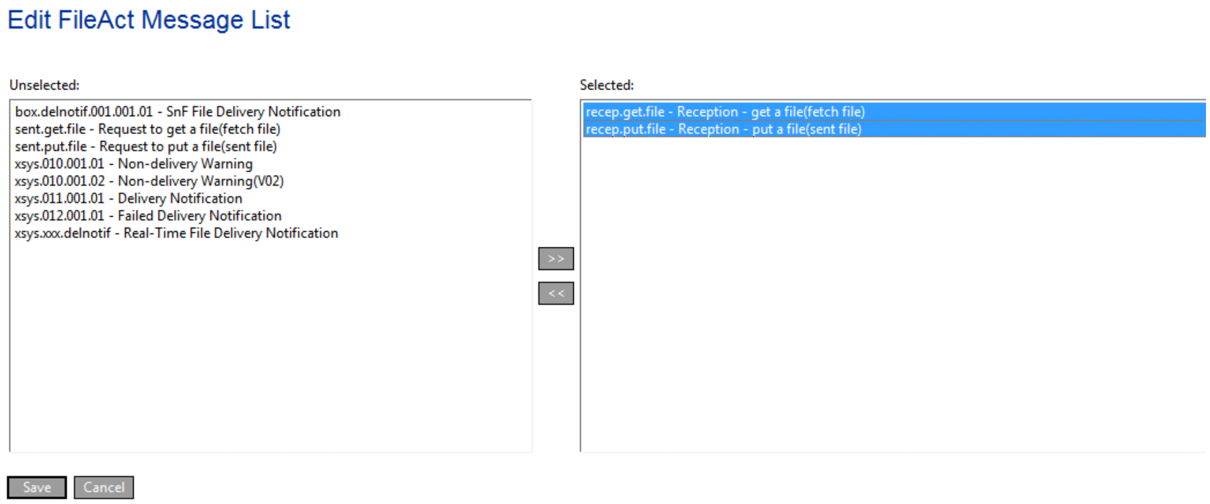
Edit FileAct Message List

Unselected:

box.delnotif.001.001.01 - SnF File Delivery Notification
recep.get.file - Reception - get a file(fetch file)
recep.put.file - Reception - put a file(sent file)
sent.get.file - Request to get a file(fetch file)
sent.put.file - Request to put a file(sent file)
xsys.010.001.01 - Non-delivery Warning
xsys.010.001.02 - Non-delivery Warning(V02)
xsys.011.001.01 - Delivery Notification
xsys.012.001.01 - Failed Delivery Notification
xsys.xxx.delnotif - Real-Time File Delivery Notification

Selected:

Select the desired type of messages in the **Unselected** section on the left hand side and move them to the **Selected** section on the right hand side by pressing the **>>** button.



Edit FileAct Message List

Unselected:

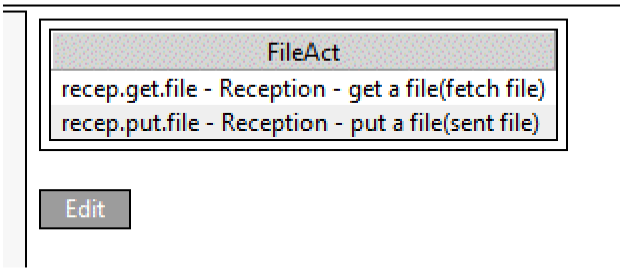
box.delnotif.001.001.01 - SnF File Delivery Notification
sent.get.file - Request to get a file(fetch file)
sent.put.file - Request to put a file(sent file)
xsys.010.001.01 - Non-delivery Warning
xsys.010.001.02 - Non-delivery Warning(V02)
xsys.011.001.01 - Delivery Notification
xsys.012.001.01 - Failed Delivery Notification
xsys.xxx.delnotif - Real-Time File Delivery Notification

Selected:

recep.get.file - Reception - get a file(fetch file)
recep.put.file - Reception - put a file(sent file)

Press **Save**.

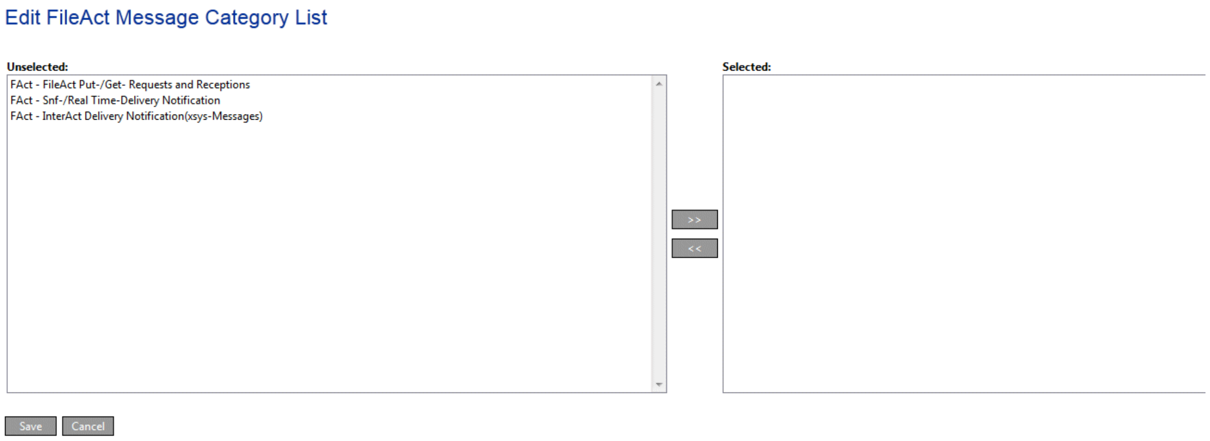
The selected types of messages now appear in the list.



FileAct
recep.get.file - Reception - get a file(fetch file)
recep.put.file - Reception - put a file(sent file)

To grant access for categories of FileAct messages, go to **FileAct List / Category** in the tree view on the left hand side.

Press **Edit**.



Edit FileAct Message Category List

Unselected:

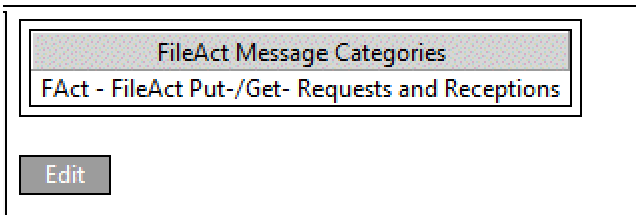
FAct - FileAct Put-/Get- Requests and Receptions
FAct - Snf-/Real Time-Delivery Notification
FAct - InterAct Delivery Notification(xsys-Messages)

Selected:

Select the desired category of messages in the **Unselected** section on the left hand side and move them to the **Selected** section on the right hand side by pressing the **>>** button.

Press **Save**.

The selected categories of messages now appear in the list.



FileAct Message Categories
FAct - FileAct Put-/Get- Requests and Receptions

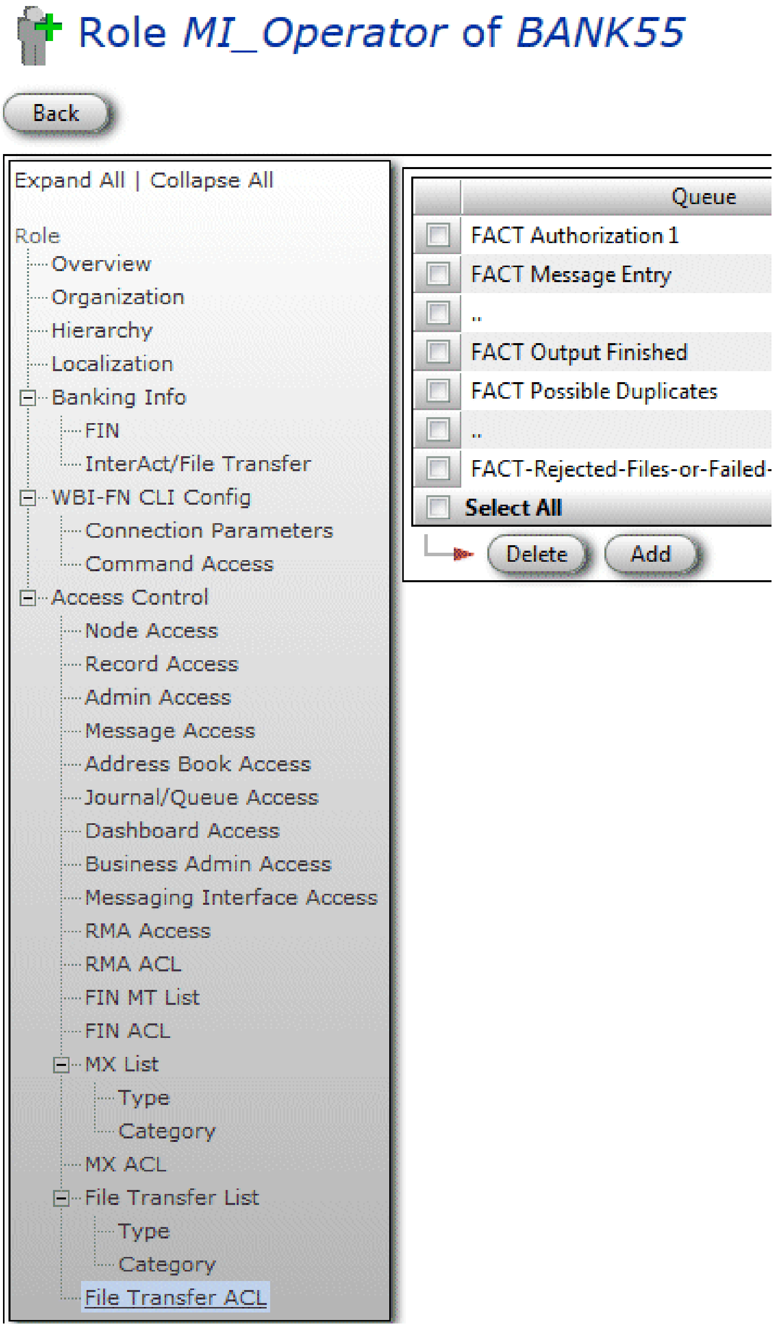
3.3.3 FileAct ACL

On this page you specify which **Tasks** in the BOX Application Queues a user (attached to this UPM Node or with this UPM Role) can access.

The access on application queue task level is checked only if the parameter **ACL-Check** has been activated for respective task.

If this parameter has not been activated, the access is granted as specified in FileAct List (see above, 3.3.2).

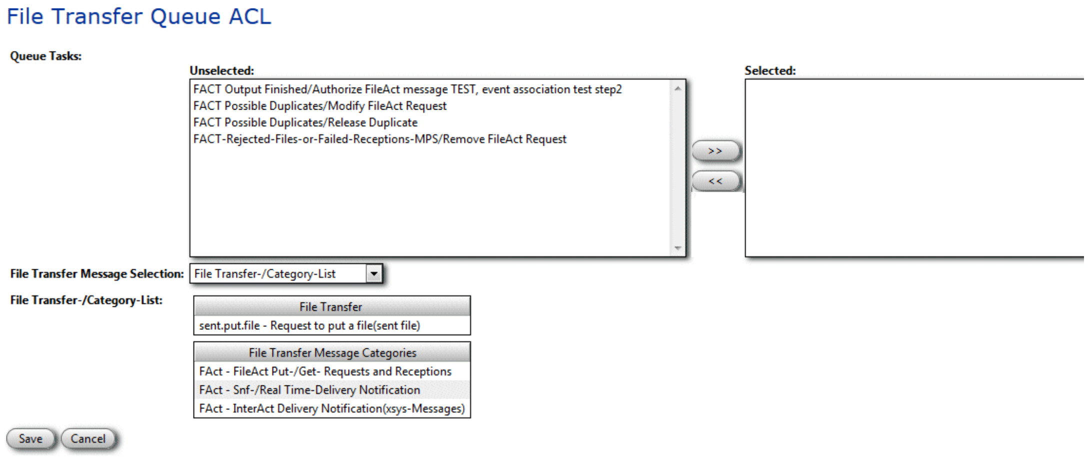
To specify a **Task** select **FileAct ACL** in the tree view on the left hand side:



Role *MI_Operator of BANK55*

	Queue
□	FACT Authorization 1
□	FACT Message Entry
□	..
□	FACT Output Finished
□	FACT Possible Duplicates
□	..
□	FACT-Rejected-Files-or-Failed-
□	**Select All**

Press the **Add** button.



File Transfer Queue ACL

Queue Tasks:

Unselected:

FACT Output Finished/Authorize FileAct message TEST, event association test step2
FACT Possible Duplicates/Modify FileAct Request
FACT Possible Duplicates/Release Duplicate
FACT-Rejected-Files-or-Failed-Receptions-MPS/Remove FileAct Request

Selected:

File Transfer Message Selection: File Transfer-/Category-List

File Transfer-/Category-List:

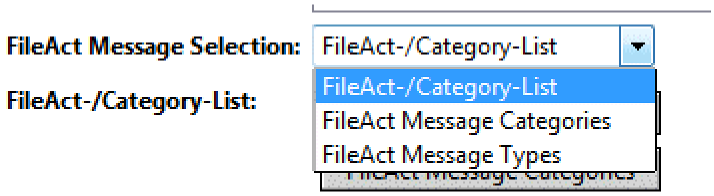
File Transfer
sent.put.file - Request to put a file(sent file)

File Transfer Message Categories
FAct - FileAct Put-/Get- Requests and Receptions
FAct - Snf-/Real Time-Delivery Notification
FAct - InterAct Delivery Notification(xsys-Messages)

The **Queue Tasks:** parameter shows the name of the application queue (**FACT Message Entry**) and the Task (button) in this queue.

To allow the user to perform this task in this queue select the task (e.g. **FACT Message Entry/FileAct Req. to fetch a file**) and move it into the **Selected** field on the right hand side using the **>>** button.

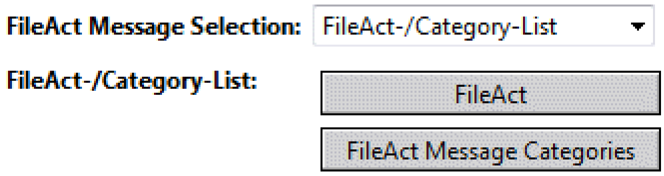
The **FileAct Message Selection:** pull down menu filters the view below:



FileAct Message Selection: FileAct-/Category-List

FileAct-/Category-List:

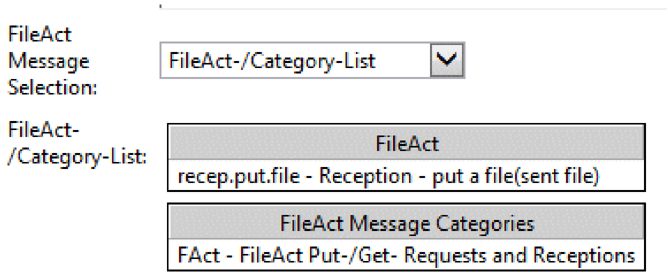
If you select **FileAct- / Category-List**, a list of FileAct messages and FileAct Categories is shown. This list shows the Types and Categories specified in **FileAct List / Type** and **FileAct List / Category** (see above, 3.3.2). The following picture shows a situation where no **FileAct List / Type** and no **FileAct List / Category** has been specified.



FileAct Message Selection: FileAct-/Category-List

FileAct-/Category-List:

The picture below shows a situation where one **Type** (recep.put.file – Reception – put a file (sent file) and one **Category** (FacT – FileAct Put-/Get- Requests and Receptions) have been specified:



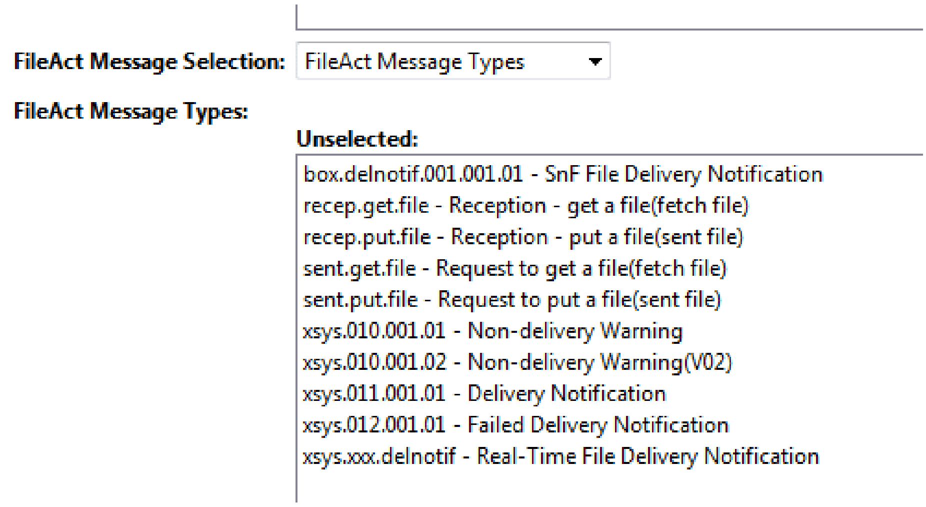
FileAct Message Selection: FileAct-/Category-List

FileAct-/Category-List:

FileAct
recep.put.file - Reception - put a file(sent file)

FileAct Message Categories
FAct - FileAct Put-/Get- Requests and Receptions

The **FileAct Message Types** option allows selecting message types (by moving selected types from the **Unselected** field into the **Selected** field using the **>>** button).



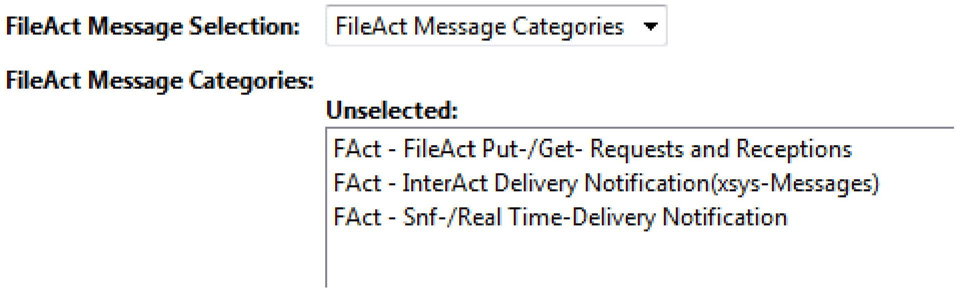
FileAct Message Selection: FileAct Message Types ▾

FileAct Message Types:

Unselected:

box.delnotif.001.001.01 - SnF File Delivery Notification
recep.get.file - Reception - get a file(fetch file)
recep.put.file - Reception - put a file(sent file)
sent.get.file - Request to get a file(fetch file)
sent.put.file - Request to put a file(sent file)
xsys.010.001.01 - Non-delivery Warning
xsys.010.001.02 - Non-delivery Warning(V02)
xsys.011.001.01 - Delivery Notification
xsys.012.001.01 - Failed Delivery Notification
xsys.xxx.delnotif - Real-Time File Delivery Notification

The **FileAct Message Categories** option allows selecting message categories (by moving selected types from the **Unselected** field into the **Selected** field using the **>>** button).



FileAct Message Selection: FileAct Message Categories ▾

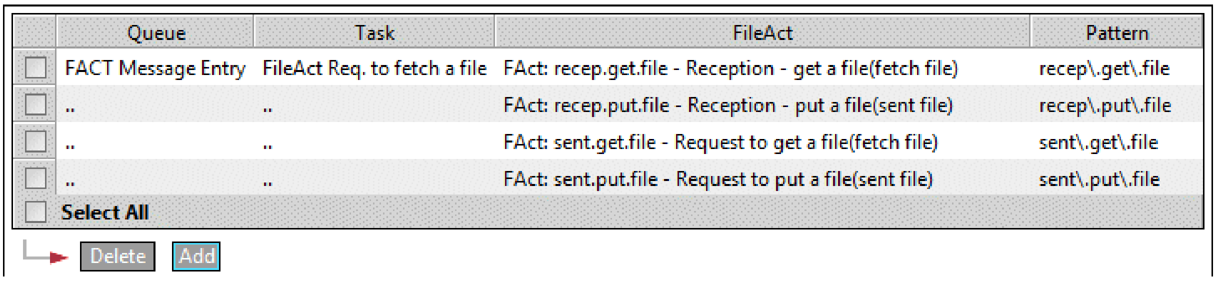
FileAct Message Categories:

Unselected:

FAct - FileAct Put-/Get- Requests and Receptions
FAct - InterAct Delivery Notification(xsys-Messages)
FAct - Snf-/Real Time-Delivery Notification

Select the Queue Tasks and the FileAct messages using the above described options and press **Save**.

The FileAct ACL page is displayed:



	Queue	Task	FileAct	Pattern
□	FACT Message Entry	FileAct Req. to fetch a file	FAct: recep.get.file - Reception - get a file(fetch file)	recep\.get\.file
□	..	..	FAct: recep.put.file - Reception - put a file(sent file)	recep\.put\.file
□	..	..	FAct: sent.get.file - Request to get a file(fetch file)	sent\.get\.file
□	..	..	FAct: sent.put.file - Request to put a file(sent file)	sent\.put\.file
□	Select All			

The specifications above would describe the following situation:

In the **FACT Message Entry** queue the user may create a **FileAct Request to fetch a file** message for the message types:

- recep.get.file - Reception - get a file(fetch file)
- recep.put.file - Reception - put a file(sent file)
- sent.get.file - Request to get a file(fetch file)
- sent.put.file - Request to put a file(sent file)

4 The BOX Messaging Interface

4.1 Architecture

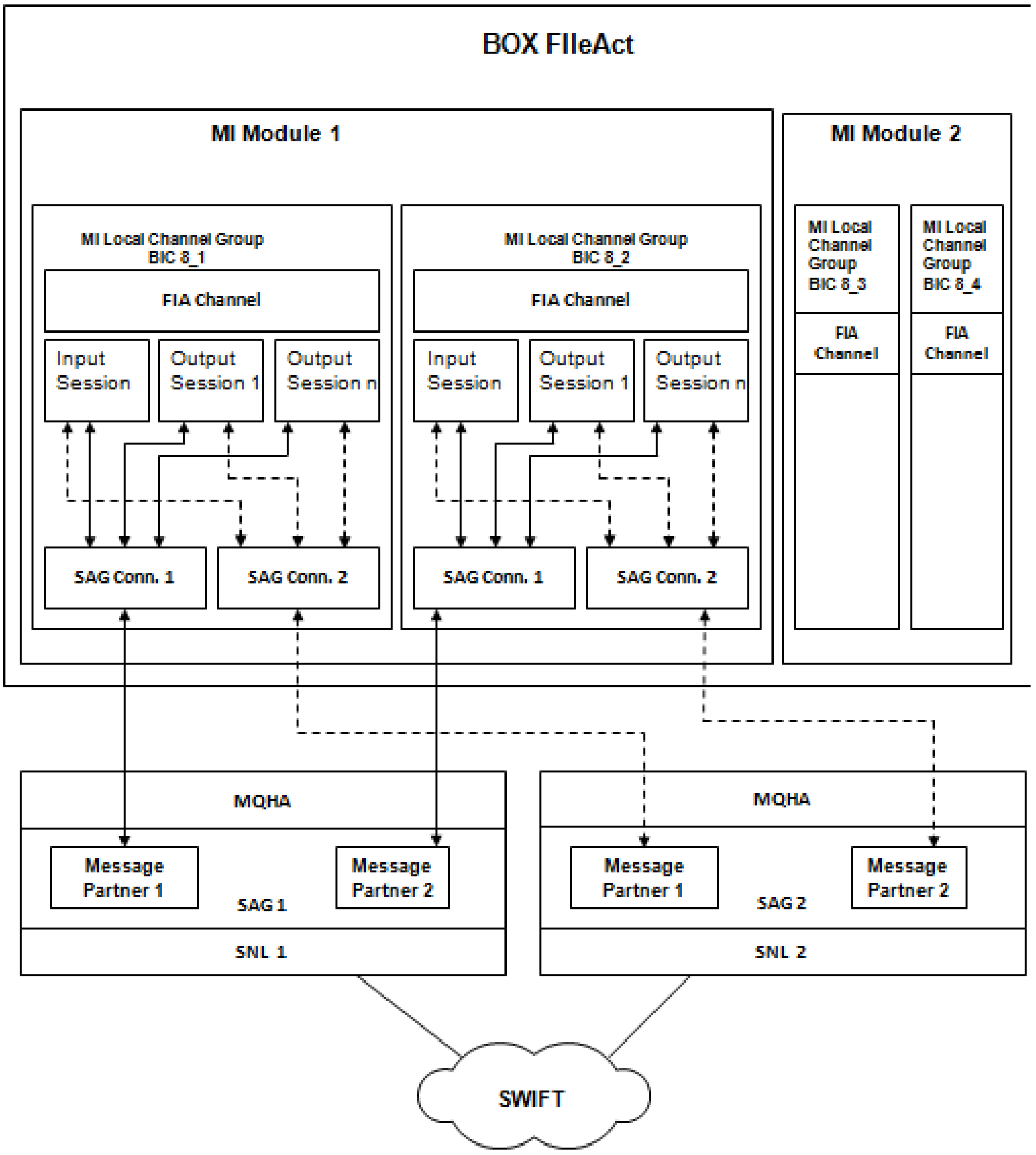
BOX FileAct allows sending files to and receiving files from SWIFTNet via a SWIFT Alliance Gateway, without requiring any additional SWIFT connectivity solution.

The BOX FileAct functionality uses the BOX Messaging Interface that connects BOX FileAct channels to SAG via MQ.

There may be multiple Messaging Interface Modules.

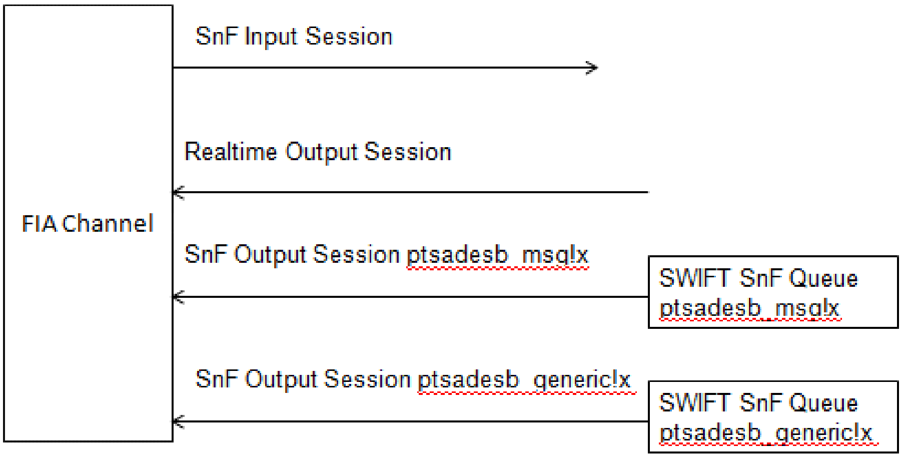
Generally seen:

- Each Messaging Interface Module (MI Module) can handle one or more FileAct Channels.
- The channels belonging to an MI Module are organized in Messaging Interface Local Channel Groups (MI LCG), each of them including one (or more) FileAct Channel(s).
- Each FileAct channel is connected via MQ to one SAG Message Partner.



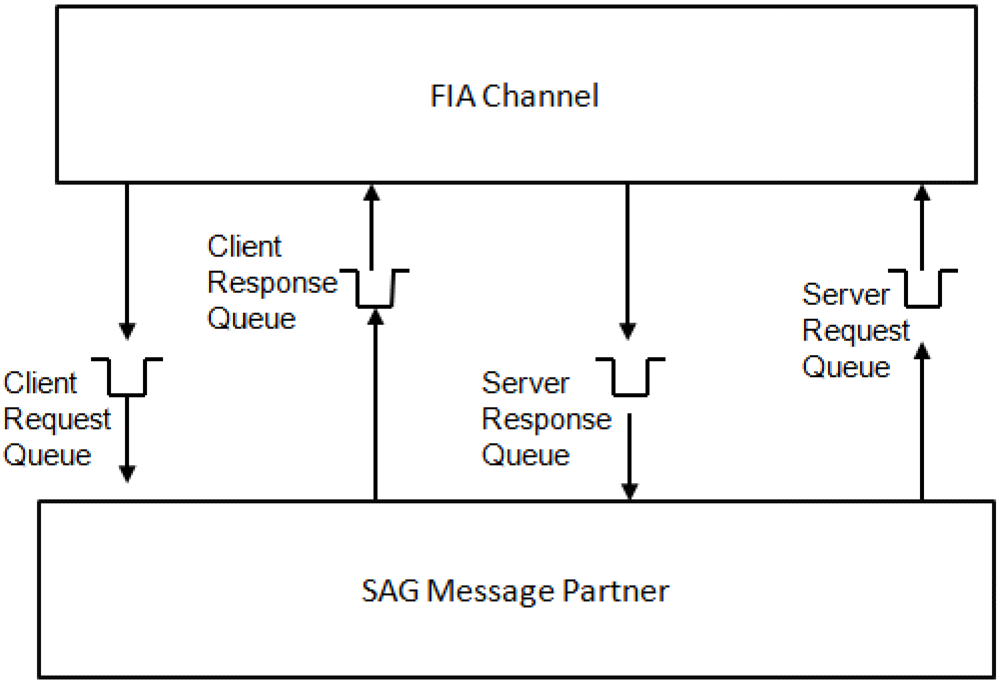
Furthermore (as can be seen in the figure above)

- One LCG handles the traffic for a single BIC8 only.
- One FIA Channel establishes one Input Session and one or more Output Sessions.
- One FIA Channel usually makes use of one SAG Connection and thus connects to one Message Partner. However, alternative connections can be configured e.g. in order to account for possible connection failures.
- A FIA channel consists of an input session and zero or more output sessions.

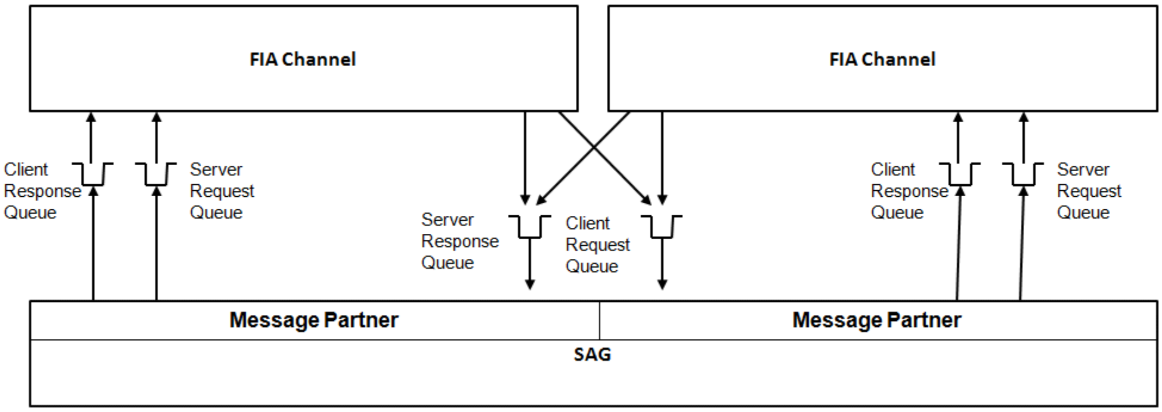

FIA Channel
SnF Input Session
Realtime Output Session
SnF Output Session ptsadesb_msg!x
SWIFT SnF Queue ptsadesb_msg!x
SnF Output Session ptsadesb_generic!x
SWIFT SnF Queue ptsadesb_generic!x

Input and output sessions may be started/stopped independently. The input session handles both realtime and SnF input traffic.
Dedicated output sessions have to be defined for realtime output traffic and each SnF-queue to be received from.

The communication between BOX FileAct and SAG requires 4 (four) queues per one FileAct channel on BOX side:



If the BOX FileAct MI Module handles more than one FileAct channel, the Client Request Queue and the Server Response Queue can be shared by multiple FIA channels.



4.1.1 File Transfer Mechanisms

In order to send files, they first have to be transferred to and stored in SAG.

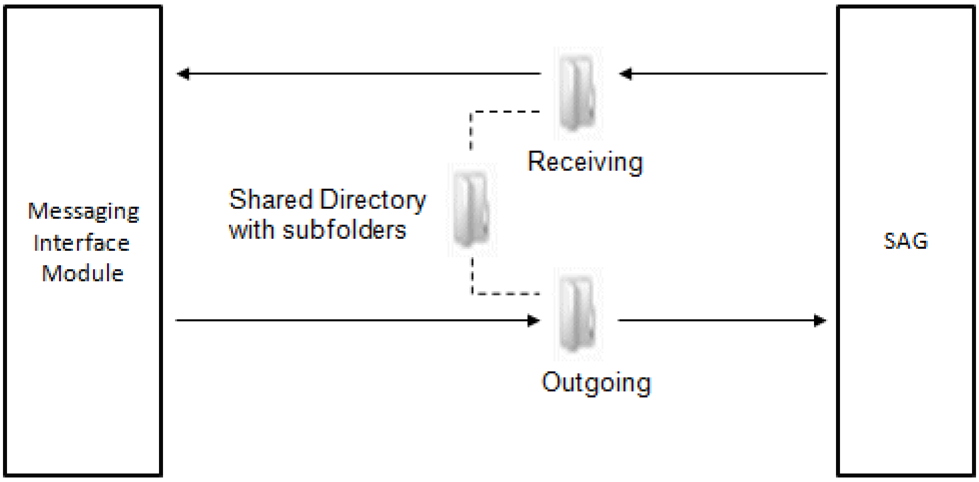
There are three different ways of transferring files to SAG:

- Directory Sharing
- Using SWIFT “Remote File Handler” (only available on AIX, Solaris and Windows operating systems)
- Using Local File Transfer via MQ (LFT).

The mechanism to be used is specified in the connection configuration. Please refer to section 8.2 in this document.

4.1.1.1 Directory Sharing

This file transfer mechanism is based on a shared directory, which is accessed by both the MI Module and SAG:



The name and the location of the shared directory is configured with the BOX parameters

FACT_TRANSFER_ROOTDIR (MI Module point of view) and

FACT_TRANSFER_SAGDIR (SAG point of view).

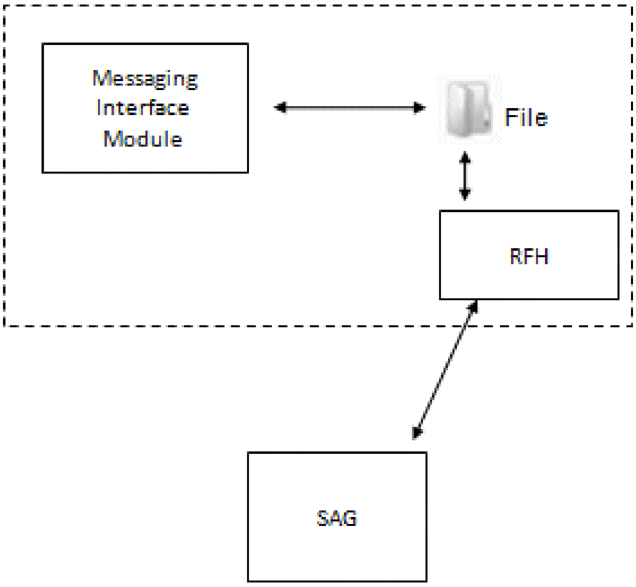
Both parameters can be found in section **[<FIA-CHANNELNAME>.CONNXX]**.

The parameter values (may) vary because the path differs, depending on from which point of view (MI Module or SAG) the path to the directory is specified.

Note that both the MI Module and SAG must have access to the directory.

4.1.1.2 SWIFT Remote File Handler

This file transfer mechanism is based on the “Remote File Handler”, which is part of the **SWIFT Remote API**. Details on how to install and use the Remote API can be found in the SWIFT documentation.

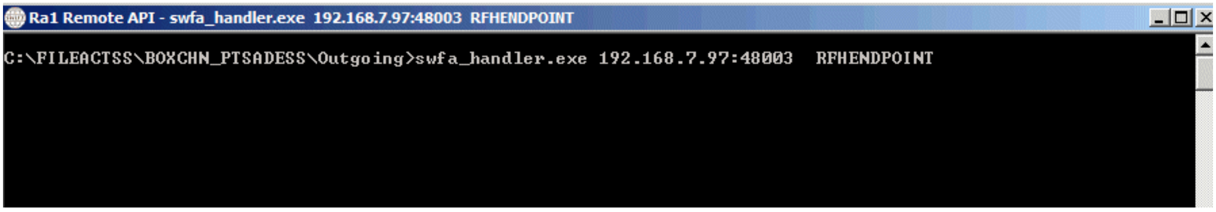


The BOX configuration parameter **RFH_ENDPOINT** in section **[<FIA-CHANNELNAME>.CONNXX]** must be set to ‘RFHENDPOINT’

The parameters **FACT_TRANSFER_ROOTDIR** and **FACT_TRANSFER_SAGDIR** in section **[<FIA-CHANNELNAME>.CONNXX]** specify the name and the path to the directory in which the files are put. The values must be identical as the MI Module and the ‘Remote File Handler’ (RFH) reside on the same machine.

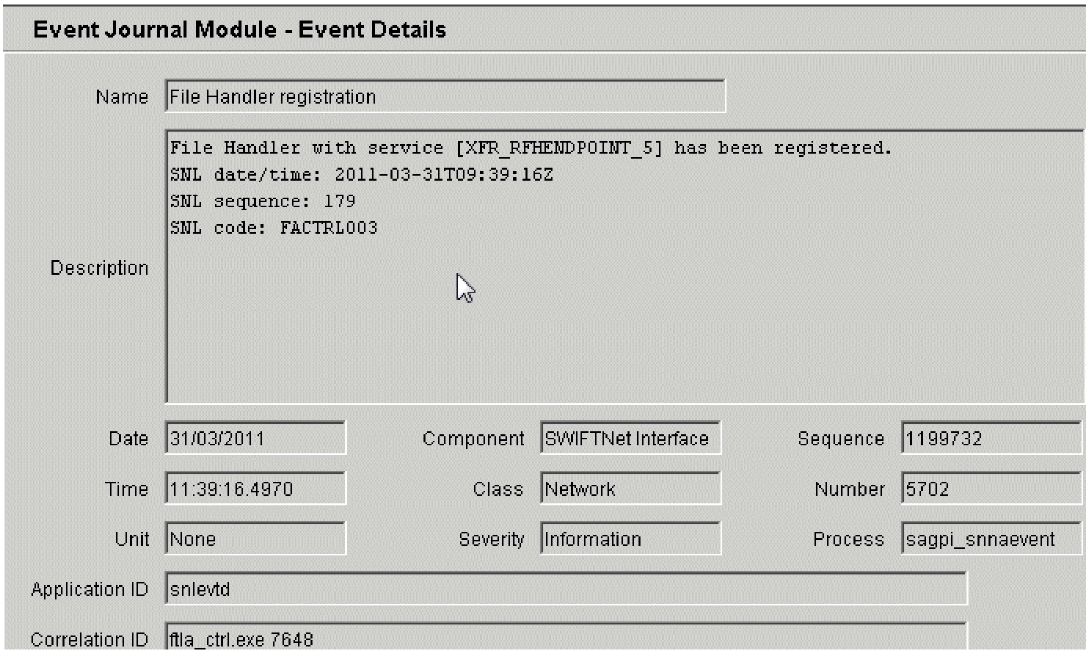
Note that both the MI Module and the RFH must have access to the directory.

To start the RFH under Windows, use the **RA1 Remote API** command shell delivered with the Remote API and enter the IP address and Port of SAG and the value of the BOX parameter **RFH_ENDPOINT** in section **[<FIA-CHANNELNAME>.CONNXX]**,
e.g. ..\RA\bin>swfa_handler.exe 192.168.7.97:48003 **RFHENDPOINT**:


Ra1 Remote API - swfa_handler.exe 192.168.7.97:48003 RFHENDPOINT

```
C:\FILEACTSS\BOXCHN_PTSADESS\Outgoing>swfa_handler.exe 192.168.7.97:48003   RFHENDPOINT
```

If the RFH was successfully started, the registration information can be seen in the SAG Event Journal:



Event Journal Module - Event Details

Name	File Handler registration

Description:

```
File Handler with service [XFR_RFHENDPOINT_5] has been registered.
SNL date/time: 2011-03-31T09:39:16Z
SNL sequence: 179
SNL code: FACTRL003
```

Date	31/03/2011	Component	SWIFTNet Interface	Sequence	1199732
Time	11:39:16.4970	Class	Network	Number	5702
Unit	None	Severity	Information	Process	sagpi_snnaevent
Application ID	snlevtd				
Correlation ID	ftla_ctrl.exe 7648				

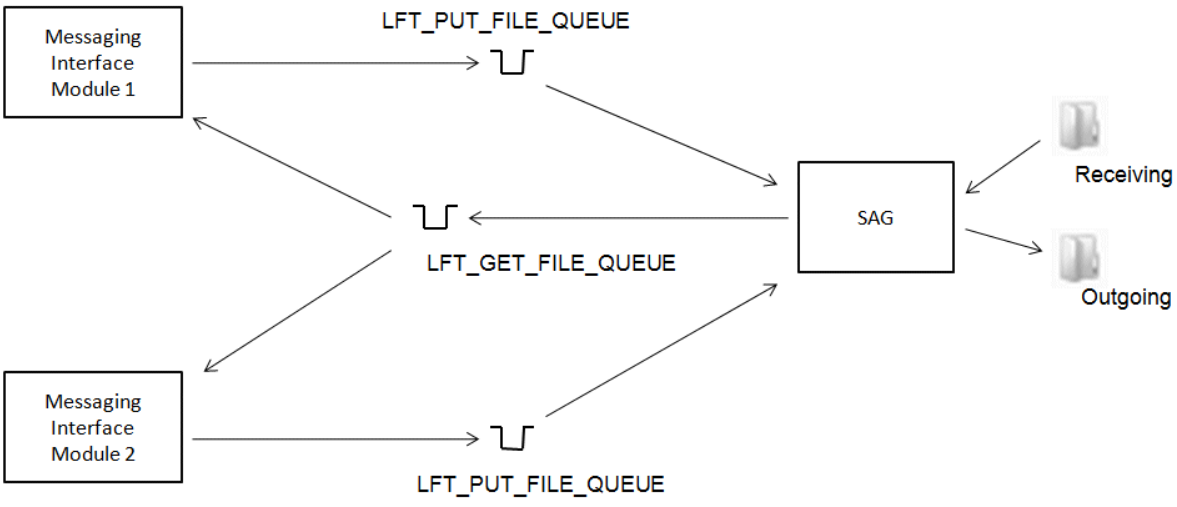
Under AIX or Solaris the SWIFTNet Link environment must be initialized before starting the RFH:

```
mpo@DEMOAIX /opt/RAHA6/bin >. ./swiftnet init
SWIFTNet Link environment initialized for instance Ra1
mpo@DEMOAIX /opt/RAHA6/bin >./swfa_handler 192.168.7.97:48003 RFHENDPOINT
```

4.1.1.3 Via Local File Transfer (LFT)

In order to make use of the Local File Transfer mechanism the parameter **FACT_USE_LFT** in section **[<FIA-CHANNELNAME>.CONNXX]** must be set to **YES**.

The LFT mechanism is based on MQ Queues. In the figure and in the description below the queue names (LFT_PUT_FILE_QUEUE, etc.) correspond to the configuration parameter names in section **[<FIA-CHANNELNAME>.CONNXX].**



SWIFT Input files are put into an MQ queue (LFT_PUT_FILE_QUEUE) from which SAG reads and stores them temporarily into a folder (Outgoing) and then sends them after having received a Send command (via the LFT_COMMAND_QUEUE). The LFT_PUT_FILE_QUEUE name must match the System Configuration Parameter **Put File Queue** in the SAG configuration.
As the **Put File Queue** parameter is Application Interface specific, there may be multiple Put File Queues (each one representing one LFT_PUT_FILE_QUEUE) to which different MI Modules can write.

SAG stores SWIFT Output files temporarily in a folder (Receiving) and then, after having received a Get command puts them into an MQ queue (LFT_GET_FILE_QUEUE) from which the MI Module reads them. The LFT_GET_FILE_QUEUE name must match the MQ Connection Parameter **GetFileQueue** of the Application Interface Module in the SAG configuration. As the GetFileQueue is an SAG System Parameter, there may be only one LFT_GET_FILE_QUEUE. However, if multiple MQ Queue Managers are used, SAG may write to multiple **GetFileQueues** (each one representing one LFT_GET_FILE_QUEUE) as long as the names of the queues do not differ.

Note: The two folders (Outgoing, Receiving) must be created manually on the machine on which SAG is running.

The BOX parameters **FACT_TRANSFER_SAGDIR** and **FACT_TRANSFER_ROOTDIR** in section **[<FIA-CHANNELNAME>.CONNXX]** point to different directories.

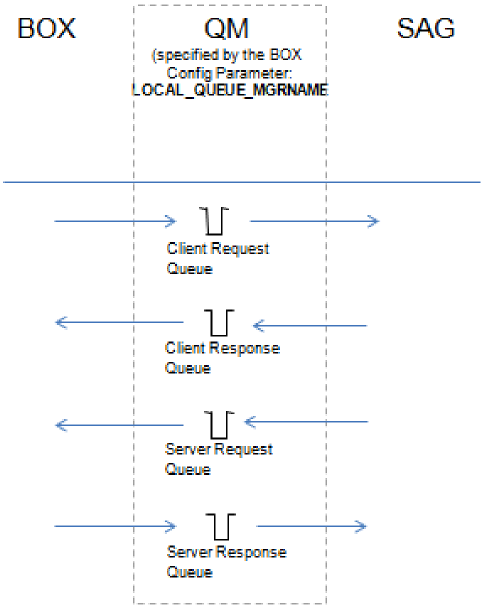
For further configuration details, see section 8.2 (SAG) in this document.

4.2 SAG Connection

4.2.1 Single Queue Manager

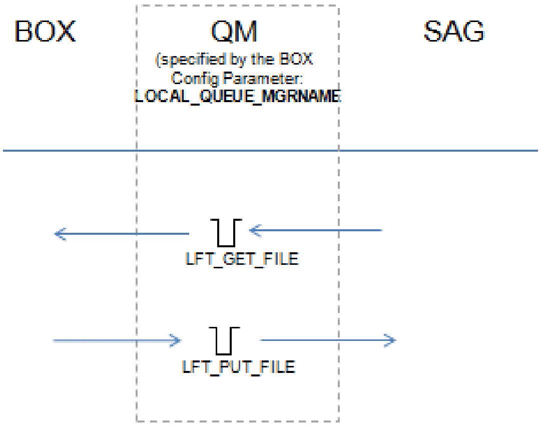
4.2.1.1 FACT Traffic

All queues (Client Request Queue, Client Response Queue, Server Response and the Server Request Queue) are located on one and the same Queue Manager (QM). The name of this Queue Manager is specified by the BOX configuration parameter LOCAL_QUEUE_MGRNAME in section **[<FIA-CHANNELNAME>.CONNXX]**.



4.2.1.2 LFT Traffic

If Local File Transfer is used, both the LFT_GET_FILE queue and the LFT_PUT_FILE queue are located on the Queue Manager specified by the BOX configuration parameter LOCAL_QUEUE_MGRNAME in section **[<FIA-CHANNELNAME>.CONNXX]**.



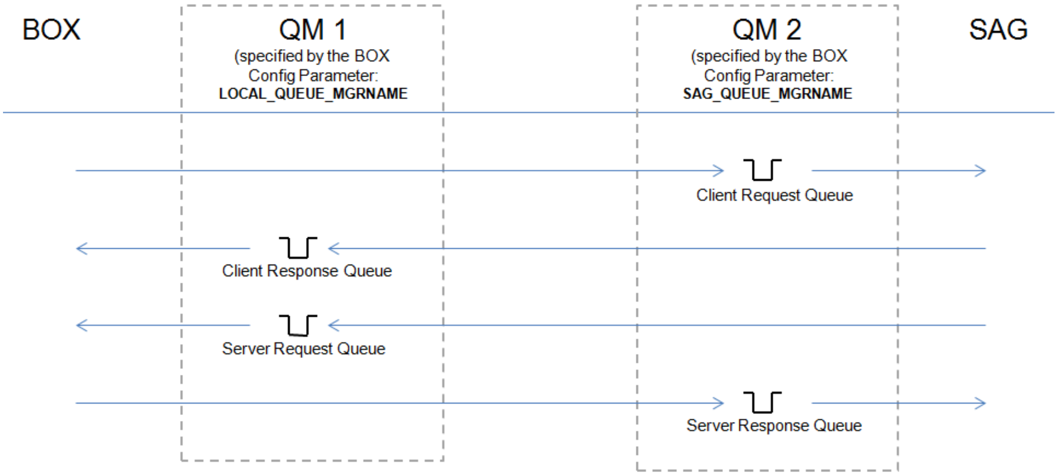
4.2.2 Multiple MQ Queue Managers

If multiple MQ Queue Managers are used, the location of the queues is of importance:

4.2.2.1 FACT Traffic

The Client Response Queue and the Server Request Queue must be located on one and the same Queue Manager (QM 1). The name of this Queue Manager is specified by the BOX configuration parameter LOCAL_QUEUE_MGRNAME in section **[<FIA-CHANNELNAME>.CONNXX]**.

The Client Request Queue and the Server Response Queue must be located on the other Queue manager (QM 2).). The name of this Queue Manager is specified by the BOX configuration parameter SAG_QUEUE_MGRNAME in section **[<FIA-CHANNELNAME>.CONNXX]**.

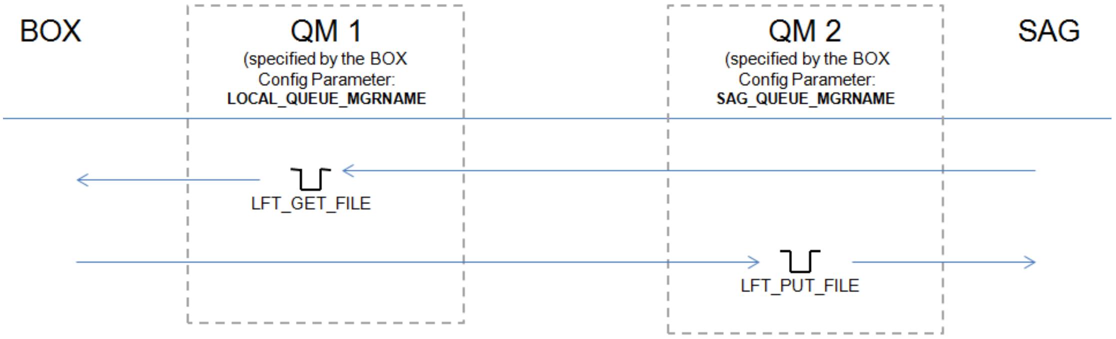


4.2.2.2 LFT Traffic

If Local File Transfer is used, the LFT_GET_FILE queue must be located on the Queue Manager specified by the BOX configuration parameter LOCAL_QUEUE_MGRNAME in section **[<FIA-CHANNELNAME>.CONNXX]**.

The LFT_PUT_FILE queue must be located on the Queue Manager specified by the BOX configuration parameter SAG_QUEUE_MGRNAME in section **[<FIA-CHANNELNAME>.CONNXX]**

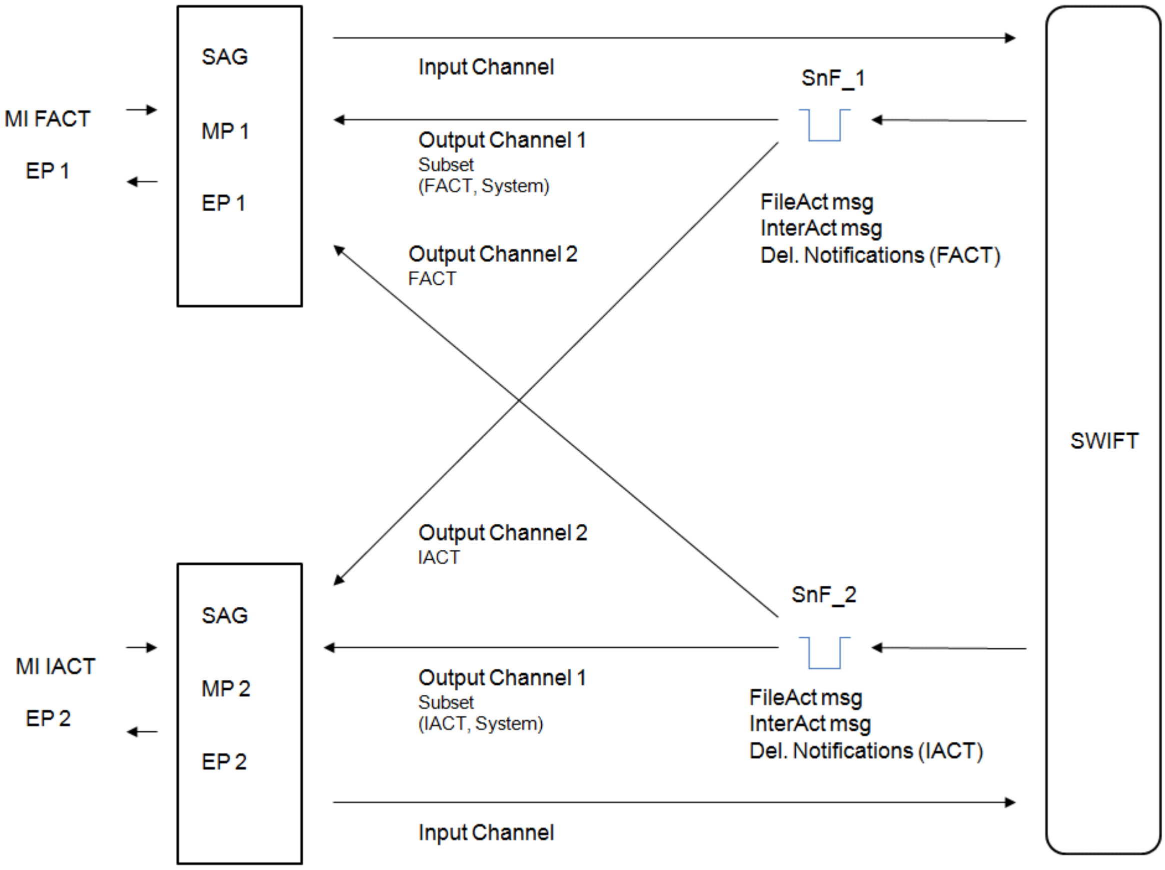
.



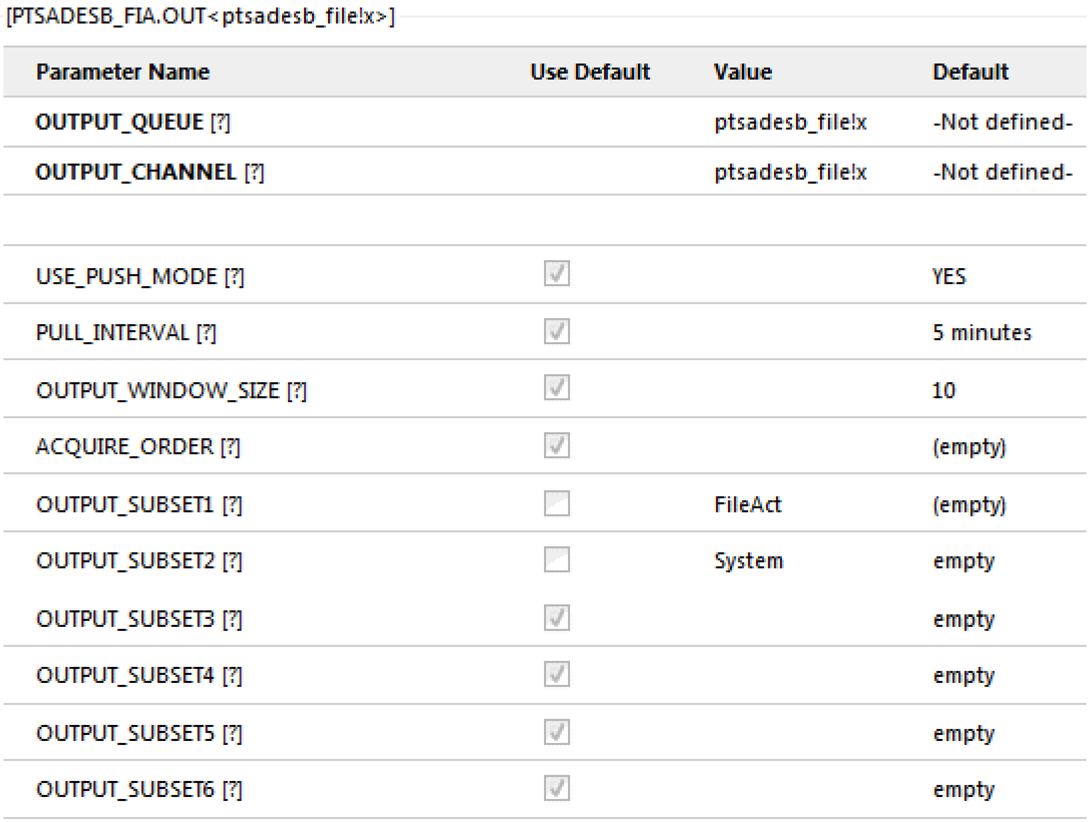
4.3 Using FileAct and InterAct over 2 Messaging Interfaces

It is possible to use two different Messaging Interfaces, one for FileAct messages and one for InterAct messages. In such a case the following must be considered:

- In order to avoid Delivery Notifications (System messages) to Messaging Interface 1 (FileAct) being received by Message Interface 2 (InterAct), two different notification queues must be used, one for FileAct message notifications and one for InterAct message notifications. This can, for example, be achieved by configuring Submission Profiles accordingly.
- If two Messaging Interfaces shall access one and the same SnF queue, each **Output Session** must use different OUTPUT_CHANNELs.



- In the **Output Session** configuration of the SnF queue for FileAct messages the parameter OUTPUT_SUBSET1 must be set to “FileAct”. This way only FileAct messages will be read from the queue.
  In the **Output Session** configuration of the SnF queue for InterAct messages the parameter OUTPUT_SUBSET1 must be set to “InterAct”.
- In the **Output Session** configuration of the SnF queue in which delivery notifications are expected, the parameter OUTPUT_SUBSET2 must be set to “System”.



[PTSADESB_FIA.OUT<ptsadesb_file!x>]

Parameter Name	Use Default	Value	Default
OUTPUT_QUEUE [?]		ptsadesb_file!x	-Not defined-
OUTPUT_CHANNEL [?]		ptsadesb_file!x	-Not defined-
USE_PUSH_MODE [?]	☑		YES
PULL_INTERVAL [?]	☑		5 minutes
OUTPUT_WINDOW_SIZE [?]	☑		10
ACQUIRE_ORDER [?]	☑		(empty)
OUTPUT_SUBSET1 [?]	☐	FileAct	(empty)
OUTPUT_SUBSET2 [?]	☐	System	empty
OUTPUT_SUBSET3 [?]	☑		empty
OUTPUT_SUBSET4 [?]	☑		empty
OUTPUT_SUBSET5 [?]	☑		empty
OUTPUT_SUBSET6 [?]	☑		empty

4.4 Create New Messaging Interface Module

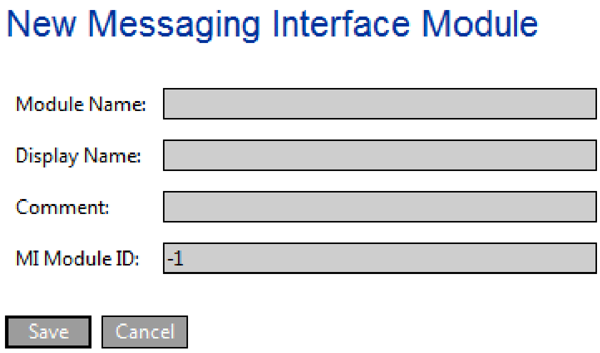
To create a new Messaging Interface module select **Administration / Messaging Interfaces** from the main menu.

Select **Modules** from the tree-view on the left hand side.

The **Messaging Interface Modules** page is displayed.

Press **Add**.

The **New Messaging Interface Module** page is displayed:



New Messaging Interface Module

Module Name:	
Display Name:	
Comment:	
MI Module ID:	-1

Fill in the parameters as follows:

Parameter	Description
Module Name	Name of the Messaging Interface (MI) module.
Display Name	Name of the MI module as displayed in the web-client.
Comment	Optional comment.
MI Module ID	ID of the Messaging Interface Module. **Mandatory parameter**. This Module ID has to be used in the start script of the MI Module.

Press **Save**.

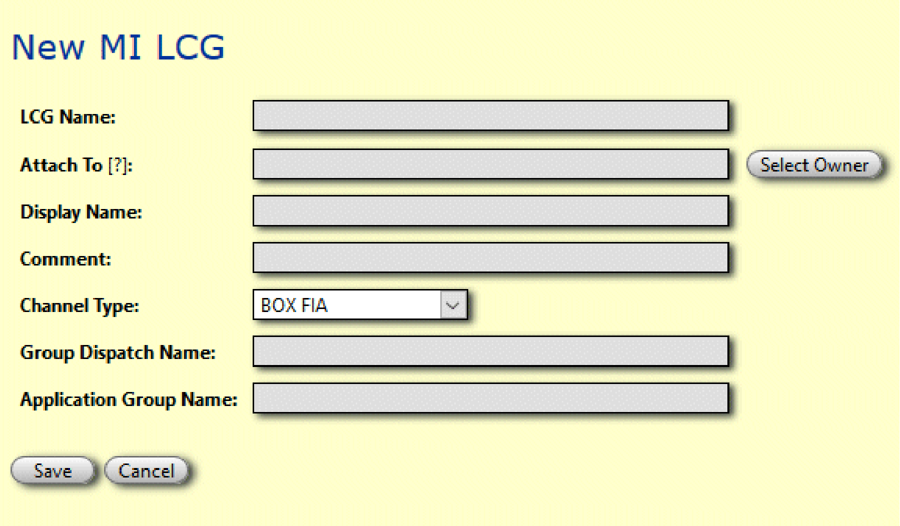
The newly created Messaging Interface (MI) module will now be visible in the list of MI modules.

4.5 Create New MI LCG

As shown in section **4.1 Architecture**, a Messaging Interface Module consists of one or more MI Local Channel Groups.

Select **MI LCG** from the tree view and press **Add**.

The **New MI LCG** page is displayed:



New MI LCG

LCG Name:

Attach To [?]:

Display Name:

Comment:

Channel Type: BOX FIA

Group Dispatch Name:

Application Group Name:

Fill in the parameters as follows:

Parameter	Description
LCG Name	Mandatory parameter! This is the name of this LCG. The name **must begin** with the BIC 8 of the institution to which the LCG belongs, i.e. with BIC 8 of the Operational Owner of this LCG (=UPM Company Node to which this LCG is attached). This BIC 8 corresponds with the BIC 8 in the second level node of the RequestorDN, e.g. if RequestorDN is "cn=intercope,**o=ptsadess**,o=swift", the LCG Name could be "**PTSADESS**_FIA". This **mandatory** parameter is used for linking to configuration. For this parameter the use of **replacement tokens is not possible.**
Attach To:	Specifies a company node to which the MI LCG shall be attached. This node becomes owner of the FIA Channel which is essential for being able to access it. Press **Select Owner** and select the node that the LCG shall be attached to.
Display Name	Name of the MI LCG as displayed in the web-client. Can be freely given as it is not referenced elsewhere in the system.
Comment	Optional comment
Channel Type	Channel type that all channels belonging to this MI module use: Select the option **BOX FIA** for FileAct
Group Dispatch Name:	This parameter used when selecting an LCG for submission of messages. For this parameter **replacement tokens can be used**
Application Group Name	For future use. Leave empty.

Press **Save**.

The newly created MI LCG module will now be visible in the list of MI LCGs.

Subsequently you will have to **Edit** and **Save** the configuration (**Config Parameters** page in the tree view) of the module in order to activate it.

4.6 Create New FileAct Channel

As shown in section **4.1 Architecture**, an MI LCG usually consists of one FIA channel.

To create an MI channel select **Administration / Messaging Interfaces** from the main menu.

Select **Modules** from the tree-view on the left hand side.

The **Messaging Interface Modules** page is displayed.

Select the MI module to which you want to add a channel by clicking to the respective **Module Name** link in the list of Messaging Interface Modules.

The **Messaging Interface Module <Module Name>** page is displayed.

Select **MI LCG** from the tree view.

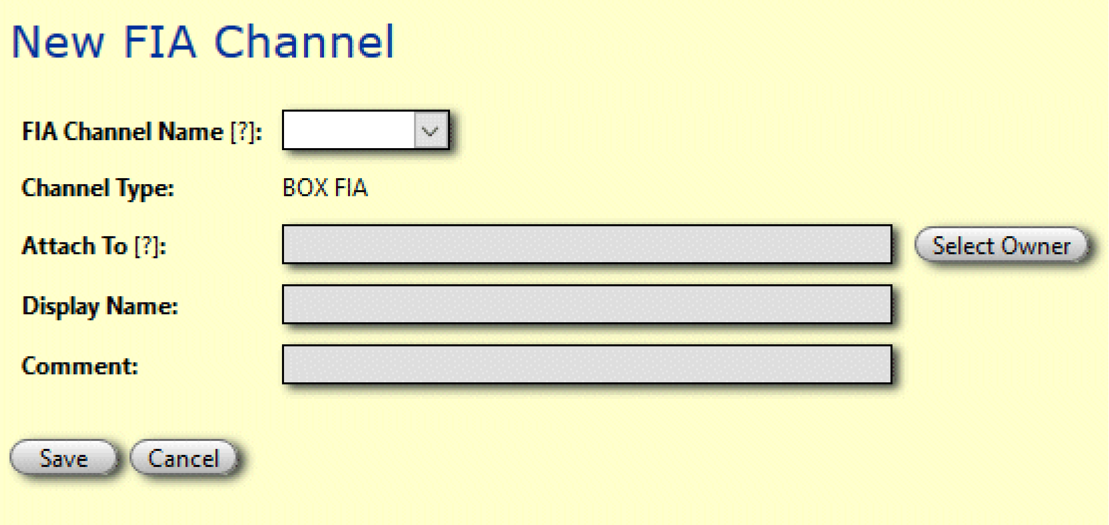
Select the MI LCG module to which you want to add the channel by clicking to the respective **Display Name** link in the list of LCGs.

The **BOX FIA LCG <LCG Name>** page is displayed.

Select **FIA Channels** from the tree view.

Press Add.

The **New FIA Channel** page is displayed:



New FIA Channel

FIA Channel Name [?]:

Channel Type: BOX FIA

Attach To [?]:

Display Name:

Comment:

Fill in the parameters as follows:

Parameter	Description
FIA Channel Name	Name of this Channel. The name must begin with the BIC 8 of the institution to which the channel belongs, i.e. with BIC 8 of the Operational Owner of this channel (=UPM Company Node to which the LCG to which this channel belongs is attached). This BIC 8 corresponds with the BIC 8 in the second level node of the RequestorDN, e.g. if RequestorDN is "cn=intercope,**o=ptsadess**,o=swift", the FIA Channel Name could be "**PTSADESS**_FIA". **Note** that the selection list shows only the FIA Channels listed in the FileAct/InterAct Channel ACL in the UPM (see 3.2) and that have not yet been attached to a node (see below).
Channel Type	Specifies the channel type. The value has been taken from the Channel type parameter of the LCG to which this channel belongs and cannot be changed. For FileAct channels the type is **BOX FIA**.

Parameter	Description
Attach To:	Specifies a company node to which the FIA Channel shall be attached. This node becomes owner of the FIA Channel which is essential for being able to access it. Press **Select Owner** and select the node that the FIA Channel shall be attached to.
Display Name	Name of the channel as displayed in the web-client
Comment	Optional comment.

Press **Save**.

The newly created FIA Channel will now be visible in the list of channels.

Subsequently you will have to **Edit** and **Save** the configuration (**Config Parameters subitems** FIA and LCG Channel in the tree view) of the module in order to activate it.

If you click on the **Name** link in the list of FIA Channels, the displayed page will show more detailed information on the selected channel.

If you select **Owner** from the tree view, you will be able to see the ownership details.

The upper section of the page, **Owner**, shows the ownership details for the Enterprise (level 5) node.

The lower section of the page, **Operator Owner**, shows the ownership details for the Company (level 4) node to which the FIA Channel has been attached.

4.7 FileAct Channel Scheduling

In order to view or to specify the Stop / Start behavior of a certain FIA Channel, select the respective channel from the list of FIA Channels and select **Scheduling** from the tree view.

The page that is displayed shows the following parameters:

Section **Channel Scheduling**

Parameter	Description
Schedule Mode	Specifies the Stop/Start behavior of the channel.
Scheduling Status	Scheduling status of channel. Active Scheduling is active Inactive Scheduling is not active Error Scheduling does not take place due to (configuration) error.
Scheduling Error	Error code applicable to scheduling.
Last Schedule Update	Last time when the schedule was updated.
Last Start Time	Last time when channel was started (selected) by automatic operation.
Last Stop Time	Last time when channel was stopped (logged out) by automatic operation.
Next Start Time	Next time when channel will be started (selected) by automatic operation.
Next Stop Time	Next time when channel will be stopped (logged out) by automatic operation.

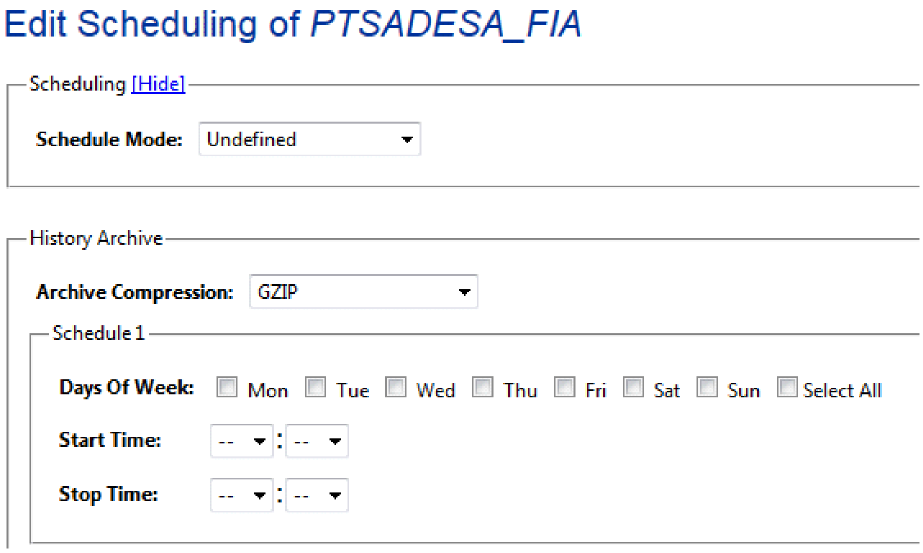
Section **History Archive Scheduling**

Parameter	Description
Archive Error:	Error code applicable to history archive scheduling.

Parameter	Description
Last Start Time:	Last time when history archiving was started.
Next Start Time:	Time when next history archiving will be started.
Days Of Week:	Configured day(s) of week when history archiving will be performed.
Start Time:	Configured time of day when history archiving will be started.
Archive Compression:	Compression mode of archived data.

To modify scheduling options, press **Edit**.

The **Edit Scheduling of <channel_name>** page is displayed:



Edit Scheduling of *PTSADESA_FIA*

Scheduling [Hide]

Schedule Mode: Undefined

History Archive

Archive Compression: GZIP

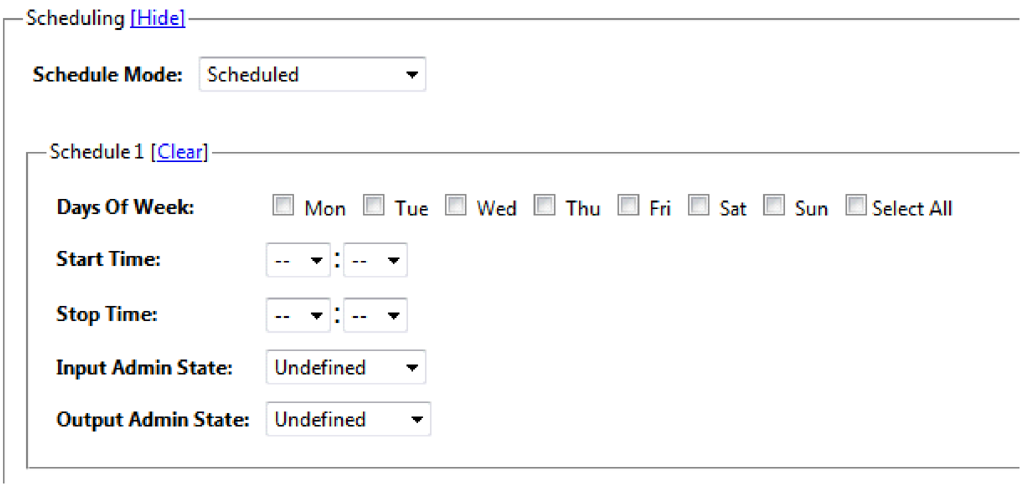
Schedule 1

Days Of Week: □ Mon □ Tue □ Wed □ Thu □ Fri □ Sat □ Sun □ Select All

Start Time: -- : --

Stop Time: -- : --

If you select the mode **Scheduled**, the page will be expanded and you will be able to enter scheduling options. The following figure shows a clipping of the displayed page:



Scheduling [Hide]

Schedule Mode: Scheduled

Schedule 1 [Clear]

Days Of Week: □ Mon □ Tue □ Wed □ Thu □ Fri □ Sat □ Sun □ Select All

Start Time: -- : --

Stop Time: -- : --

Input Admin State: Undefined

Output Admin State: Undefined

To ensure best possible scheduling options, there are three (3) separate schedules (the clipping above shows only one).

The following rules are valid for scheduling:

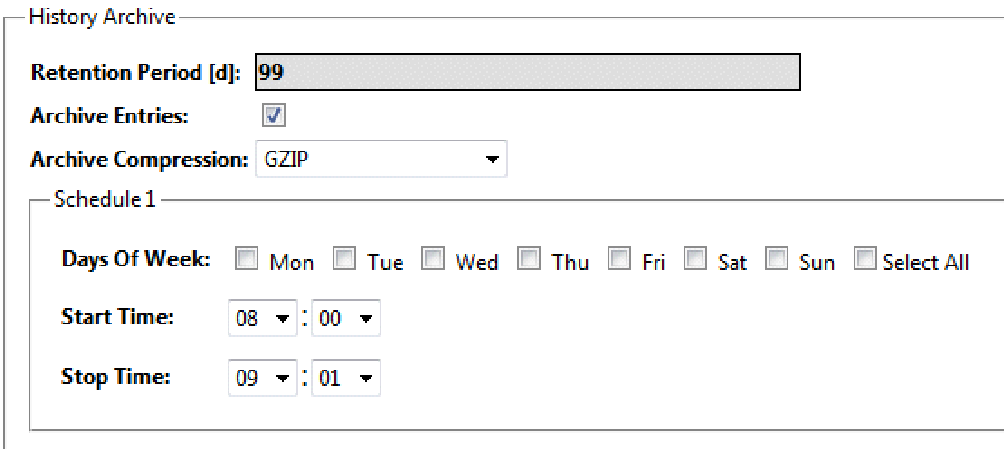
- A manual operator intervention in a schedule does not suspend scheduling. The effect of a manual operator intervention will last until the next scheduled start / stop.
- Open scheduling intervals are possible, i.e. undefined start or end time can be configured.
- If the stop time is earlier or equal than the start time, the effective stop time will be the next day.
- Scheduling intervals may span midnight.
- Schedule intervals may overlap or may include intervals of other schedules.

Enter the desired scheduling options and press the **Save** button at the bottom of the page.

4.8 Channel History Archive Scheduling

Channel History data can be archived as CLOB (Character Large Object) data into the database.

On the bottom of the Scheduling page you may configure archiving options for Channel History entries.



History Archive

Retention Period [d]: 99

Archive Entries: ☑

Archive Compression: GZIP

Schedule 1

Days Of Week: □ Mon □ Tue □ Wed □ Thu □ Fri □ Sat □ Sun □ Select All

Start Time: 08 : 00

Stop Time: 09 : 01

The **Retention Period [d]** parameter specifies the number of days the Channel History entries will remain in the Channel History table (see 10.5) before being handled by the Archiver.

If the **Archive Entries** flag is set, the Channel History entries will be archived. If the flag is not set, the entries will be deleted. Also, if this flag is set, then history entries may be exported in same XML-encoding as in archive into the file system. Please refer to the BOX Configuration Guide, parameter **ARCHIVE_EXPORT_DIR** in section **[<FIA-CHANNELNAME>.HISTARCH]** or to the parameter description in **MI LCG / FileAct/InterAct Channels / Scheduling and Config / Archive Config Parameters** in the Web-Client.

Archive Compression specifies the format in which the Channel History entries shall be compressed before archiving.

To ensure best possible scheduling options, there are two (2) separate schedules (the clipping above shows only one). For both archive schedules an optional stop time may be specified. If the stop time is reached, the archiver will finish immediately (directly after completion of current archive entry/file).

Enter the desired scheduling options and press **Save**.

The history entries will then be archived as specified. Archived history entries will not be visible in the History table (see 10.4) anymore but can be viewed in the History Archive (see 10.5).

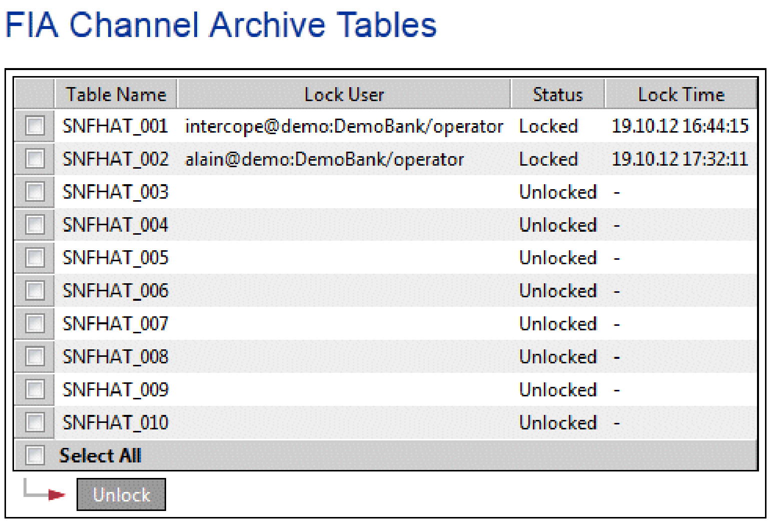
4.8.1 Unlock Channel Archive Tables

When a user decompresses a Channel History Archive (see 10.5) the archived history entries of the selected channel are written into a database table (Channel Archive Table). The Archive Table will be locked for other users.

As the number of available Channel Archive Tables is limited (normally 10 tables but if required, you can contact Intercope for additional tables), there may be need to unlock Channel Archive Tables.

To unlock Channel Archive Tables that are currently locked by other users, select **MI Operation** from the main menu.

Select **History Archive Tables / FIA Channel** from the tree-view on the left hand side.



FIA Channel Archive Tables

	Table Name	Lock User	Status	Lock Time
□	SNFHAT_001	intercope@demo:DemoBank/operator	Locked	19.10.12 16:44:15
□	SNFHAT_002	alain@demo:DemoBank/operator	Locked	19.10.12 17:32:11
□	SNFHAT_003		Unlocked	-
□	SNFHAT_004		Unlocked	-
□	SNFHAT_005		Unlocked	-
□	SNFHAT_006		Unlocked	-
□	SNFHAT_007		Unlocked	-
□	SNFHAT_008		Unlocked	-
□	SNFHAT_009		Unlocked	-
□	SNFHAT_010		Unlocked	-
□	**Select All**			

Select the FIA Channel Archive Table (with Status 'Locked') you want to unlock and press **Unlock**.

4.9 Input / Output Sessions

4.9.1 Input Session

The default Input Session is used for all FileAct traffic. There is no need to create a dedicated Input Session.

4.9.2 OutPut Sessions

Dedicated Output Sessions must be defined for real-time output traffic and for each SnF queue to be received from. This is done as follows:

To create FIA channel Output Sessions select **Administration / Messaging Interfaces f**rom the main menu.

Select **Modules** from the tree-view on the left hand side.

The **Messaging Interface Modules** page is displayed.

Select the MI module to which you want to add output sessions by clicking to the respective **Module Name** link in the list of Messaging Interface Modules.

The **Messaging Interface Module <Module Name>** page is displayed.

Select **MI LCG** from the tree view.

Select the MI LCG module to which you want to add output sessions by clicking to the respective **Display Name** link in the list of LCGs.

The **BOX FIA LCG <LCG Name>** page is displayed.

Select **FIA Channels** from the tree view.

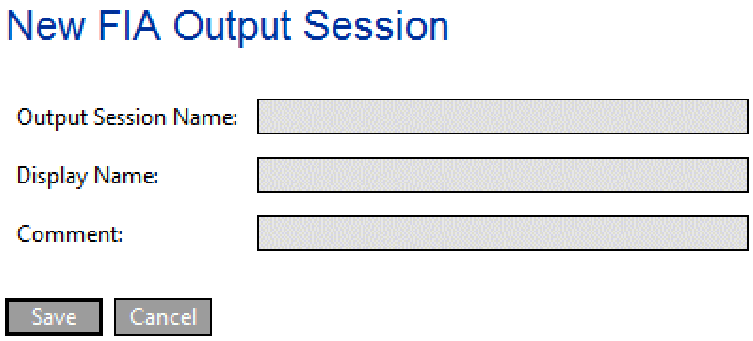
Select the channel to which you want to add output sessions by clicking to the respective **Name** link in the list of channels.

The **FIA Channel <Channel Name>** page is displayed.

Select **Output Sessions** from the tree view.

Press **Add**.

The **New FIA Output Session** page is displayed:



New FIA Output Session

Output Session Name:

Display Name:

Comment:

Parameter	Description
Output Session Name	Name of this Output Session. **The Session name must be same as the name of the SWIFT SnF Queue** in which SWIFT Output messages are received, e.g. ptsadess_msg!x. This name corresponds to the SWIFT Service Name. The Output Session Name for the session used for real-time traffic must be “realtime”.
Display Name	Name of the Output Session as displayed in the web-client.
Comment	Optional comment.

Press **Save**.

The newly created output session will now be visible in the list of output sessions.

Click to link of the created Output Session.

The **FIA Output Session <outputsession_name>** page is displayed

Go to **Configuration Parameters** in the tree view.

Press **Edit**.

The **Edit Server Config Parameters of <Channel_Name>** page is displayed.

Configure the Output Session. For values and parameter descriptions, move the cursor to the **[?]** symbol.

Press **Save**.

Remember that you must create **one dedicated Output Session for real-time traffic** and **one Output Session for each SWIFT SnF Queue** in which SWIFT Output messages are received.

4.9.3 SnF Output Channels

For each Output Session you can specify whether an Output Channel shall be used for message traffic or not. If no channel is used, the traffic uses a queue session started by an AcquireSnFRequest.

If an Output Channel shall be used for opening a SnF Output session, the configuration parameter **OUTPUT_CHANNEL** in section **[<FIA-CHANNELNAME>]** must be set accordingly, i.e. a channel name must be specified.

If no Output Channel is configured (i.e. the configuration parameter **OUTPUT_CHANNEL** in section **[<FIA-CHANNELNAME>]** is left empty), no Output Channel is used and the message traffic uses a queue session started by an AcquireSnFRequest.

If an Output Channel is specified, the name of it must follow the following convention:

```
Outputchannel = bic8"_"component["!" environment]
```

The component part allows identification of different Output Channels for a given BIC8. Users can freely choose this part of the name.

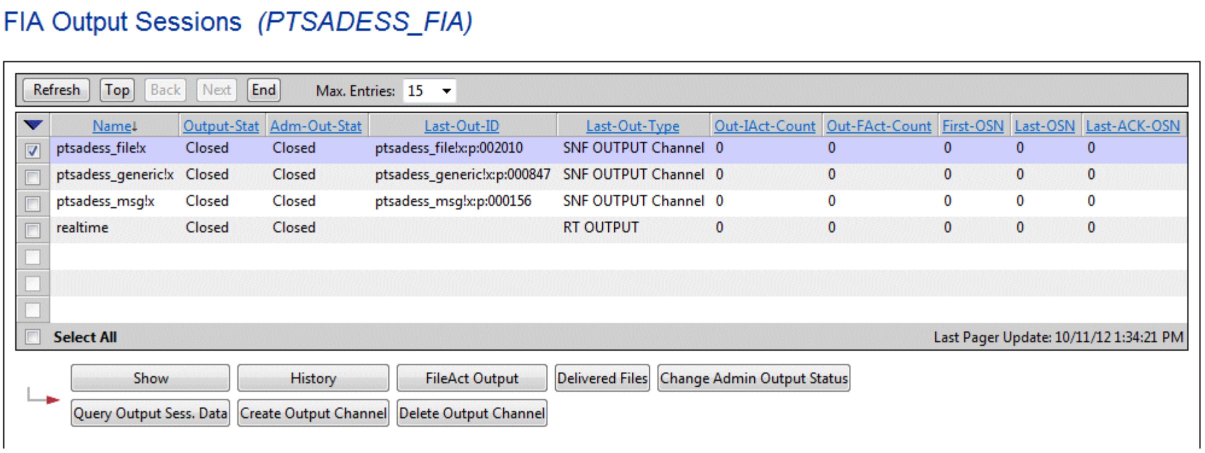
The environment identifies whether the input channel is used on ITB, Pilot or Live, using the same naming conventions as for the service name.

For more information about the conventions for the service name, see the *SWIFTNet Service Design Guide* and the *Interface SWIFTNet 7.0 Vendor Specifications for InterAct and FileAct*.

Output channel names are always in **lower case**, e.g.: `ptsadess_file!x`

To create an Output Channel press you must first specify the name of it in the configuration parameter **OUTPUT_CHANNEL** in section **[<FIA-CHANNELNAME>]**) and then select the respective FIA Channel in the **MI Operation / FileAct/InterAct Channels** menu and then press **Output Sessions**.

The **FIA Output Sessions (<FIA_Channel_Name>)** page is displayed.



FIA Output Sessions *(PTSADESS_FIA)*

	Name↓	Output-Stat	Adm-Out-Stat	Last-Out-ID	Last-Out-Type	Out-IAct-Count	Out-FAct-Count	First-OSN	Last-OSN	Last-ACK-OSN
☑	ptsadess_file!x	Closed	Closed	ptsadess_file!x:p:002010	SNF OUTPUT Channel	0	0	0	0	0
□	ptsadess_generic!x	Closed	Closed	ptsadess_generic!x:p:000847	SNF OUTPUT Channel	0	0	0	0	0
□	ptsadess_msg!x	Closed	Closed	ptsadess_msg!x:p:000156	SNF OUTPUT Channel	0	0	0	0	0
□	realtime	Closed	Closed		RT OUTPUT	0	0	0	0	0
□										
□										
□										

□ Select All

Last Pager Update: 10/11/12 1:34:21 PM

Select the Output Session for which you want to create an Output Channel and press **Create Output Channel**.

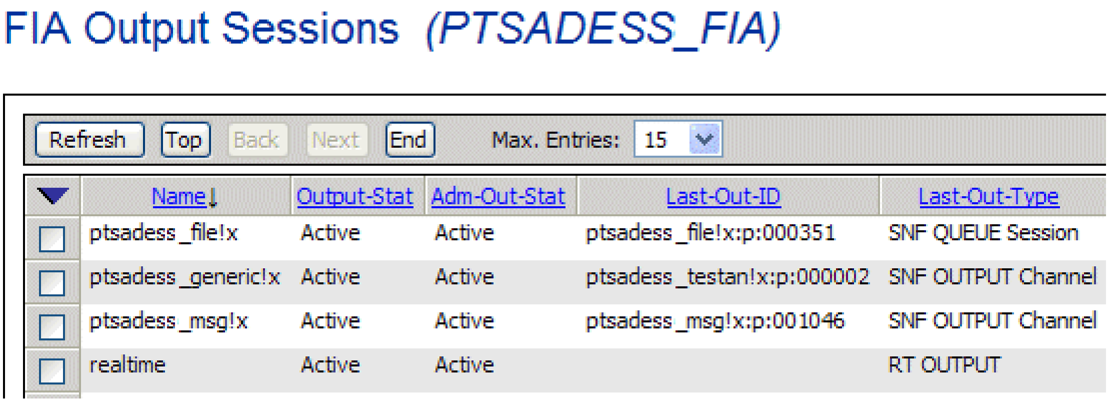
Now a Request to create an Output Channel **(CreateOutputChannelSnFRequest)** is sent to SWIFT.

The figure below shows three SnF Output Sessions (and one real-time session).

ptsadess_file!x does not use an Output Channel (see column **Last-Out-Type**).

ptsadess_msg!x uses an Output Channel (see column **Last-Out-Type**) with the same name as the SNF queue. This can be seen in the **Last-Out-ID** column.
Last-Out-ID is the **Sw:SnFSessionId** containing the queue name or the output channel name, the session mode (pull or push) and a session number.

ptsadess_generic!x uses an Output Channel (see column **Last-Out-Type**) with another name than the SNF queue (ptsadess_testan!x). This can be seen in the **Last-Out-ID** column.



FIA Output Sessions *(PTSADESS_FIA)*

	Name	Output-Stat	Adm-Out-Stat	Last-Out-ID	Last-Out-Type
□	ptsadess_file!x	Active	Active	ptsadess_file!x:p:000351	SNF QUEUE Session
□	ptsadess_generic!x	Active	Active	ptsadess_testan!x:p:000002	SNF OUTPUT Channel
□	ptsadess_msg!x	Active	Active	ptsadess_msg!x:p:001046	SNF OUTPUT Channel
□	realtime	Active	Active		RT OUTPUT

4.10 Create New FileAct Channel Connection

As shown in section **4.1 Architecture**, an MI LCG consists of (at least) one MI channel.
Each channel may use several SAG connections **(e.g. for fall-back)**.

To create MI channel connections select **Administration / Messaging Interfaces** from the main menu.

Select **Modules** from the tree-view on the left hand side.

The **Messaging Interface Modules** page is displayed.

Select the MI module to which you want to add channel connections by clicking to the respective **Module Name** link in the list of Messaging Interface Modules.

The **Messaging Interface Module <Module Name>** page is displayed.

Select **MI LCG** from the tree view.

Select the MI LCG module to which you want to add channel connections by clicking to the respective **Display Name** link in the list of LCGs.

The **BOX FIA LCG <LCG Name>** page is displayed.

Select **FIA Channels** from the tree view.

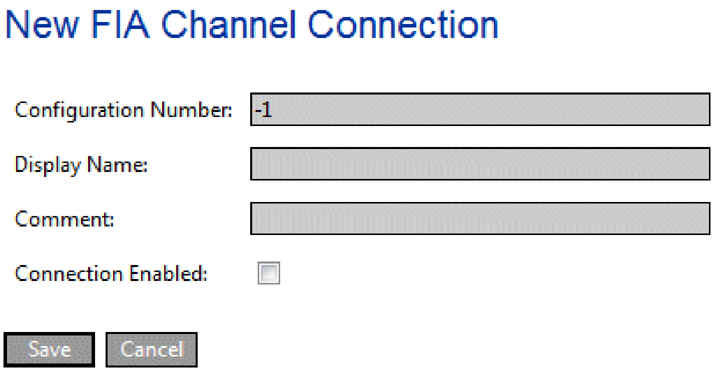
Select the channel to which you want to add connections by clicking to the respective **Name** link in the list of channels.

The **FIA Channel <Channel Name>** page is displayed.

Select **Connections** from the tree view.

Press **Add**.

The **New FIA Channel Connection** page is displayed:



New FIA Channel Connection

Configuration Number:	-1
Display Name:	
Comment:	
Connection Enabled:	□

Fill in the parameters as follows:

Parameter	Description
Configuration Number	Configuration Number of this Channel Connection. Unique number of this connection configuration. Use **1** for the first connection, **2** for the second, etc.
Display Name	Name of the channel as displayed in the web-client
Comment	Optional comment
Connection Enabled	Tick-box to enable the connection, i.e. to specify that the connection may be used when connecting to SAG.

Press **Save**.

The newly created Channel Connection will now be visible in the list of Connections.

If you click on the **Display Name** link in the list of Connections, the displayed page will show more detailed information on the selected Connection. Additionally to the parameters at creation time, an **internal ID** of the connection is displayed.

4.10.1 Multiple SAG Connections

If you create **multiple SAG connections** for one Channel, each connection gets a unique **Order Number**, i.e. the order number **1** for first connection, **2** for the second, etc.

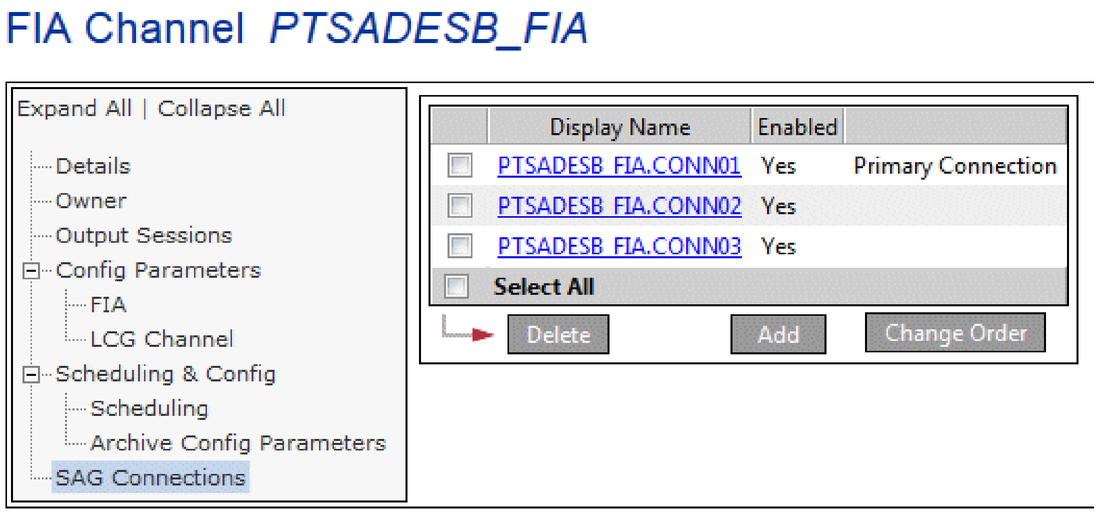
The SAG connections will be used for connecting to SAG in the order specified by the order number.

The connection with the lowest order number is called **Primary Connection** and this connection will be used primarily when connecting to SAG.

If one connection fails, the connection with the next lowest order number (configured for this Channel and with the **Enabled** flag set) will be used.

Note: The next time when the system is started the connection last used - not the Primary connection - will be used for connecting to SAG. To have the Primary Connection be used again, you can use the **Reset to Primary Connection** function on the **FileAct / InterAct Channels** page (see 10.16).

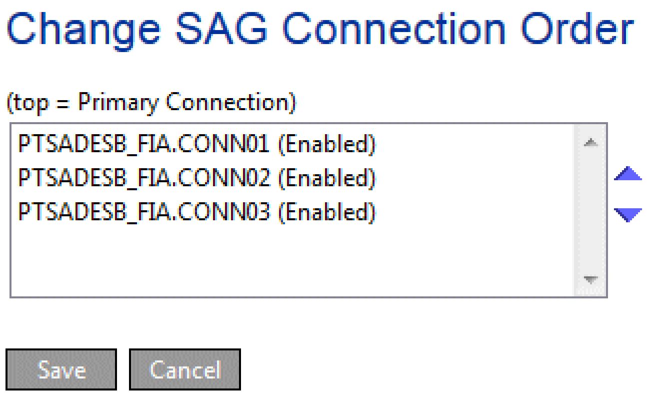
The order of the connections can be changed:


FIA Channel *PTSADESB_FIA*

	Display Name	Enabled	
□	PTSADESB_FIA.CONN01	Yes	Primary Connection
□	PTSADESB_FIA.CONN02	Yes	
□	PTSADESB_FIA.CONN03	Yes	
□	Select All		

Press the **Change Order** button.

The **Change SAG Connection Order** page is displayed:


Change SAG Connection Order

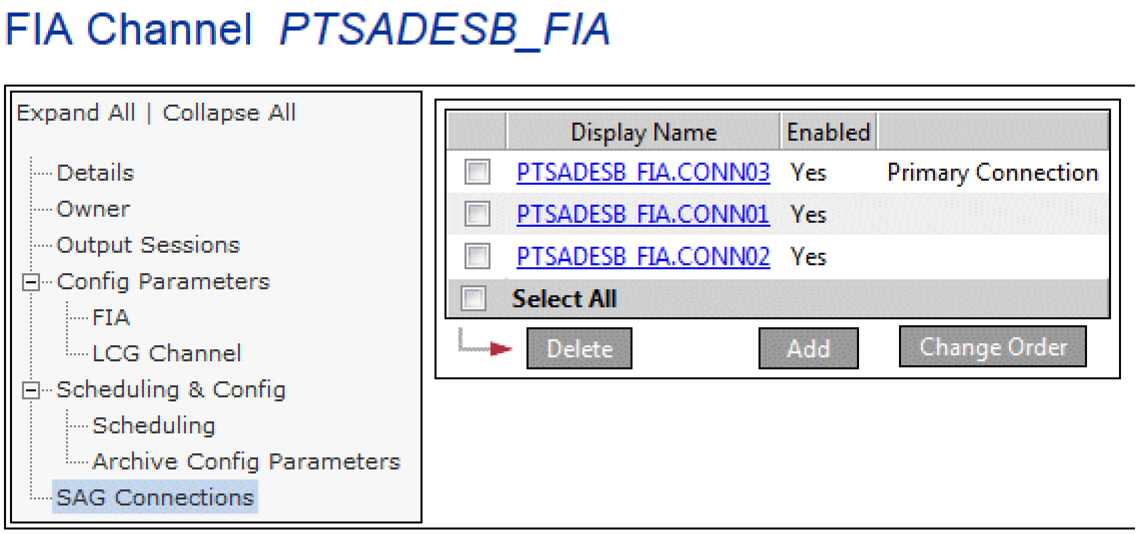
(top = Primary Connection)

PTSADESB_FIA.CONN01 (Enabled)
PTSADESB_FIA.CONN02 (Enabled)
PTSADESB_FIA.CONN03 (Enabled)

Select the connection you want to move and use the **Up** and / or **Down** arrow(s) on the right hand side.

Press **Save**.

The changed order will be visible on the page listing the SAG Connections specified for this channel:


FIA Channel *PTSADESB_FIA*

	Display Name	Enabled	
□	PTSADESB_FIA.CONN03	Yes	Primary Connection
□	PTSADESB_FIA.CONN01	Yes	
□	PTSADESB_FIA.CONN02	Yes	
□	Select All		

4.11 FileAct Submission Profiles

FileAct Submission Profiles can be seen as templates for sending requests to put files (Put File Profiles) and to get files (Get File Profiles) and are used in order to enrich message data (Put/GetFileTransferParameters and, if applicable, the Generic Attributes) with data from the Submission profile.

4.11.1 FileAct – Put File Profiles

FileAct Put File Profiles are used for sending a request to put a file.

The messages sent by a backend application or created using the Web-client are checked by a Content Processing Instruction (CPI) module in regard to matching data in the Submission Profiles.

The Submission Profile to be used can be retrieved in two ways.

If a '**profile**' attribute is supplied in the IGXMTransactionData node of the FileAct message (MPS), the CPI tries to find a profile with a matching **ReferenceName**.

If no '**profile**' attribute is supplied, the CPI uses the **Filter Regular Expression** parameter in the Submission Profiles for finding a matching Submission Profile.

If a Submission Profile is found, the CPI enriches the IGXMTransactionData node in the FileAct message with data from the determined Submission Profile.

If no profile is found, the CPI fails and processing continues with the failure label.

The data in the FilterExpression is of the form: <FilterName>=<FilterValue>

The FilterName can be one item from the following list:

Filter Name	Description
GENATTR[REQUESTOR_DN]	Compares filter value with requestor DN in generic attributes
GENATTR[RESPONDER_DN]	Compares filter value with responder DN in generic attributes
GENATTR[SERVICE_NAME]	Compares filter value with service name in generic attributes
GENATTR[REQUEST_TYPE]	Compares filter value with request type in generic attributes
GENATTR[MESSAGE_TYPE]	Compares filter value with message type in generic attributes
GENATTR[CATEGORY]	Compares filter value with message category in generic attributes.
GENATTR[LOGICAL_FILE_NAME]	Compares filter value with Logical Filename in generic attributes.
IAFABA[QUEUE_MGR]	Compares filter value with queue manager name of origination report protocol data (FileAct BA)
IAFABA[QUEUE]	Compares filter value with queue name of origination report protocol data (FileAct BA)
IAFABA[CORREL_ID]	Compares filter value with correlation id of origination report protocol data (FileAct BA)
IAFABA[MSG_ID]	Compares filter value with message id of origination report protocol data (FileAct BA)
IAFABA[REPLY_QUEUE_MGR]	Compares filter value with reply queue manager name of origination report protocol data (FileAct BA)
IAFABA[REPLY_QUEUE]	Compares filter value with reply queue name of origination report protocol data (FileAct BA)
IAFABA[APPL_ORIG_DATA]	Compares filter value with put application orig data origination report protocol data (FileAct BA)
IAFABA[PUT_APPL_NAME]	Compares filter value with put application name of origination report protocol data (FileAct BA)

Filter Name	Description
IAFABA[APPL_ID_DATA]	Compares filter value with application id data of origination report protocol data (FileAct BA)
FILE[EXCH_DIR]	Compares filter value with exchange directory name of origination report protocol data (FILE EXCH)
FILE[FILE_NAME]	Compares filter value with file name of origination report protocol data (FILE EXCH)
ORIGREP[ORIG_DISP_ID]	Compares filter value with originator display id of origination report
ORIGREP[RECV_DISP_ID]	Compares filter value with receiver display id of origination report
ORIGREP[ORIG_MSG_ID]	Compares filter value with originator message id of origination report
ORIGREP[RECV_MSG_ID]	Compares filter value with receiver message id of origination report

Additionally the following Application Data fields can be used as FilterName, i.e. they can be used for comparing the filter value with data in the Application Data fields of the Generic Attributes:

Filter Name
APPLDATA[SOURCEAPPL]
APPLDATA[TARGETAPPL]
APPLDATA[APPLDATA1]
APPLDATA[APPLDATA2]
APPLDATA[APPLDATA3]
APPLDATA[APPLDATA4]
APPLDATA[USERDATA1]
APPLDATA[USERDATA2]
APPLDATA[USERDATA3]
APPLDATA[MINIAPPLDATA1]
APPLDATA[MINIAPPLDATA2]
APPLDATA[SHORTAPPLDATA1]
APPLDATA[SHORTAPPLDATA2]
APPLDATA[SHORTAPPLDATA3]
APPLDATA[SHORTAPPLDATA4]
APPLDATA[LARGEAPPLDATA1]
APPLDATA[LARGEAPPLDATA2]

The FilterValue can be a regular expression that is to be compared with the MPS data.

Example:

The FilterExpression "GENATTR[REQUESTOR_DN] = cn=requestor, o=abcdefgh, o=swift" matches if the RequestorDN field of the generic attributes of the MPS equals to "cn=requestor, o=abcdefgh, o=swift".

Note: The filter names are normalized. All characters are converted to upper case and all blanks and tabs are removed. so "GenAttr[Requestor_ DN]" and "GENATTR[REQUESTOR_DN]" both are valid.

For filter values only leading and trailing blanks are removed. The requestor and responder DN are also normalized so that blanks in the DN are removed before comparison. So both "cn=requestor,o=abcdefgh,o=swift" and "cn=requestor, o= abcdefgh, o=swift" will have the save result.

The Profiles are checked until a matching Submission Profile is found.

If no Submission Profile was found, no enrichment of the IGXMTransactionData will occur, and in the CPI history result message the text like "No matching Submission Profile found for reference name '<profile name>'" indicates that no Submission Profile was found.

If the <profile name> field is empty, the Profile could not be found by means of Filter Expression search. If the <profile name> field contains a profile name, the CPI could not find a Submission Profile with the ReferenceName matching the ‘profile’ attribute in the IGXMTransactionData node of the message)

If the enrichment of the TransactionData was successful, the result message contains a message like "MPS enriched with data from profile 'Profile3' (id 3, order number 3) found by filter expression 'GENATTR[REQUESTOR_DN]=cn = johndoe,o=*' ".

If a Profile was found, the enrichment is done in the following way. The data from the Submission Profile is added to the IGXMTransactionData if the data is not already existing in the TransactionData. If the data already exists in the TransactionData, the data from the Profile is ignored.

The enriched message is stored in a new Content Version (CV) / Content Version Representation (CVR).
If data in the TransactionData that is also stored in the generic attributes (requestor DN, responder DN, service name, message type) is changed, the generic attributes will be updated by the CPI.

The values in Submission Profiles (like Requestor Distinguished Name(DN)) may contain configuration replacement tokens from configuration replacement file.

These tokens are replaced by the actual values when a Profile is used to enrich a MPS.

The values in Submission Profiles (like Logical Name of File) may contain message attribute tokens to use information from the specific message in the subsequent file transfer, e. g. the logical filename may be set from some message attribute. The general form of a message attribute is ${<filter name>}, where <filter name> is chosen from the list above.

For example ${IAFABA[APPL_ID_DATA]} would choose the information contained in ApplicationIdentityData from the MQMD if the MPS/File was received through the FileAct backend gateway via MQ. These tokens are replaced by there actual values when a profile is used to enrich a MPS.

The values in Submission Profiles (like Logical Name of File) may contain date replacement tokens to use the creation date of the specific message in the subsequent file transfer,
e. g. the logical filename may be set or enriched with the creation date. The general form of a date replacement token is ${DATE[<format string>]}, where <format string> defines the format of the date string representation. It can contain any characters and the following substitutions will take place:

YYYY will be replaced with the year (4 digits)
YY will be replaced with the year (2 digits)
MM will be replaced with the month (2 digits)
DD will be replaced with the day (2 digits)

For example the string "File${DATE[YY-MM-DD]}" will be transformed to "File12-09-07" if the message creation date is 2012/09/07.

The replacement token TIMESTAMP allows enriching the message data with the current timestamp.

Syntax: ${TIMESTAMP[<timestamp-format>]}

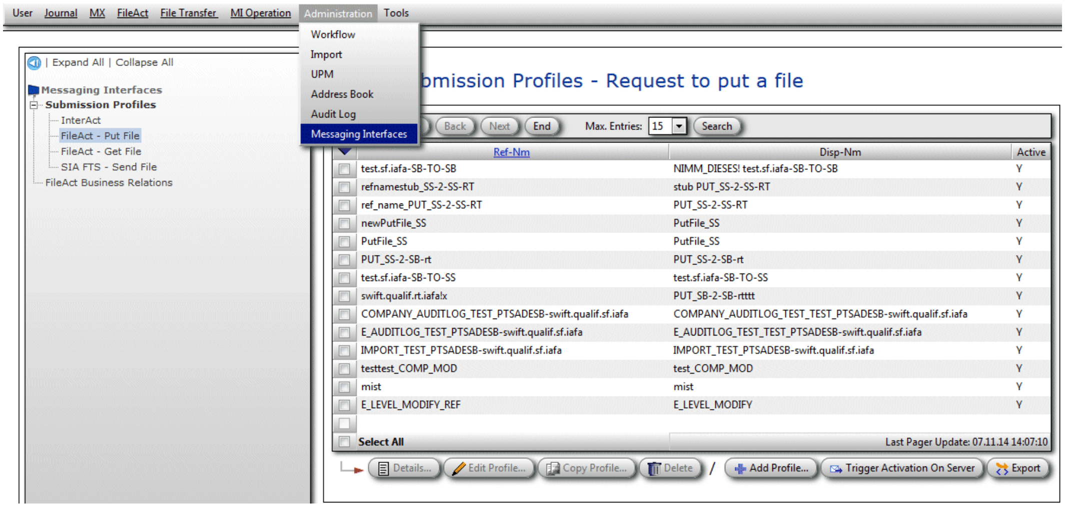
The timestamp-format defines how the timestamp is inserted into the message data:

YYYY	year (4 digits)
YY	year (2) digits
MM	month (2 digits)
DD	day (2 digits)
hh	hours (2 digits)
mm	minutes (2 digits)
ss	seconfs (2 digits)

4.11.1.1 Create a FileAct Put File Submission Profile

To create a FileAct Put File Submission Profile select **Administration / Messaging Interfaces** from the main menu.

Select **Submission Profiles / FileAct - Put File** from the tree-view.

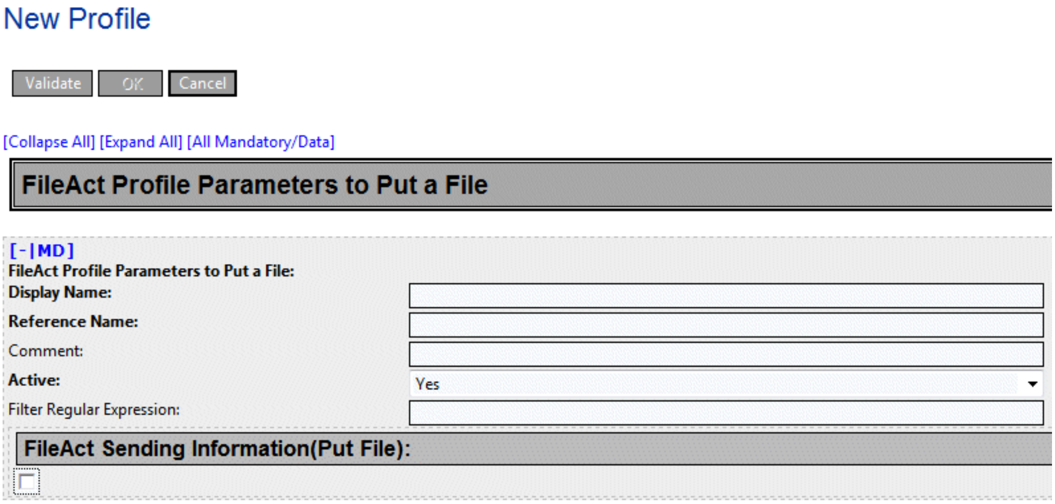


	Ref-Nm	Disp-Nm	Active
□	test.sf.iafa-SB-TO-SB	NIMM_DIESES! test.sf.iafa-SB-TO-SB	Y
□	refnamestub_SS-2-SS-RT	stub PUT_SS-2-SS-RT	Y
□	ref_name_PUT_SS-2-SS-RT	PUT_SS-2-SS-RT	Y
□	newPutFile_SS	PutFile_SS	Y
□	PutFile_SS	PutFile_SS	Y
□	PUT_SS-2-SB-rt	PUT_SS-2-SB-rt	Y
□	test.sf.iafa-SB-TO-SS	test.sf.iafa-SB-TO-SS	Y
□	swift.qualif.rt.iafa!x	PUT_SB-2-SB-rtttt	Y
□	COMPANY_AUDITLOG_TEST_PTSADESB-swift.qualif.sf.iafa	COMPANY_AUDITLOG_TEST_TEST_PTSADESB-swift.qualif.sf.iafa	Y
□	E_AUDITLOG_TEST_PTSADESB-swift.qualif.sf.iafa	E_AUDITLOG_TEST_TEST_PTSADESB-swift.qualif.sf.iafa	Y
□	IMPORT_TEST_PTSADESB-swift.qualif.sf.iafa	IMPORT_TEST_PTSADESB-swift.qualif.sf.iafa	Y
□	testtest_COMP_MOD	test_COMP_MOD	Y
□	mist	mist	Y
□	E_LEVEL_MODIFY_REF	E_LEVEL_MODIFY	Y
□			
□	Select All	Last Pager Update: 07.11.14 14:07:10	

The **FileAct Submission Profiles – Request to put a file** page is displayed:

Press the **Add Profile…** button.

The **New Profile** page is displayed:


New Profile
Validate
OK
Cancel
[Collapse All] [Expand All] [All Mandatory/Data]
FileAct Profile Parameters to Put a File
[-|MD]
FileAct Sending Information(Put File):

FileAct Profile Parameters to Put a File:

Display Name:

Reference Name:

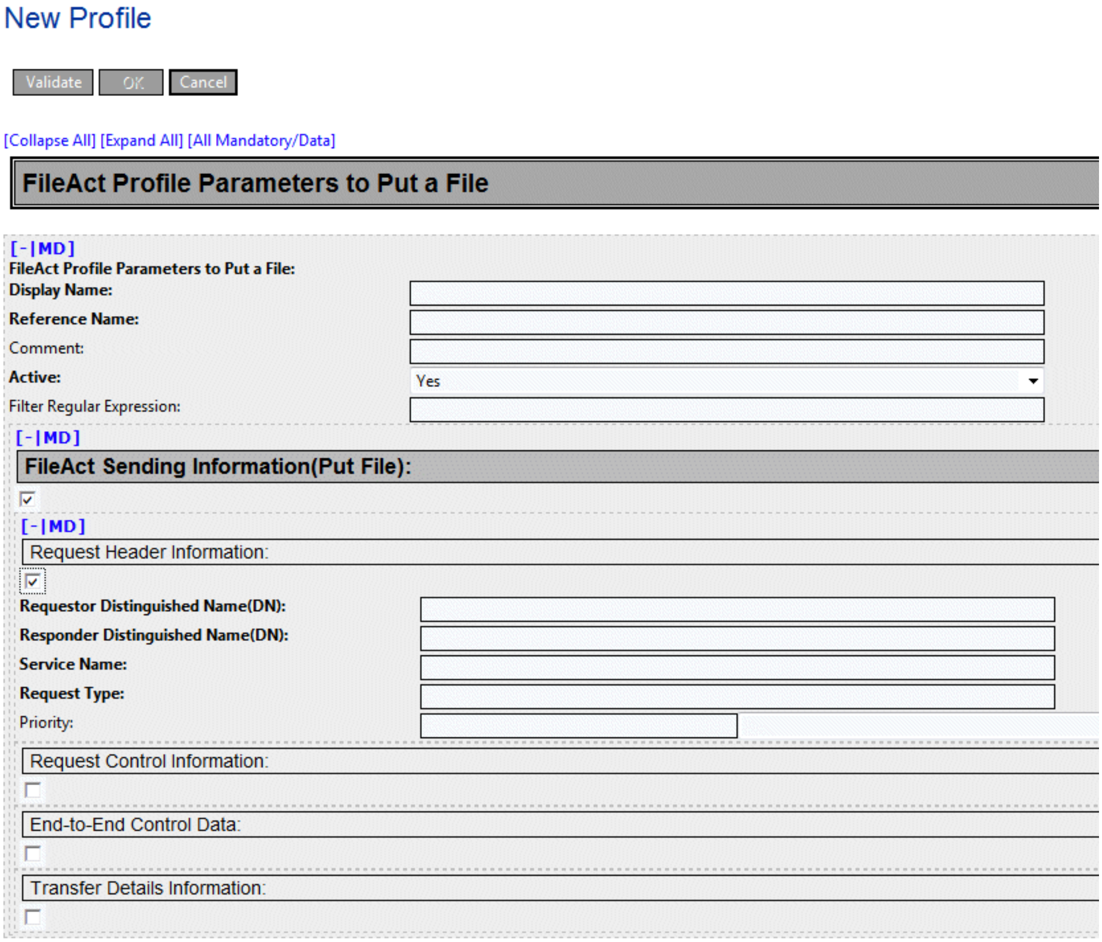
Comment:

Active: Yes

Filter Regular Expression:

□

By ticking the box under FileAct Sending Information (Put File) you can expand the page to show more file transfer parameters.



New Profile

FileAct Profile Parameters to Put a File

[-|MD]
FileAct Profile Parameters to Put a File:

Display Name:	
Reference Name:	
Comment:	
Active:	Yes
Filter Regular Expression:	

[-|MD]
FileAct Sending Information(Put File):
☑
[-|MD]
Request Header Information:
☑

Requestor Distinguished Name(DN):	
Responder Distinguished Name(DN):	
Service Name:	
Request Type:	
Priority:	

Request Control Information:
□
End-to-End Control Data:
□
Transfer Details Information:
□

The parameters in **bold** are mandatory.

The parameter descriptions can be seen by holding the cursor on the respective parameter.

The following parameters are of special importance:

Parameter	Description
Display Name	Name of the Profile as displayed in the client application.
Reference Name	This parameter is used for finding the ‘correct’ Profile (see above, 4.11.1).
Active	Flag specifying whether this Profile shall included in the check for matching data or not.
Filter Regular Expression	This parameter is used for filtering purposes (see above, 4.11.1). The field supports regular expressions and expects all data in the syntax of regular expressions (for a detailed description of regular expressions and syntax, see e.g. http://download.oracle.com/javase/1.4.2/docs/api/java/util/regex/Pattern.html).
Requestor Distinguished Name (DN)	This parameter is of special importance because the channel to which the message will be dispatched is determined by the level 2 node (o=<bic8>) of the entry. The channel name corresponds to the 0=<bic8>) value.
Service Name	Name of SWIFT Service to be used.
Request Type	Request type (please refer to SWIFT documentation).

Fill in the data and press **Validate**. If the validation test is successful, press **OK** to save the profile.

Any number of Submission Profiles may exist.

To view details of a FileAct Input Profile select the profile by ticking the respective box in the list and press the **Details** button.

To Modify a FileAct Input Profile select the profile to be modified by ticking the respective box in the list and press the **Modify** button.

To delete a FileAct Input Profile select the profile to be deleted by ticking the respective box in the list and press the **Delete** button.

Be sure to press the **Trigger Activation On Server** button after each change (modify, delete, add, change order).

4.11.2 FileAct Get File Profiles

FileAct Get File Profiles are used for sending requests to get a file.

The concept of determining the Get File Profile to be used is the same as described above for Put File Profiles.

Creating FileAct Get File Profiles is analog to creating of Put File Profiles (see 4.11.1).

4.11.3 Export Submission Profiles

Submission Profiles can be exported to a file by pressing the **Export** button on the bottom of the page.

Pressing this button exports complete Submission Profile sets. A Profile set is defined by the ownership of the Profile (UPM object it belongs to) and the type of Profile (Put File, Get File).

This means that you cannot export single Profiles but sets of Profiles only and on the other hand that you must export Put File Profiles and Get File Profiles separately.

The data to be exported will be displayed in XML format.

Save the page using the **Save as...** functionality of the browser.

4.11.4 Import Submission Profiles

To import Submission Profiles select **Administration / Messaging Interfaces.**

Select **Submission Profiles / FileAct - Put File** (or **/ FileAct - Get File**) from the tree view.

The displayed page shows an **Import** button. Press the button.

The **Import** page is displayed.

Either enter file name and path of the file you want to import or browse for the file you want to import.

As by the export of Profiles the import of Submission Profiles always covers a complete set of Profiles.

Note: When a set of Profiles is imported, the complete existing set (of the same type and belonging to the same owner) will be deleted.

4.12 FileAct Business Relations

The decision whether a received file shall be accepted or rejected can be done by using **FileAct Business Relations**.

BOX allows defining a **set of FileAct Business Relations for each Responder BIC** (= Own BIC).

The messaging interface compares the message data with data (filtering rules) defined in the Business Relations.

The relations will be checked one after another according to their order number. The first matching relation will be used to determine whether the message shall be accepted or rejected.

The relation can also be used for setting a workflow pattern for the MPS and for adding Application data into the Generic Attributes.

If no Business Relation set is defined for a responder BIC, no check will be performed. (No check will be performed if a set either does not contain any rule or it does not contain any active rule).

If there is a set defined for a responder BIC but none of the contained rules match, the file will be rejected.

Files with size 0 will always be rejected.

In the two latter cases the workflow will start with the IPS defined in the optional parameters `FACT_RECEPTION_ERROR_PATTERN` and `FACT_RECEPTION_ERROR_PATTERN_PREFIX` in section `[LCG<XXXX>.F201]` of the server configuration.

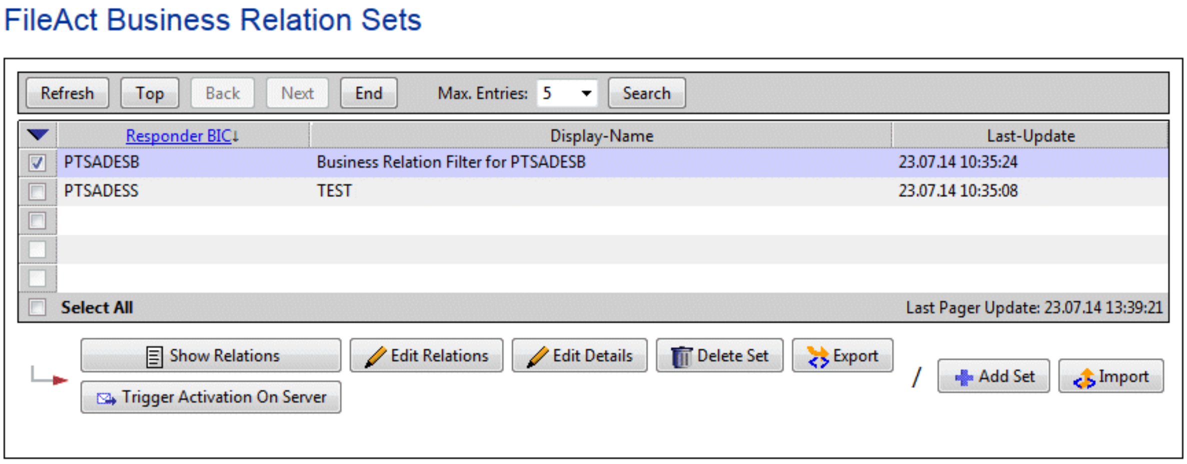
4.12.1 Create New Business Relation Set

As above mentioned a Business Relation is created as part of a Business Relation Set.

If a user shall be able to access (either view or create, modify, delete) a Business Relation for a certain Responder BIC, he must have the corresponding ACL entry assigned in the UPM record, see 3.2.4.

Then he can access the menu item **Administration / Messaging Interfaces / FileAct Business Relations**.

The **FileAct Business Relation Sets** page is displayed:



FileAct Business Relation Sets

Max. Entries: 5

	Responder BIC↓	Display-Name	Last-Update
☑	PTSADESB	Business Relation Filter for PTSADESB	23.07.14 10:35:24
□	PTSADESS	TEST	23.07.14 10:35:08
□			
□			
□			

□ Select All

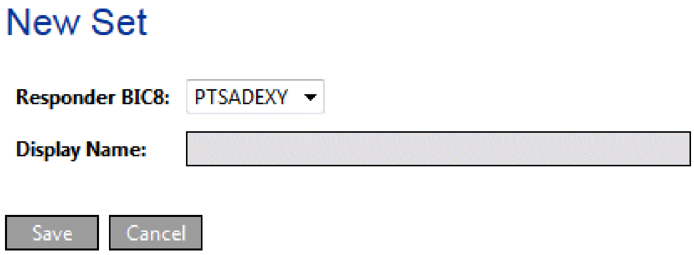
Last Pager Update: 23.07.14 13:39:21

The page contains the following buttons:

Button	Description
Show Relations	Lists the relations belonging to the selected set. The displayed page offers further buttons for viewing and editing the relations of this set.
Edit Relations	Lists the relations belonging to the selected set in Edit mode.
Edit Details	Displays the **Edit Set** page. The Display Name of the set can be modified.
Delete Set	Removes the selected set. You will be asked to confirm the deletion before the set is removed.
Export	Exports the selected set, i.e. displays the **Export Result** page explaining further steps for exporting.
Add Set	Creates a new Business Relation Set (see below). There must be corresponding ACL entry with admin access in your UPM record.
Import	Enables browsing for file to import.
Trigger Activation On Server	Activates the changes on the server. This button must be pressed after any changes in the Relation Set or Relation configuration, otherwise they will not take effect.

To create a new set, press **Add Set**.

The **New Set** page is displayed. However, only if there are ACL entries that have not been referenced yet in your Business Relations ACL UPM record.

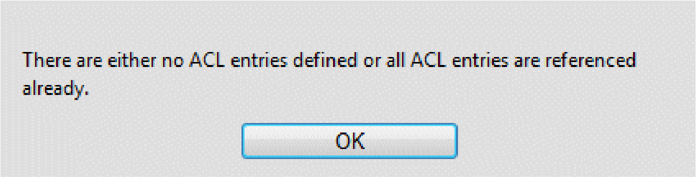


New Set

Responder BIC8: PTSADEXY

Display Name:

Otherwise you will get the following warning:



There are either no ACL entries defined or all ACL entries are referenced already.

Select the **Responder BIC8** from the list. The list shows only BICs to which you have access and that have not been referenced yet.

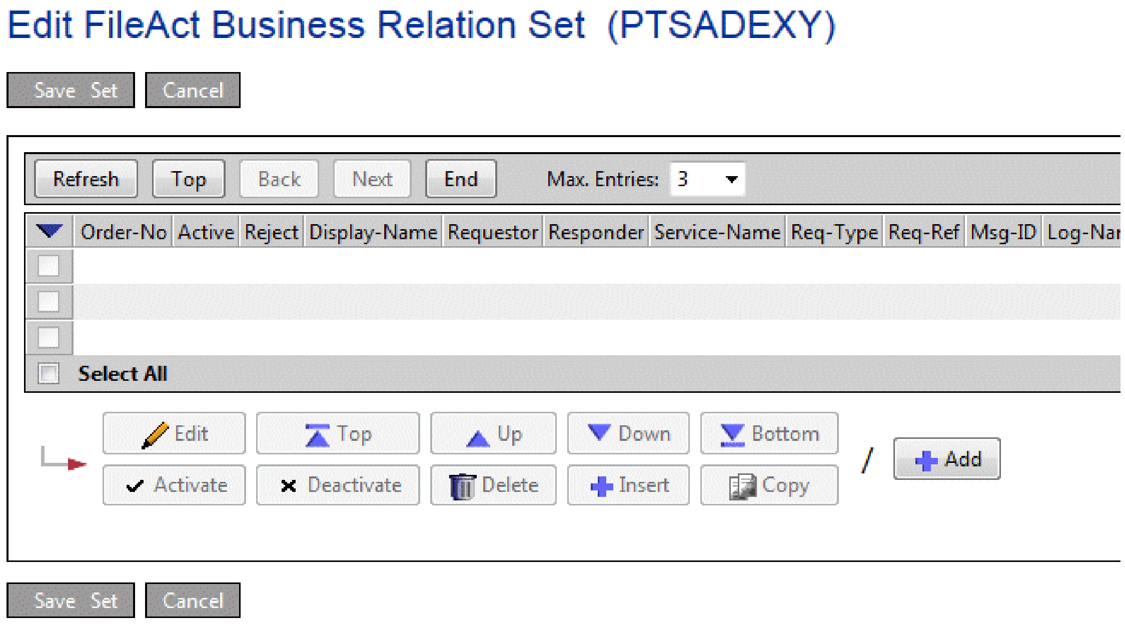
Enter a **Display Name** and press **Save**.

4.12.2 Create New Business Relation

Select the Business Relation Set to which you want to add a Business Relation.

Press **Edit Relations**.

The **Edit FileAct Business Relation Set** page is displayed:



Edit FileAct Business Relation Set (PTSADEXY)

Press the **Add** button.

The **New FileAct Business Relation** page is displayed.

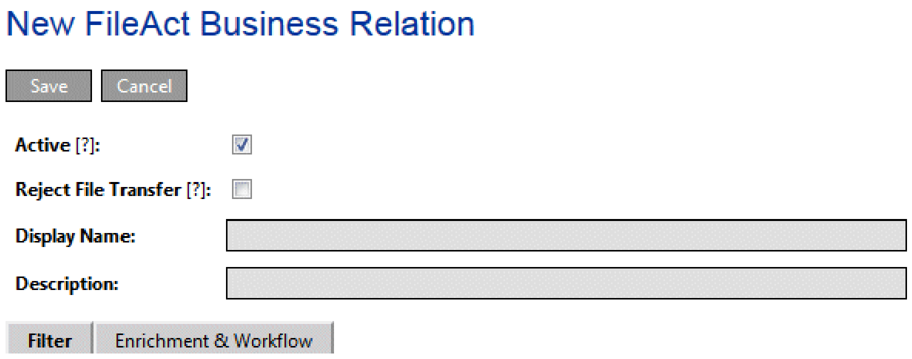
The page contains the following buttons:

Button	Description
Edit	Displays the **Edit FileAct Business Relation** page.
Top, Up, Down, Bottom	Moves a selected relation accordingly.
Activate	Turns the **Active** flag on, i.e. the relation will be used by the Messaging Interface.
Deactivate	Turns the **Active** flag off, i.e. the relation will not be used by the Messaging Interface.
Delete	Removes the selected relation. You will be asked to confirm the deletion before the relation is removed.
Insert	Displays the **New FileAct Business Relation** page and inserts the created relation before the selected one.
Copy	Copies the selected relation and inserts it to the end of the list.
Add	Displays the **New FileAct Business Relation** page and inserts the created relation to the end of the list.

A FileAct Business Relation can be divided in three parts. These parts (sections) will be described below:

4.12.2.1 General Section

This section contains the **Display Name** and a **Description** of the Business Relation.



New FileAct Business Relation

Active [?]: ☑

Reject File Transfer [?]: □

Display Name:

Description:

The **Active** flag determines whether the relation shall be used by the Messaging Interface or not.

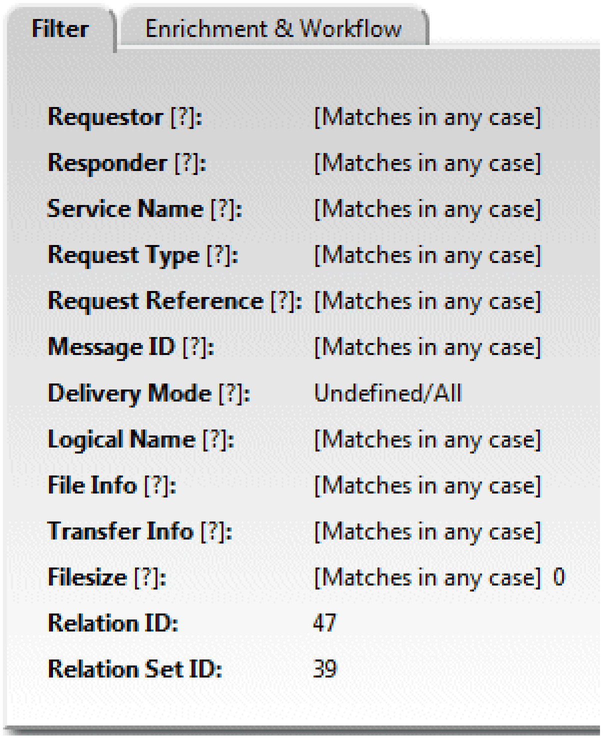
There are two different approaches to specifying Business Relation rules, either you specify rules for messages (files) that shall be accepted in case of matching or you specify rules for messages (files) that shall be rejected. If the **Reject File Transfer** flag is activated, the file will be rejected if it matches rule(s) specified in the **Filter** section.

When you activate the **Reject File Transfer** flag, an additional field will be visible, the **Reject Description** field. In this field you may enter a reject description that will be inserted into the **Response/Non Delivery Notification sent back to the sender** of the file.

The information in **Description** field is for internal information and will not be sent with any notifications.

4.12.2.2 Filter Section

In this section you specify data that will be compared against message data.



Requestor [?]:	[Matches in any case]
Responder [?]:	[Matches in any case]
Service Name [?]:	[Matches in any case]
Request Type [?]:	[Matches in any case]
Request Reference [?]:	[Matches in any case]
Message ID [?]:	[Matches in any case]
Delivery Mode [?]:	Undefined/All
Logical Name [?]:	[Matches in any case]
File Info [?]:	[Matches in any case]
Transfer Info [?]:	[Matches in any case]
Filesize [?]:	[Matches in any case] 0
Relation ID:	47
Relation Set ID:	39

The following fields can be used:

- Requestor
- Responder
- Request Reference
- Logical Filename
- File Info
- File Description
- Transfer Info
- Transfer Description
- Service Name
- Request Type

The following match operators can be used for the fields above:

- Equal / Equal Not
- Contains / Contains Not
- Ends With / Ends Not With
- Begins With / Begins Not With
- Regular Expr. Match
- Match in any case

The default operator is **Match in any case** when a new relation is created and will always match.

With **Regular Expr. Match** it is possible to specify regular expressions. Only a very limited subset of the known regex capabilities (defined in Posix 1003.2, regex) are supported.

Unsupported are e.g.:
branches separated by |
bounds defined by {n} or {n,m...}

Usage examples:
[a] matches char 'a'
[a-z0-9] matches all digits and all lowercase letters
[^A-Z] matches all but uppercase letters
[]][matches ']' and '[' (']' must be the first one)
[-0-9] matches all digits plus '-'
. matches any character
c? matches 0 or 1 occurrence of c (., char or [])
c* matches 0 or more occurrences of c (., char or [])
c+ matches 1 or more occurrences of c (., char or [])
^ as first char binds pattern to the beginning /
$ as last char binds pattern to the end
{ } used to define sub-expressions for replacement /

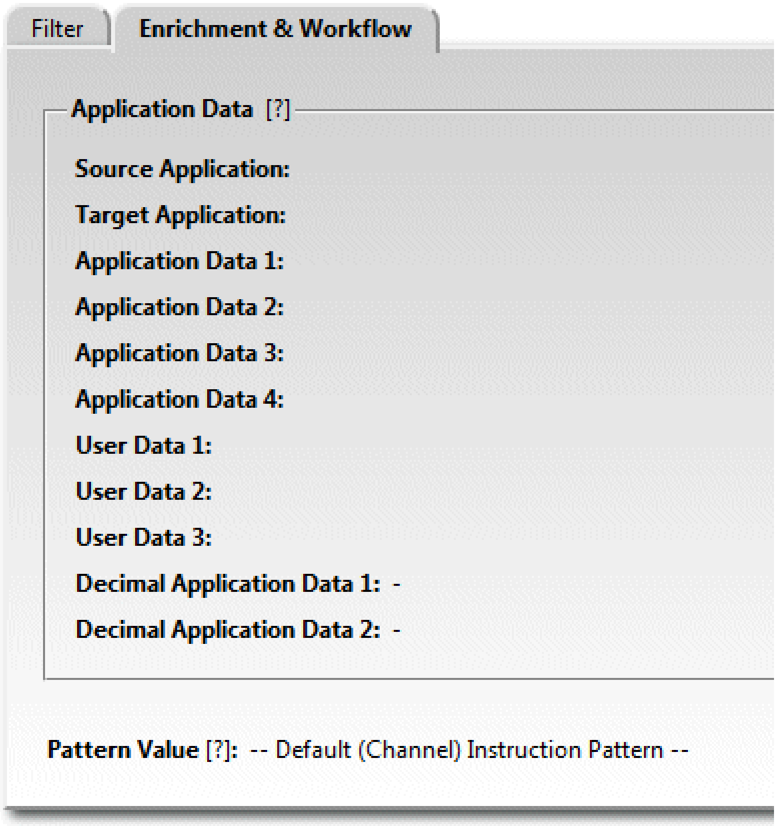
Delivery Mode: SnF or RealTime

Filesize: The file size in MiB (MebiBytes).

For the Filesize three operators are available: **Less or Equal**, **Greater or Equal**, **Match in any case**. The default operator is **Match in any case**. Files with size 0 will always be rejected.

4.12.2.3 Enrichment & Workflow Section:

In this section a workflow pattern can be set for the received message. It is also possible to set ApplicationData of the Generic Attributes.



Source Application:

Target Application:

Application Data 1:

Application Data 2:

Application Data 3:

Application Data 4:

User Data 1:

User Data 2:

User Data 3:

Decimal Application Data 1: -

Decimal Application Data 2: -

Pattern Value [?]: -- Default (Channel) Instruction Pattern --

The following ApplicationData fields can be set:

- SourceApplication
- TargetApplication
- ApplicationData1-4
- UserData1-3
- MiniApplicationData1 – 2
- ShortApplicationData1 - 4
- LargeApplicationData1 – 2
- NumericApplicationData1- 2
- DecimalApplicationData1 - 2, (delimiter '.')

The **Pattern Value** parameter specifies the Instruction Pattern with which the message processing shall continue.

If no Pattern is specified the Default Channel Pattern will be taken. This Default Pattern depends on whether the **Reject File Transfer** flag has been set or not.

If the Reject flag is set, the Default (Channel) Pattern is the IPS specified in the parameters `FACT_RECEPTION_ERROR_PATTERN` and `FACT_RECEPTION_ERROR_PATTERN_PREFIX` in section `[LCG<XXXX>.F201]` of the server configuration.

If the Reject flag is NOT set, the Default (Channel) Pattern is the IPS specified in the parameters `FACT_OUTPUT_PATTERN` and `FACT_OUTPUT_PATTERN_PREFIX`) in section `[LCG<XXXX>.F201]` of the server configuration.

After you have finished creating / modifying the Business Relation, press **Save**.

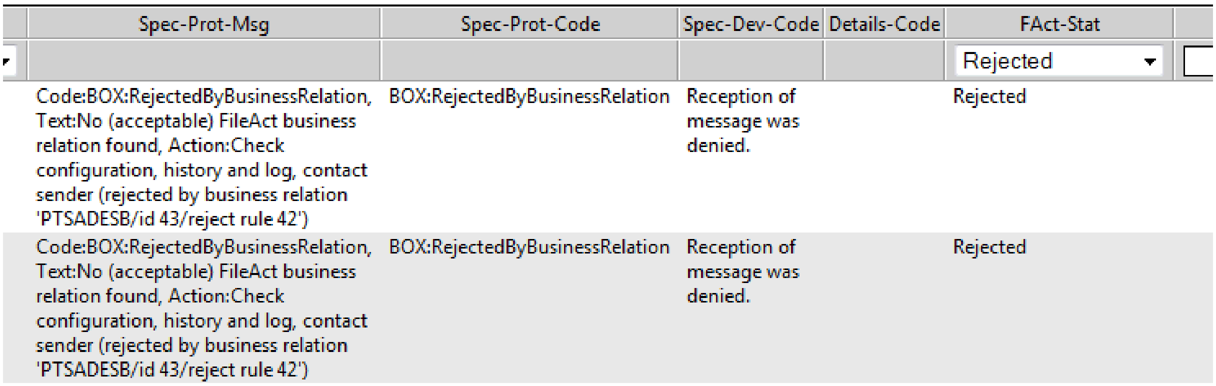
The **Edit Business Relation Set** page is displayed.

Press **Save Set**.

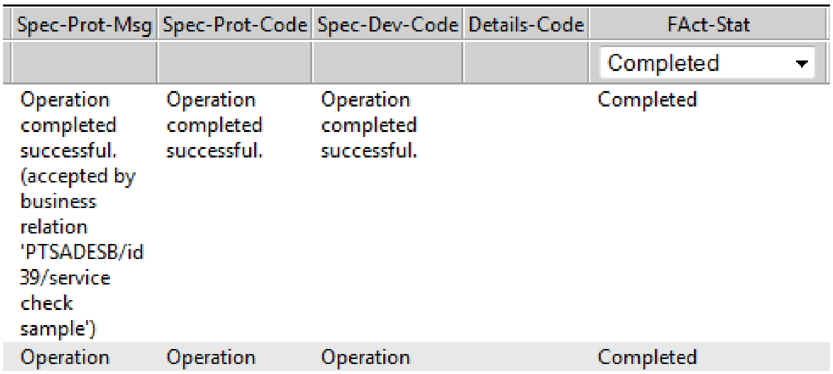
The **FileAct Business Relation Sets** page is displayed.

Press **Trigger Activation On Server** to have the changes take effect.

The result of the Business Relation check kan be seen in the Spec-Prot-Msg column of the Output Journals and the relevant Application Queues:



Spec-Prot-Msg	Spec-Prot-Code	Spec-Dev-Code	Details-Code	FAct-Stat
				Rejected
Code:BOX:RejectedByBusinessRelation, Text:No (acceptable) FileAct business relation found, Action:Check configuration, history and log, contact sender (rejected by business relation 'PTSADESB/id 43/reject rule 42')	BOX:RejectedByBusinessRelation	Reception of message was denied.		Rejected
Code:BOX:RejectedByBusinessRelation, Text:No (acceptable) FileAct business relation found, Action:Check configuration, history and log, contact sender (rejected by business relation 'PTSADESB/id 43/reject rule 42')	BOX:RejectedByBusinessRelation	Reception of message was denied.		Rejected



Spec-Prot-Msg	Spec-Prot-Code	Spec-Dev-Code	Details-Code	FAct-Stat
				Completed
Operation completed successful. (accepted by business relation 'PTSADESB/id 39/service check sample')	Operation completed successful.	Operation completed successful.		Completed
Operation	Operation	Operation		Completed

4.13 T2S Business Signature Handling

BOX supports the handling of business signatures.

To calculate business signatures for SWIFT Input FileAct messages the BusinessSignatureDN (in the T2SParameterList in the PutFileTransferParameters) must be set.

This can be done either directly by the backend application or with a submission profile.

In this case the Messaging Interface calculates digests of the file and prepares a digital signature. This prepared signature is sent to the SAG, completed and then inserted into the business file header.

The verification of the business signature for received SWIFT output messages is controlled by the optional configuration parameter BUSINESS_SIGNATURE_SERVICE_LIST in section [<channel_name>.CONNXX]. It contains a list of (endings of) services for which business signature verification is required. For example, BUSINESS_SIGNATURE_SERVICE_LIST !x requires business signatures for all test and training services. Default value is an empty list, meaning no business signature verification.

If business signature verification is required, the Messaging Interface extracts the business signature from business file header, verifies the digests and sends the signature to the SAG for verification.

If the business signature could be verified, the BusinessSignatureVerificationResult in the Generic Attributes is set to SUCCESS.

If the verification fails or if signature verification is required but no signature is available, the BusinessSignatureVerificationResult as well as the reception result code of the origination report are set to BUSINESS_AUTHENTICATION_FAILURE and the message is sent to the authentication failure label.

If no verification is required, the BusinessSignatureVerificationResult in the Generic Attributes is set to NOTAPPLICABLE.

If business signature verification is required, the business signature verification result is stored in the BOX Message XML. For FileAct also the signature itself is stored in the BOX Message XML.

For SWIFT Input FileAct messages the BusinessSignatureVerificationResult in the Generic Attributes is always set to NOTAPPLICABLE.

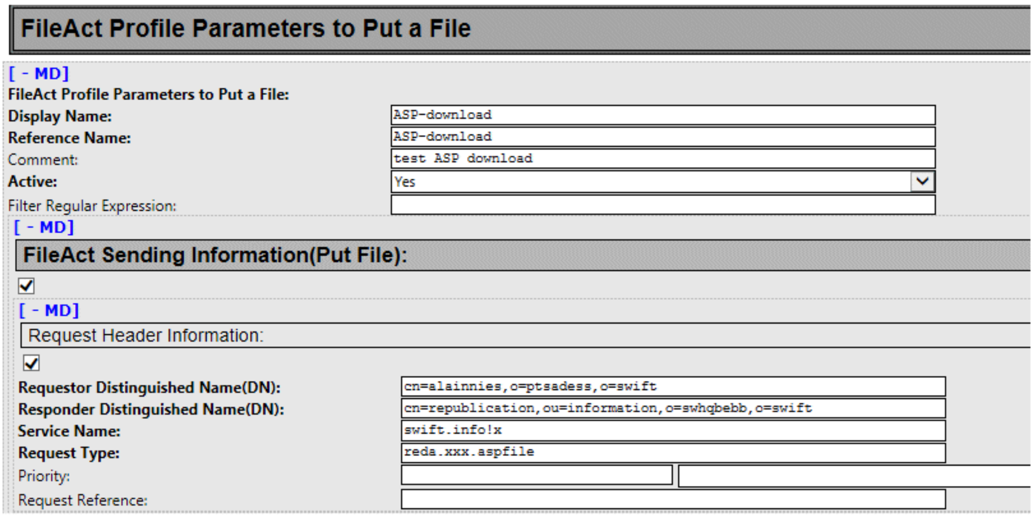
4.14 ITB ASP over FileAct Sparring Partner

There is a Sparring Partner that simulates the distribution of the Application Service Profiles for the Integration Test Bed (ITB ASP) over FileAct.

In order to request a push of the (latest) ITB ASP file, you must send a dummy file with the following parameters in the request:

Parameter	Parameter Value
Service :	swift.info!x Make sure that your actual ASP includes the service "swift.info!x"
Requestor DN	DN that has been specified during the subscription
Responder DN	cn=republication,ou=information,o=swhqbebb,o=swift
Request type	reda.xxx.aspfile
TransferInfo	Answer=ASP

Create a FileAct Profile with the parameters listed above.



FileAct Profile Parameters to Put a File

[- MD]

FileAct Profile Parameters to Put a File:

Display Name:	ASP-download
Reference Name:	ASP-download
Comment:	test ASP download
Active:	Yes
Filter Regular Expression:	

[- MD]

FileAct Sending Information(Put File):

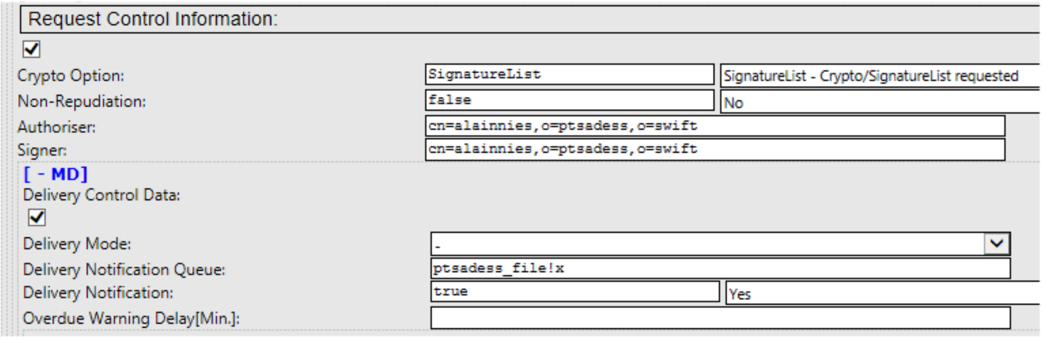
☑

[- MD]

Request Header Information:

☑

Requestor Distinguished Name(DN):	cn=alainnies,o=ptsadess,o=swift
Responder Distinguished Name(DN):	cn=republication,ou=information,o=swhqbebb,o=swift
Service Name:	swift.info!x
Request Type:	reda.xxx.aspfile
Priority:	
Request Reference:	



Request Control Information:

☑

Crypto Option:	SignatureList	SignatureList - Crypto/SignatureList requested
Non-Repudiation:	false	No
Authoriser:	cn=alainnies,o=ptsadess,o=swift	
Signer:	cn=alainnies,o=ptsadess,o=swift	

[- MD]

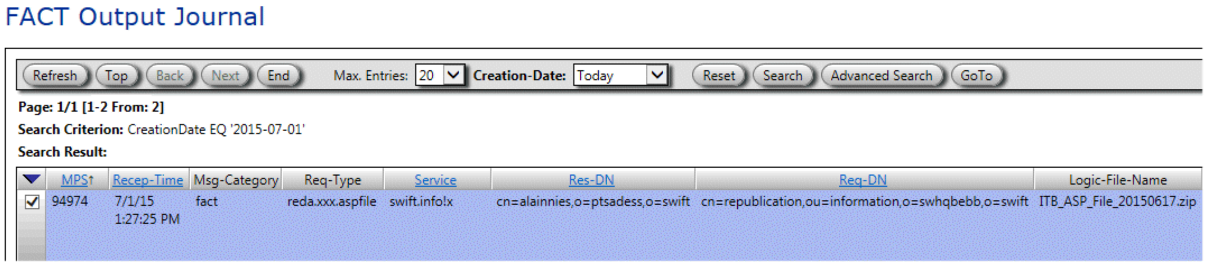
Delivery Control Data:

☑

Delivery Mode:	-	
Delivery Notification Queue:	ptsadess_file!x	
Delivery Notification:	true	Yes
Overdue Warning Delay[Min.]:		

Send a dummy file with this Profile.

The latest ITB ASP file (filename is **ITB_ASP_File_yyyymmdd.zip**) will be delivered using the information provided by the FileAct Profile parameters:



FACT Output Journal

Max. Entries: 20 **Creation-Date:** Today

Page: 1/1 [1-2 From: 2]

Search Criterion: CreationDate EQ '2015-07-01'

Search Result:

▼	MPS↑	Recep-Time	Msg-Category	Req-Type	Service	Res-DN	Req-DN	Logic-File-Name
☑	94974	7/1/15 1:27:25 PM	fact	reda.xxx.aspfile	swift.info!x	cn=alainnies,o=ptsadess,o=swift	cn=republication,ou=information,o=swhqbebb,o=swift	ITB_ASP_File_20150617.zip

The ASP will be received in the queue **ptsadess_file!x** (this is the default queue for processing incoming SWIFTNet FileAct Store and Forward traffic as specified in the SWIFTNet Service Subscription for the service swift.info!x).

The file is signed and authenticated and Delivery Notification is requested.

5 BOX / Backend Application Connection

The interface used by BOX in order to enable backend applications to send and to receive files through SWIFT FileAct is an MQ Gateway with a InterAct/FileAct Backend Application plug-in.

- When an application wants to send a file via SWIFT it forwards the file transfer data and file data as an MQ message to a queue which is read by BOX.
- When an application wants to process a file received from SWIFT it reads an MQ message from a queue into which BOX writes.

The format of the data exchanged between BOX and the backend application depends on the format that the backend application sends / expects to receive (with or without RFH2 Header).

5.1 SWIFT Input

The format of messages from a backend application can be either MQMD + file data or MQMD + RFH2 Header + file data.

If the backend application provides only MQMD and file data, the BOX InterAct/FileAct Backend Application plug-in processes the data and generates a message in BOX XML format (MPS).

For backend applications that do not provide RFH2 Header data, the BOX configuration parameter **RFH2_MODE** in section **[LCG<XXX>.F002.IAFABA_PLUGIN]** must be set to NO.

If the backend application provides an RFH2 Header, the InterAct/FileAct Backend Application plug-in creates an internal XML data structure that contains all name values of the message as children of the root node (canonical RFH2.xml).

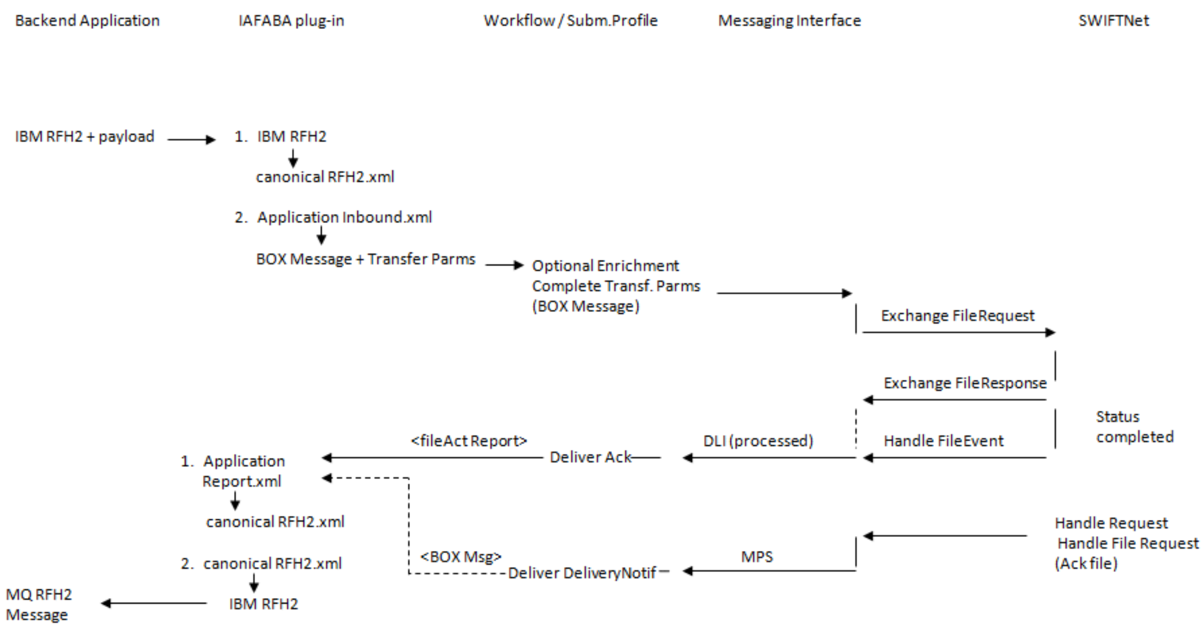
This internal XML is of the form:

```
<RFH2>
  <namevalue1> ... </namevalue1>
  <namevalue2> ... </namevalue12>
  <namevalue3> ... </namevalue3>
  ...
</RFH2>
```

Then the data from this internal XML is transformed into a message in BOX XML format (MPS) by means of XSLT.

The message can then (optionally) be enriched with data from a Submission Profile (see 4.11).

The figure below shows an overview of the data exchange between a back-end application providing RFH2 Header data and SWIFTNet.



5.2 SWIFT Output

The format of SWIFT Output messages that shall be routed to a backend application is either MQMD + file data or MQMD + RFH2 Header + file data.

The format to be used depends on the backend application, i.e. on the format that the application expects.

If the backend application expects only MQMD and file data, the InterAct/FileAct Backend Application plug-in processes the BOX format message and hands it over to the backend application in the expected format.

For backend applications that cannot handle RFH2 Header data, the BOX configuration parameter **RFH2_MODE** in section **[LCG<XXX>.F002.IAFABA_PLUGIN]** must be set to **NO**.

If the backend application shall receive a message with RFH2 Header, the message in BOX XML format is transformed into an internal XML data structure by means of XSLT.

This internal XML is of the form:

```
<RFH2>
  <namevalue1> ... </namevalue1>
  <namevalue2> ... </namevalue12>
  <namevalue3> ... </namevalue3>
  ...
</RFH2>
```

It contains the name values of the RFH2 message to be created as children of the root node (canonical RFH2.xml). The InterAct/FileAct Backend Application plug-in then creates the RFH2 message from this internal XML (see figure above).

For backend applications that can handle RFH2 Header data, the BOX configuration parameter **RFH2_MODE** in section **[LCG<XXX>.F002.IAFABA_PLUGIN]** must be set to **YES**.

5.2.1 Handing Over Data to the Backend Application

5.2.1.1 Application Identity Data

One way of handing over data to the Backend Application is to use the **Application Identity Data** parameter in the MQ message descriptor (MQMD).

The BOX configuration parameter **OUTPUT_MQ_APPLIDENTITY** in section **[LCG<XXX>.F002.IAFABA_PLUGIN]** specifies how Application Identity Data within the MQMD Header is set when FileAct output files are routed through the InterAct/FileAct Backend Application plug-in.

The configured string will be copied to Application Identity Data.

This way it is possible to use the Application Identity Data field (maximum length of 32 byte) for embedding some information accompanying the received file.

The following replacements strings can be used:

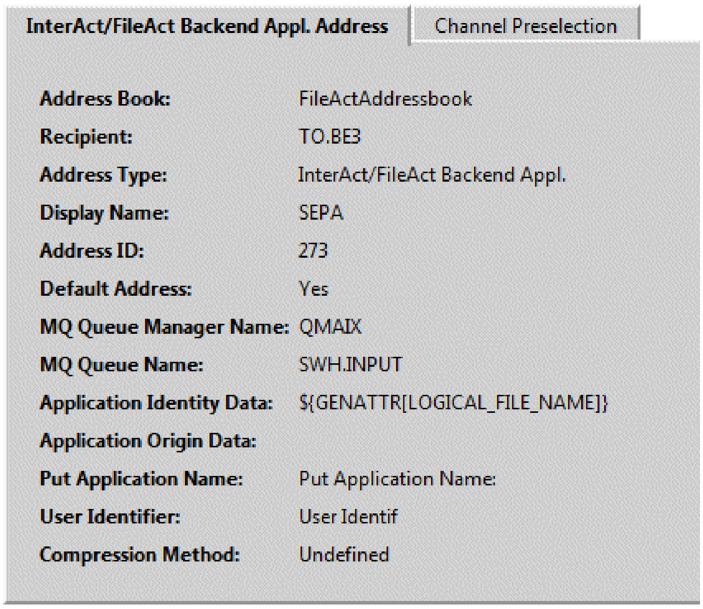
Replacement String	Description
${GENATTR[REQUESTOR_DN]}	String is replaced with the requestor DN of the received file
${GENATTR[RESPONDER_DN]}	String is replaced with the responder DN of the received file
${GENATTR[SERVICE_NAME]}	String is replaced with the Service Name of the received file
${GENATTR[REQUEST_TYPE]}	String is replaced with the Request Type of the received file
${GENATTR[MESSAGE_TYPE]}	String is replaced with the Message Type of the received file
${GENATTR[CATEGORY]}	String is replaced with the message category of the received file
${GENATTR[LOGICAL_FILE_NAME]}	String is replaced with the LogicalFilename of the received file.

5.2.1.2 From Address Book Entry

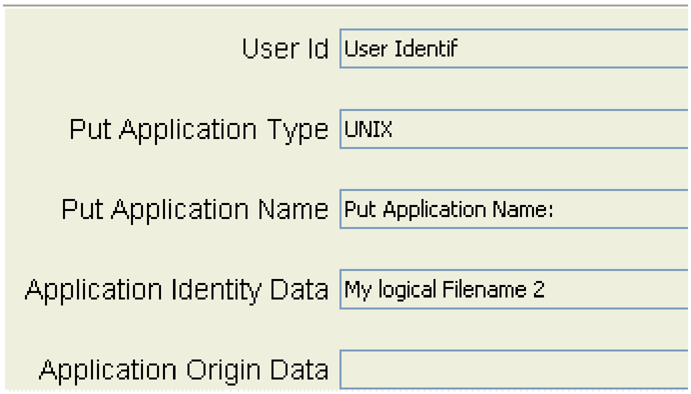
Another way of handing over parameter values is to write data from the address book entry of the backend application into the MQMD Header fields.

In this case the information in the fields **Application Identity Data**, **Application Origin Data**, **Put Application Name** and **User Identifier** can be written into the header fields.

The pictures below show the address book entry for a FileAct backend application and the corresponding information in the MQMD Header:



Address Book:	FileActAddressbook
Recipient:	TO.BE3
Address Type:	InterAct/FileAct Backend Appl.
Display Name:	SEPA
Address ID:	273
Default Address:	Yes
MQ Queue Manager Name:	QMAIX
MQ Queue Name:	SWH.INPUT
Application Identity Data:	${GENATTR[LOGICAL_FILE_NAME]}
Application Origin Data:	
Put Application Name:	Put Application Name:
User Identifier:	User Identif
Compression Method:	Undefined



User Id	User Identif
Put Application Type	UNIX
Put Application Name	Put Application Name:
Application Identity Data	My logical Filename 2
Application Origin Data	

The **Application Identity Data** parameter can be used as described above and the replacement strings are the same ones. If the configuration parameter **OUTPUT_MQ_APPLIDENTITY** in section **[LCG<XXX>.F002.IAFATBA_PLUGIN]** has been set, the information will be written from the parameter value.

If **Application Origin Data** is set in the recipient address but **Put Application Name** is left empty, the gateway writes "MPO SERVER" or "MPO CGTW" into the **Put Application Name** field in the MQMD Header, depending on whether the gateway runs in the server or as an external process.

5.2.1.3 Message Enrichment

The IAFABA plugin also allows sending MPS related data (e.g. MPS ID, reference data, generic attributes) to the backend application.

The BOXMessage node (or XXXReport node) is enriched with a **MessageInformation** node containing the MPS data (e.g. **BOXMessage/MessageInformation** or **XXXReport/MessageInformation**).

The '**MesageInformation**' node can contain the following subnodes:

5.2.1.3.1 The 'ProcessingSequenceData' Node:

This node contains nodes '**ProcessingSequenceID'** and '**CreationTime**', also a folder '**Reference**' containg the following MPS reference data:

'DisplayReference', 'ExternalTextTypeInfo', 'ExternalTextReference1', 'ExternalTextReference2', 'ExternalTextReference3', 'ExternalTextReference4', 'ExternalNumTypeInfo', 'ExternalNumReference1', 'ExternalNumReference2', 'ExternalNumReference3'

5.2.1.3.2 The 'ApplicationAtttributes' Node

This folder contains the following general application attribute data:

'**CreationDate**', '**DisplayReference**', '**ApplicationstatusValue**', '**ApplicationQueueID**'.

The subfolder '**ApplicationData**' contains the following data:

'SourceApplication', 'TargetApplication',
'Applicationdata1', 'ApplicationData2', ApplicationData3', 'ApplicationData4',
'UserData1', 'UserData2', 'UserData3',
'MiniApplicationData1', 'MiniApplicatinData2',
'ShortApplicationData1', 'ShortApplicationData2', 'ShortApplicationData3', 'ShortapplicationData4',
'LargeApplicationData1', 'LargeApplicationData2',
'NumericApplicationData1', 'NumericApplicationData2',
'ApplicationTimestamp1', 'ApplicationTimestamp2',
'ApplicationDate1', 'ApplicationDate2',
'DecimalApplicationData1', 'DecimalApplicationData2'

The subfolder '**AttributeSet/GenericattributeSet**' contains the following data whereby the contained data is ApplicationAttributeSet Tyoe-dependent.

For ApplicationAtttributeSet Type FACT:

'Type' (Text representation of MP_APL_GENATTRS_TYPE_XX)
'SubType' (Text representation of MP_APL_GENATTRS_SUBTYPE_XX)
'ApplicationDefinedType'
'ApplicationDefinedDescription'
'PDMInformation'
'LastApplicationQueueDisplayname'
'PDEInformation'
'LogicalFilename'
'RequestorDN'
'OperatorComment'
'ResponderDN'
'LastActingUser'
'RequestType'
'Category'
'MessageID'
'ServiceName'
'LastTransferStatus'
'OutputSessionID'
'SubmissionResultCode'
'MessageType'
'RequestReference'
'ReportSource'

'FileSize'
'OutputSequenceNumber'
'PDFlag'
'FileStatusValue' (Text representation of MP_FACTSATAT_VAL_XX)
'BusinessSignatureVerificationResult' (Text reprensentation of REPRES_XX)
'CreationTime'
'LastActingUserTime'

For ApplicationAtttributeSet Type MX:

'Type' (Text representation of MP_APL_GENATTRS_TYPE_XX)
'SubType' (Text representation of MP_APL_GENATTRS_SUBTYPE_XX)
'ApplicationDefinedType'
'ApplicationDefinedDescription'
'PDMInformation'
'LastApplicationQueueDisplayname'
'PDEInformation'
'RequestorDN'
'OperatorComment'
'ResponderDN'
'LastActingUser'
'RequestType'
'MessageSet'
'MesssageID'
'ServiceName'
'InputSessionID'
'OutputSessionID'
'SubmissionResultCode'
'MessageType'
'RequestReference'
'ReportSource'
'RetrievalDirection'
'RetrievalSnFReference'
'RetrievedMessageID'
'InputSequenceNumber'
'OutputSequenceNumber'
'PDFlag'
'BusinessSignatureVerificationResult' (Text representation of REPRES_XX)
'RetrievalSequenceNo'
'CreationTime'
'LastActingUserTime'
'RetrievedCreationTime'

For ApplicationAtttributeSet Type T2SMSG, T2SMSG_SWIFT, T2SMSG_SIA:

'Type' (Text representation of MP_APL_GENATTRS_TYPE_XX)
'SubType' (Text representation of MP_APL_GENATTRS_SUBTYPE_XX)
'ApplicationDefinedType'
'ApplicationDefinedDescription'
'PDMInformation'
'LastApplicationQueueDisplayname'
'PDEInformation'
'T2SActorMessageID'
'T2SMessageID'
'Sender'
'OperatorComment'
'Receiver'
'LastActingUser'
'RequestType'
'MessageSet'

'MessageID'
'TechnicalServiceID'
'InputSessionID'
'OutputSessionID'
'SubmissionResultCode'
'MessageType
'RequestReference'
''ReportSource'
'BusinessMessageID'
'RelatedBusinessMessageID
'RetrievalDirection'
'RetrievalSnFReference'
'RetrievedMessageID'
'InputSequenceNumber'
'OutputSequenceNumber'
'PDFlag
'BusinessSignatureVerificationResult' (Text reprensentation of REPRES_XX)
'RetrievalSequenceNo'
'CreationTime'
'LastActingUserTime'
'RetrievedCreationTime'

For ApplicationAtttributeSet Type T2SFILE, T2SFILE_SWIFT, T2SFILE_SIA:

'Type' (Text representation of MP_APL_GENATTRS_TYPE_XX)
'SubType' (Text representation of MP_APL_GENATTRS_SUBTYPE_XX)
'ApplicationDefinedType'
'ApplicationDefinedDescription'
'PDMInformation'
'LastApplicationQueueDisplayname'
'PDEInformation'
'T2SActorMessageID'
'T2SMessageID'
'Sender'
'OperatorComment'
'Receiver'
'LastActingUser'
'RequestType'
'Category'
'MessageID'
'TechnicalServiceID'
'LastTransferStatus'
'OutputSessionID'
'SubmissionResultCode'
'MessageType'
'RequestReference'
'ReportSource'
'FileSize'
'OutputSequenceNumber'
'PDFlag'
'FileStatusValue' (Text representation of MP_FACTSATAT_VAL_XX)
'BusinessSignatureVerificationResult' (Text representation of REPRES_XX)

For ApplicationAtttributeSet Type SIA_FTS:

'Type' (Text representation of MP_APL_GENATTRS_TYPE_XX)
'SubType' (Text representation of MP_APL_GENATTRS_SUBTYPE_XX)
'ApplicationDefinedType'
'ApplicationDefinedDescription'
'LastApplicationQueueDisplayname'

'VirtualFileName'
'ApplicationData'
'SenderBA'
'OperatorComment'
'ReceiverBA'
'LastActingUser'
'Category'
'TruncatedApplicationData'
'StatusText'
'StatusCodeText'
'MessageType'
'ReportSource'
'FileSize'
'FileStatusValue' (Text representation of MP_FTSSTAT_VAL_XX)
'StatusCode'
'LastActingUserTime'

5.2.1.3.3 The 'RefOrigContentData' Node

This node is optional and only if referenced content is available it contains 'BOXMessage' of referenced MPS.

If there is no referenced original, there is no RefOrigContentData node.

In case of received delivery notifications the referenced original content is written to the **EnhancedExportData/RefOrigContentData** node if the ENHANCED_EXPORT_DATA_MODE config parameter is set to YES.

5.3 Configuration of InterAct/FileAct Backend Application Plug-in

For the connection between BOX and a backend application a BOX MQ Gateway with an InterAct/FileAct Backend Application Plug-in (IAFABA Plug-in) must be configured.

Below you can see an excerpt from the Server configuration file (mposerver.cfg) with a sample configuration:

```
;-----------------------------------------------------------
;
;       Server LCG 011: IAFABA
;
;-----------------------------------------------------------
[LCGIAFABA]
  CGTW_HOST                          ; P:11,1  ; <Protocol>:ModuleID,LCG-Number
  CHANNEL_TYPE                       IAFA-BA-INTERFACE
  APPLICATION_GROUP_NAME             FACTBA
  DEFAULT_DELIVERY_COMPOSITION       0x012101
  SUPPORTED_ADDRESSTYPES             IAFABA

[LCGIAFABA.PEXA]
  IMPORT_CHECK_CYCLE              5
  DEVICE_TYPE                        0xF002
  CREATOR_PREFIX                     demo
  DEFAULT_CREATOR                    DemoBank
  DEFAULT_OWNER                      DemoBank
  DEFAULT_IPS_SHORTLABEL             SendFACTFromBA
  DEFAULT_MPS_INITMODE               2     ; 1 - Instantiated, 2 - Pattern
  DELIVERY_MONITOR                   YES
  MONITOR_CARRIER_DELIVERY           NO
  STORAGE_PERIOD                     24

[LCGIAFABA.F002]
  PLUGIN_LIBRARY_NAME                expgi_factba
  LOCAL_QUEUE_MANAGER                QM_ichh2wk
  DEFAULT_OUTBOUND_QUEUE_MANAGER     QM_ichh2wk
  DEFAULT_OUTBOUND_QUEUE             TO.FACTBA
  INBOUND_QUEUE                      FROM.FACTBA
  DEFAULT_REPLY_QUEUE_MANAGER        QM_ichh2wk
  DEFAULT_REPLY_QUEUE                FROM.FACTBA
  TRASH_QUEUE_NAME                   MPO.TRASH.QUEUE
  DEFAULT_ACCOUTING_TOKEN            34343436363639
  DELIVERY_REPORT_GENERATION         1;4
                                           ;   0   // delivery report is submission report
                                           ;   1   // delivery report through COA
                                           ;   2   // delivery report through COD
                                           ;   3   // delivery report through PAN/NAN
                                           ;   4   // delivery report through reply
  MESSAGE_DUMP_LIMIT                 10000
  GENERATE_COMMAND_REPORT            NO

[LCGIAFABA.F002.IAFABA_PLUGIN]
 RFH2_MODE                           YES
 TARGET_MESSAGE_TYPE                 BOX-MESSAGE-FACT
 DEFAULT_TARGET_FILE_COMPRESSION     NONE
 REPORT_XSLT                         config/FACTBAReport.xslt
 OUTPUT_XSLT                         config/FACTBAOutput.xslt
 INPUT_XSLT                          config/FACTBAInput.xslt
 REMOVE_XML_PI                       YES
 ENHANCED_EXPORT_DATA_MODE           YES
```

The parameter ENHANCED_EXPORT_DATA_MODE is of importance if delivery notifications are routed to the backend application. If set to YES, the IAFABA plug-in sets the message type according to the message type of the received original message.

In order to allow access to the referenced original content data, the default XML format is:

```
<EnhancedExportData>
  <ContentData>
     BOXMessage XML of message to be exported
  </ContentData>
  <RefOrigContentData>
    BOXMessage XML of referenced original message (if available)
  </RefOrigContentData>
</EnhancedExportData>
```

If there is no referenced original, there is no RefOrigContentData node.

If the parameter ENHANCED_EXPORT_DATA_MODE is set to NO, the export data XML will be the normal BOXMessage XML.

The compression method specified by the parameter DEFAULT_TARGET_FILE_COMPRESSION will be overridden by the value specified in selection box Compression Method in the Address Book entry (Address Type: InterAct/FileAct Backend Application) for the recipient.

5.3.1 Message Enrichment

The parameter [LCGIAFABA.F002.IAFABA_PLUGIN] ORIGINATION_APPL_DATA_NODE defines the name of the node in BOXMessage/EnrichmentData that contains the data that is used for generating ApplicationData in the Generic Attributes of a generated MPS.

When a message with an RFH2 header is received the stylesheet defined by the parameter INPUT_XSLT can be used for creating this node and fill it with fixed data or data retrieved from the RFH2 header of the received message.

The data must be stored in the following subnodes:

SourceApplication, TargetApplication. ApplicationData1, ApplicationData2, ApplicationData3, ApplicationDate4, UserData1, UserData2, UserData3, MiniApplicationData1, MiniApplicationData2 ShortApplicationData1, ShortApplicationData2, ShortApplicationData3, ShortApplicationData4 LargeApplicationData1, LargeApplicationData2, NumericApplicationData1, NumericApplicationData2, DecimalApplicationData1, DecimalApplicationData2 (delimiter '.') ApplicationDate1, ApplicationDate2 (format YYYY-MM-DD) ApplicationTimestamp1, ApplicationTimestamp2 (format iso8601)

The values of these subnodes will be stored in the ApplicationData of the Generic Attributes of the MPS.

The node and its subnodes will be removed from the BOXMessage/EnrichmentData node after the data has been stored in the Generic Attributes.

The node and the subnodes may also be used for sending enrichment data to the backend application.

For more details on the configuration parameters, refer to the **BOX Configuration Guide**.

5.4 Message Grouping

In order to enable the transfer of huge files via Websphere MQ BOX FileAct supports MQ logical message groups.

The BOX configuration parameter **MAX_MSGLEN_IN_GROUP** in section **[LCGXXX.F002]** enables reading/writing logical message groups from/into Websphere MQ queues.

A value of 0 disables logical message group functionality, i. e. the FileAct Backend Application plug-in tries not to split big messages into logical messages building a group when putting a message into a queue. When reading a message from a queue and the MQGET call returns a 'Message in Group' or 'Last Message in Group' status those messages will be trashed and generates an exception MPS as it cannot be guaranteed that the logical message can be reconstructed correctly.

A value greater 0 specifies the (expected) maximum message size (in bytes) of a single message within a logical group. When reading messages from the queue this parameter is used for pre-allocating memory buffers used for gathering the logical message (see also parameter **MAX_EXP_GROUP_TOTAL_LEN**). When writing a message to a queue this size is the maximum size of a single message created within the logical message group.

The parameter **MAX_EXP_GROUP_TOTAL_LEN** is used for giving a hint for memory allocation when allocating a buffer big enough to contain a complete logical message. The default value for this parameter is calculated as 5 times the maximum size of a single message as specified in parameter **MAX_MSGLEN_IN_GROUP**.

Use this parameter to optimize temporary memory allocation. If used, it should be set to the expected maximum total size of logical message group.

If no logical message groups shall be used/allowed, this parameter is meaningless.

Also refer to the description of the configuration parameter **ALWAYS_LOGICAL_GROUPS** in the BOX Configuration Guide.

Note that message queues which receive logical message group should allow for Get-Message-Option 'MQGMO_LOGICAL_ORDER' which in turn requires MQ index type 'MQIT_GOUP_ID' on z/OS queue managers.

Handling of logical message groups follows the strategy described in 'Grouping logical messages' within the Websphere MQ documentation.

6 Message Flow

BOX FileAct is delivered with a sample message flow configuration which includes application queues with task buttons for different message processing steps, such as authorization, modification and repair actions. It also includes all required Instruction Pattern Sequences (IPS) controlling the message processing by e.g. tagging messages to the ‘correct’ application queue and IPS for routing SWIFT Output messages to a backend application.

6.1 Application Queues

The following table gives an overview of some typical BOX application queues and explains the usage of each queue:

Appl.Queue	Usage
FACT Message Entry	Application queue for creation of FileAct messages.
FACT Authorization	Authorization of FileAct messages (4-eyes principle).
FACT Message Modification	Application queue for modification of FileAct messages.
FACT ResponseWaiting	Application queue for messages that wait for response (e.g. ACK) before further processing is initiated.
FACT SWIFT NAK Repair	Application queue for repair of SWIFT Input messages that have received a NAK from SWIFT.
FACT Authentication Errors Input Messages	Application queue for SWIFT Input messages for which authentication failed, e.g. wrong digest.
FACT Authentication Errors Output Messages	Application queue for SWIFT Output messages for which authentication failed, e.g. wrong digest.
FACT Authorization Errors Input Messages	Application queue for SWIFT Input messages for which authorization failed.
FACT Authorization Errors Output Messages	Application queue for SWIFT Output messages for which authorization failed.
FACT Trash	Application queue to trash drafts, templates and other incomplete FileAct messages no longer needed.
FACT Internal Failure	Application queue to ‘move’ messages to in case of instruction failures (e.g. failures due to configuration failure/configuration testing), for all failure cases not specifically provided for.

Each of these queues includes queue-specific action buttons for triggering the next processing step.

6.2 Manual Message Entry

For the manual creation of messages there is a dedicated queue for message entry (FACT Message Entry) in which you can enter the message data into easy-to-handle entry forms.

The desired request type (Request to fetch file, Request to put file) can be selected and then the Submission Profile from a drop-down selection menu. Thereafter the message entry form for the respective Submission Profile is displayed.

Each form contains all required headers and data fields and mandatory fields have been highlighted. Before passing the message on to possibly required further authorization, it can be internally validated, i.e. checked in regard to correctness of the entered data. In case of errors the fields with erroneous values are highlighted and the respective validation error message is displayed.

6.2.1 File Compression

Due to performance reasons BOX client does not compress a file that is to be sent.
If the file is already compressed and the user knows it, he can select the '**This file is compressed**' checkbox in the File Input GUI. However, this is not mandatory as the BOX Client under all circumstances automatically tries to find out the compression type.

6.3 Authorization

The BOX configuration may include an authorization cycle that controls the need for messages to be authorized.

At user level you specify the application queues and the FileAct messages that a certain user may access as well as the tasks he is allowed to perform in each queue. These definitions are done in the user's UPM entry.

6.4 Modification

In a typical message flow configuration messages that could not be authorized are put into a dedicated modification queue, in which they can be viewed and opened for modification.

After completed modification and successful validation the message is again put into an authorization queue.

6.5 Repair

A typical configuration includes separate Application Queues for repairing of messages.
Such repair queues collect failed messages depending on the failure reason and offers the possibility to repair a message as well as to view the corresponding ACK/NAK report.

Depending on the configuration, BOX may analyze the ACK/NAK report of each message and - in case of failure - determine the failure reason. Depending on the found reason, the message is put into a corresponding repair queue, from which it can be opened and repaired accordingly.

6.6 SWIFT Input Messages from Other Applications

The typical BOX configuration includes the possibility to have SWIFT messages that have been created in backend applications to be passed on to SWIFT.

In case of failed processing the respective messages are collected in a corresponding queue for repair.

6.7 Routing of SWIFT OutPut Messages

The message flow configuration provides the possibility to route SWIFT Output messages to different backend applications depending on the received message data.

The message is analyzed by means of Analysis 1 in regard to SWIFT Message Type, and Service and then routed to the corresponding backend application.

Sample Analysis 1 Statement:

```
if (MessageType =="seev.006.001.04") {
   print( "MessageType =" + MessageType);
   select ABRECIP with (ABREC_DISPNAME == "TO.BE.MQ");
   setips FACT_RouteToABRecipients;
   return;
}

print( "service = " + service);

if ( service == "swift.test.rt.iafa!x") {
   select ABRECIP with (ABREC_DISPNAME == "TO.BE1");
   setips FACT_RouteToABRecipients;
   print( " BE1" );
   return;
}

if ( service == "swift.qualif.rt.iafa!x") {
   select ABRECIP with (ABREC_DISPNAME == "TO.BE2");
   setips FACT_RouteToABRecipients;
   print( "BE2" );
   return;
}

if ( service == "swift.test.sf.iafa!x") {
   select ABRECIP with (ABREC_DISPNAME == "TO.BE3");
   setips FACT_RouteToABRecipients;
   print( "BE3" );
   return;
}

found=getresultsetsize();
if( found <= 0 ) {
   print("No recipient found  " );
   setips FACT_ToInternalFailure ;
} else  {
   print("We found" + tostring(found) + " recipients ");
}
return;
```

7 Messaging Interface Configuration

The configuration of the Messaging Interface may be performed **either** in a configuration file (of the server) **or** via the GUI.

Whether the configuration shall be done via file or via Web-Client is specified with the configuration parameter CHANNEL_CONFIG in section [MPO_MGTW] in the MPO Messaging Interface configuration file (e.g.: mgtwfia1.cfg):

```
[MPO_MGTW]
   CHANNEL_CONFIG
```

This parameter defines where the configuration shall be retrieved from. The following values can be used:

@: Retrieve channel configuration from central server configuration repository, use standard naming convention to retrieve configuration file from server (mgtwxxxxxz3_lcg.cfg).

@<filename>: Retrieve configuration from central server configuration repository, use the supplied file name.

<pathname>: Read module task configuration locally, using the supplied path.

$DB:<ModuleName>: Retrieve configuration from Database (Configuration via Web-Client).
Example:
$DB:MI_FIA for the Module with the Module Name "
MI_FIA".

7.1 Configuration via Configuration File

The configuration of the FileAct Backend Application Gateway and of the Backend Application plug-in is done in the Server configuration file.

The parameter names and values are the same both in configuration files and in the GUI.
For detailed parameter descriptions please refer to the document BOX **Configuration Guide**.

7.2 Configuration via Web-Client

The configuration of **MI Modules**, **FileAct LCG**s, **FIA Channels**, **Sessions**, **In**- and **Output Channels** and **Channel Connections** is done via the Web-client.

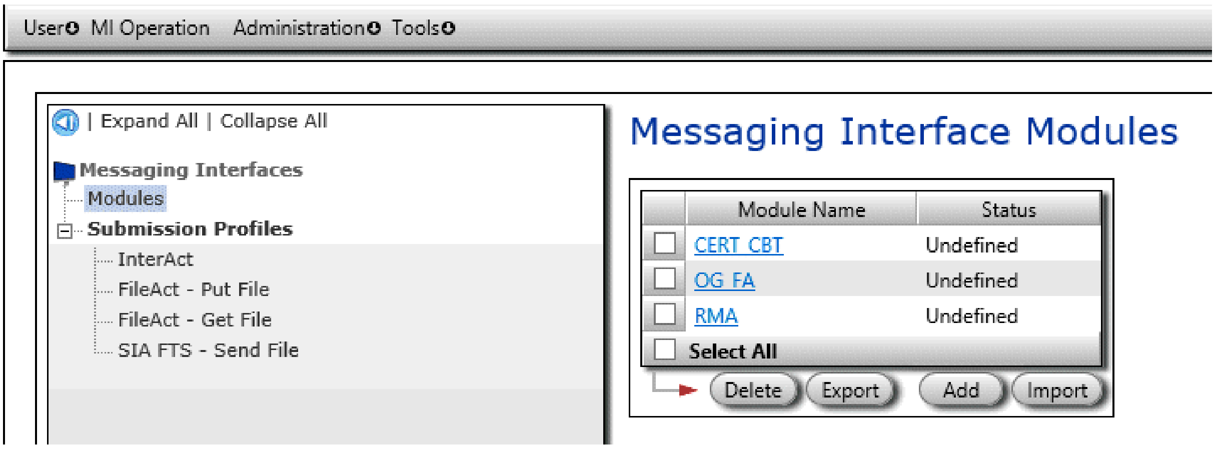
The Messaging Interface related configuration panels show all parameters that are relevant for the configuration.

The configuration is done on Enterprise level.

Select **Administration / Messaging Interfaces** from the main menu.

Select **Modules** from the tree-view.

The **Messaging Interface Modules** page is displayed.



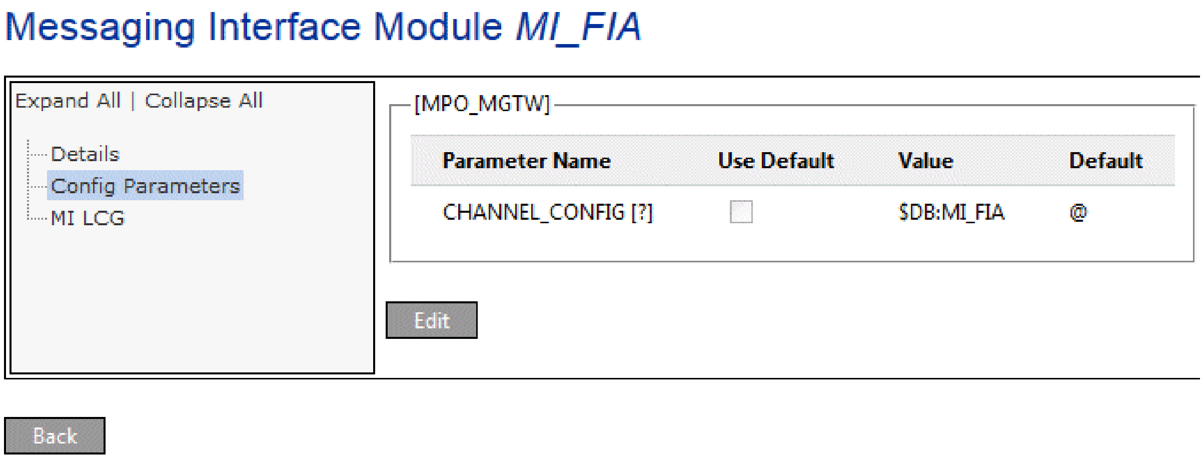
Messaging Interface Modules

	Module Name	Status
□	CERT CBT	Undefined
□	OG FA	Undefined
□	RMA	Undefined
□	**Select All**	

Select the MI Module for which you want to modify parameter values.

The **Messaging Interface Module <Module Name>** page is displayed.

Select **Config Parameters** from the Tree View on the left hand side of the page:



Messaging Interface Module *MI_FIA*

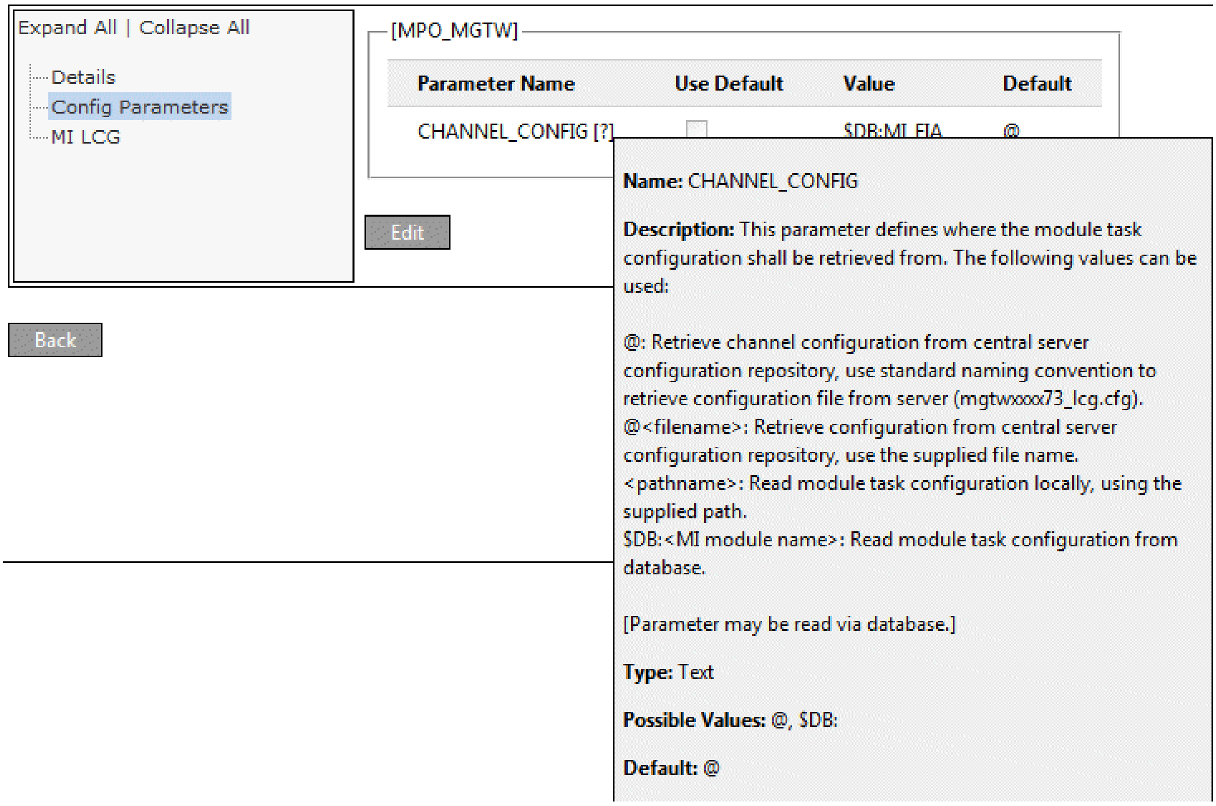
[MPO_MGTW]

Parameter Name	Use Default	Value	Default
CHANNEL_CONFIG [?]	□	$DB:MI_FIA	@

On the right hand side of the page you can see the specified configuration parameters and the value that has been set as well as the default value.

Parameters in **bold** are mandatory parameters.

By pointing at the **[?]** symbol with the cursor you can see the parameter description, the possible values and the default value.



[MPO_MGTW]

Parameter Name	Use Default	Value	Default
CHANNEL_CONFIG [?]	□	$DB:MI_FIA	@

Name: CHANNEL_CONFIG

Description: This parameter defines where the module task configuration shall be retrieved from. The following values can be used:

@: Retrieve channel configuration from central server configuration repository, use standard naming convention to retrieve configuration file from server (mgtwxxxx73_lcg.cfg).
@<filename>: Retrieve configuration from central server configuration repository, use the supplied file name.
<pathname>: Read module task configuration locally, using the supplied path.
$DB:<MI module name>: Read module task configuration from database.

[Parameter may be read via database.]

Type: Text

Possible Values: @, $DB:

Default: @

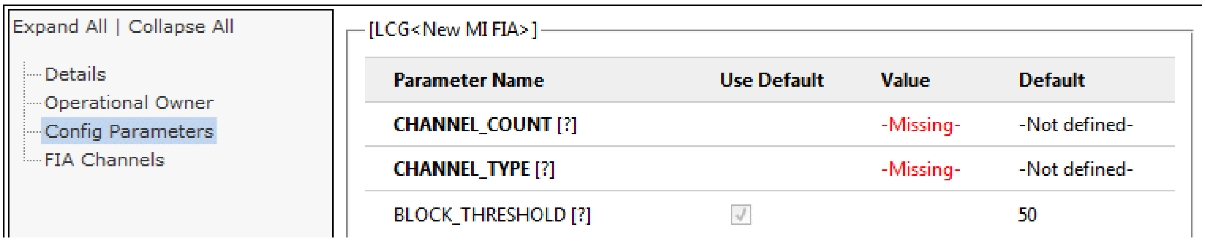
Press **Edit** to modify parameters:

The parameters and their values are displayed and may be modified.

If you check the **Use Default** box, the parameter will be reset and the default value will be used. This does not apply to mandatory parameters.

Press **Save** to have the changes become effective.

Note: When you create a new Module (**Messaging Interface Module**, **MI LCG**, **FileAct Channel**, **Output Session** or **Channel Connection**) it must be initialized before the configuration can be effective, i.e. at least the mandatory parameters must be written into the database. For this purpose the **Config Parameters** panel must be opened (by pressing **Edit**) and **Saved**. This is indicated by the word “-Missing-” (in red).



[LCG<New MI FIA>]

Parameter Name	Use Default	Value	Default
CHANNEL_COUNT [?]		-Missing-	-Not defined-
CHANNEL_TYPE [?]		-Missing-	-Not defined-
BLOCK_THRESHOLD [?]	☑		50

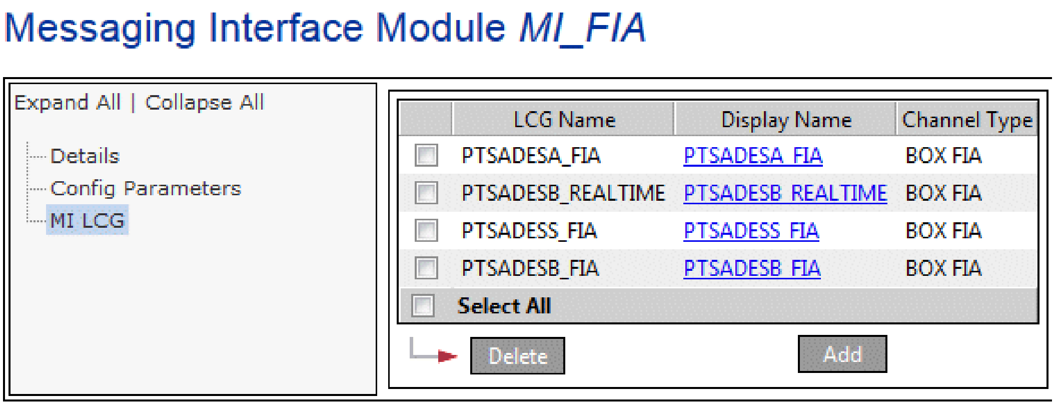
To view and to modify **MI LCG**, **FileAct Channel**, **Output Session** and **Channel Connection** configuration parameters, navigate further down in the Tree View on the left hand side and select the **MI LCG**, **FileAct Channel**, **Output Session** or **Channel Connection** you wish to configure (see figures below).

Press **Edit** on the respective page to modify the configuration.

For parameter descriptions move the cursor to the **[?]** symbol.

Always press **Save** to have the changes become effective.

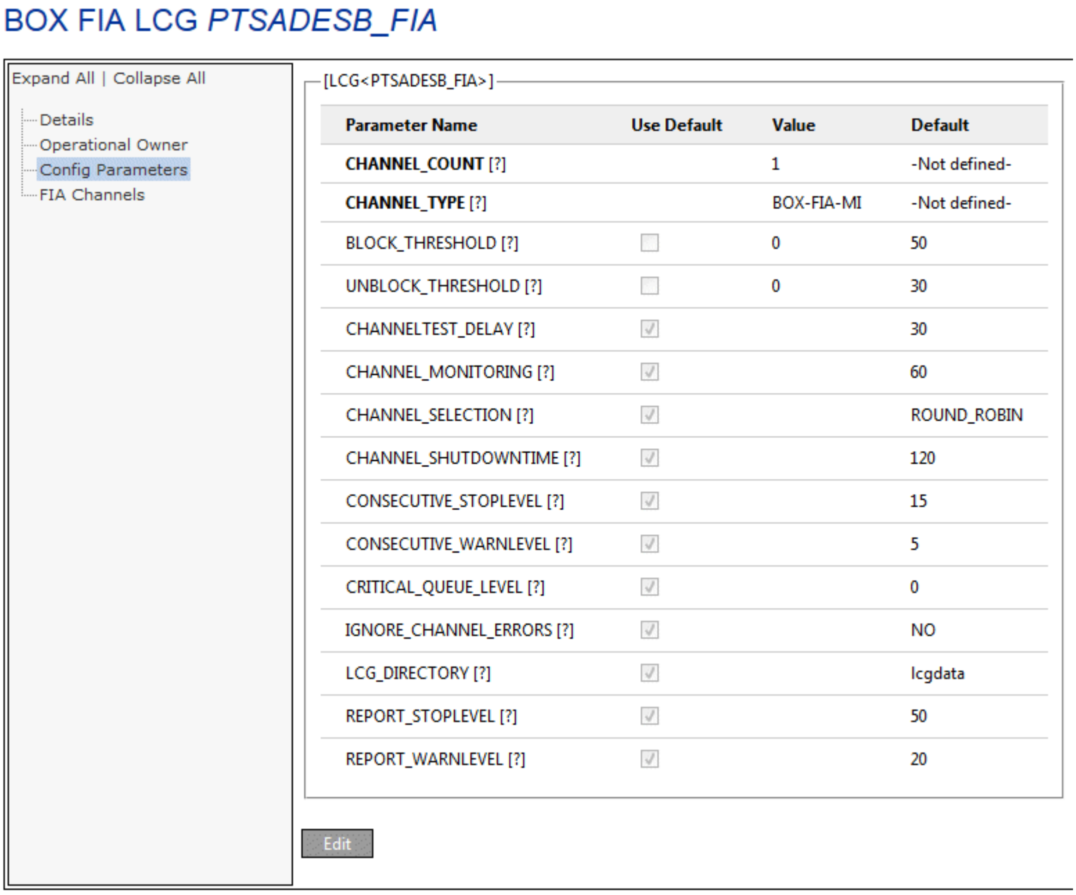
MI LCG:



Messaging Interface Module *MI_FIA*

	LCG Name	Display Name	Channel Type
□	PTSADESA_FIA	PTSADESA_FIA	BOX FIA
□	PTSADESB_REALTIME	PTSADESB_REALTIME	BOX FIA
□	PTSADESS_FIA	PTSADESS_FIA	BOX FIA
□	PTSADESB_FIA	PTSADESB_FIA	BOX FIA
□	**Select All**		

MI LCG:

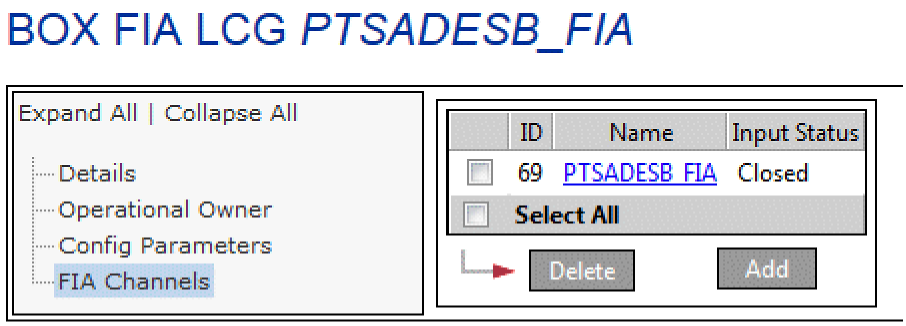


BOX FIA LCG *PTSADESB_FIA*

[LCG<PTSADESB_FIA>]

Parameter Name	Use Default	Value	Default
CHANNEL_COUNT [?]		1	-Not defined-
CHANNEL_TYPE [?]		BOX-FIA-MI	-Not defined-
BLOCK_THRESHOLD [?]	□	0	50
UNBLOCK_THRESHOLD [?]	□	0	30
CHANNELTEST_DELAY [?]	☑		30
CHANNEL_MONITORING [?]	☑		60
CHANNEL_SELECTION [?]	☑		ROUND_ROBIN
CHANNEL_SHUTDOWNTIME [?]	☑		120
CONSECUTIVE_STOPLEVEL [?]	☑		15
CONSECUTIVE_WARNLEVEL [?]	☑		5
CRITICAL_QUEUE_LEVEL [?]	☑		0
IGNORE_CHANNEL_ERRORS [?]	☑		NO
LCG_DIRECTORY [?]	☑		lcgdata
REPORT_STOPLEVEL [?]	☑		50
REPORT_WARNLEVEL [?]	☑		20

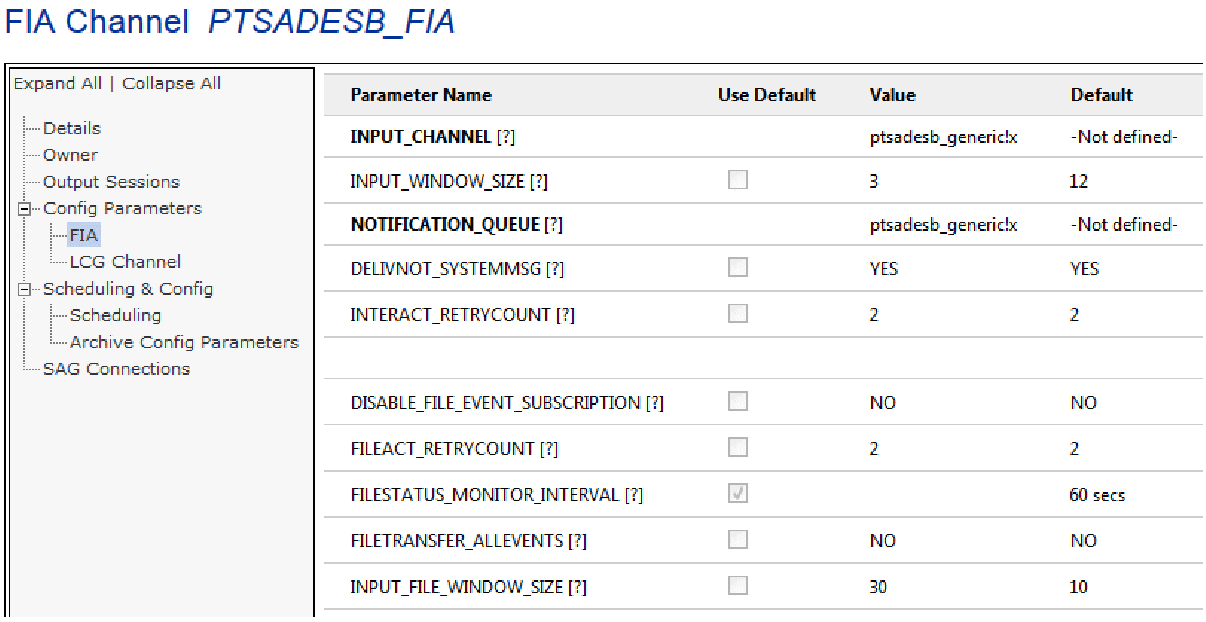
FIA Channels:



BOX FIA LCG *PTSADESB_FIA*

	ID	Name	Input Status
□	69	PTSADESB FIA	Closed
□	**Select All**		

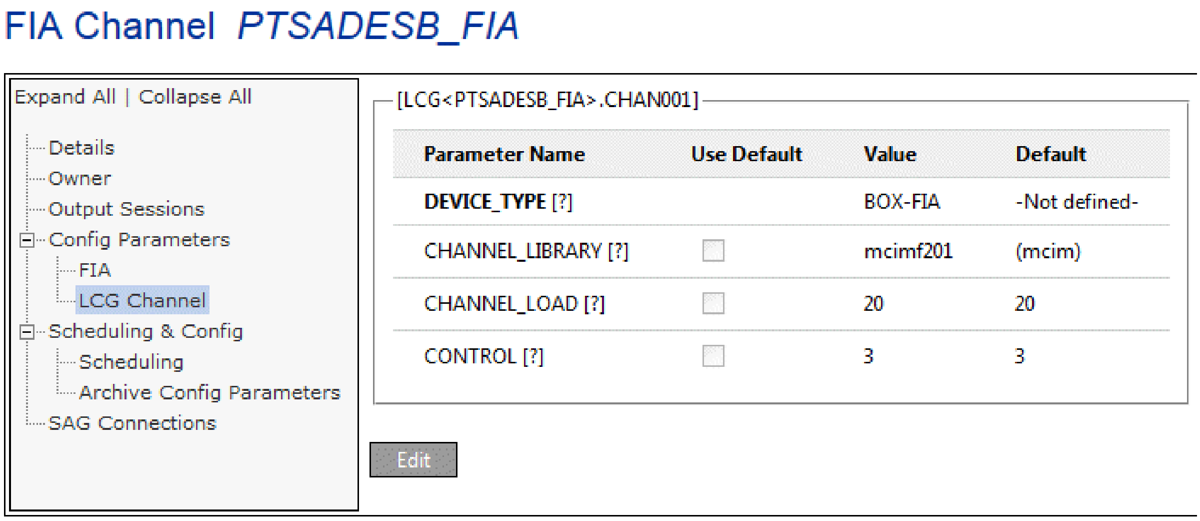
FIA Channel:



FIA Channel *PTSADESB_FIA*

Parameter Name	Use Default	Value	Default
INPUT_CHANNEL [?]		ptsadesb_generic!x	-Not defined-
INPUT_WINDOW_SIZE [?]	□	3	12
NOTIFICATION_QUEUE [?]		ptsadesb_generic!x	-Not defined-
DELIVNOT_SYSTEMMSG [?]	□	YES	YES
INTERACT_RETRYCOUNT [?]	□	2	2
DISABLE_FILE_EVENT_SUBSCRIPTION [?]	□	NO	NO
FILEACT_RETRYCOUNT [?]	□	2	2
FILESTATUS_MONITOR_INTERVAL [?]	☑		60 secs
FILETRANSFER_ALLEVENTS [?]	□	NO	NO
INPUT_FILE_WINDOW_SIZE [?]	□	30	10

LCG Channel:

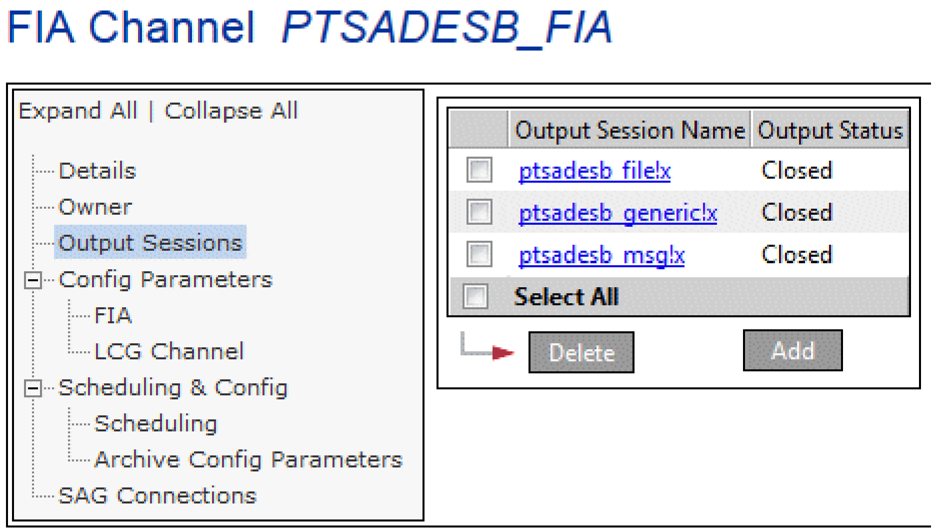


FIA Channel *PTSADESB_FIA*

[LCG<PTSADESB_FIA>.CHAN001]

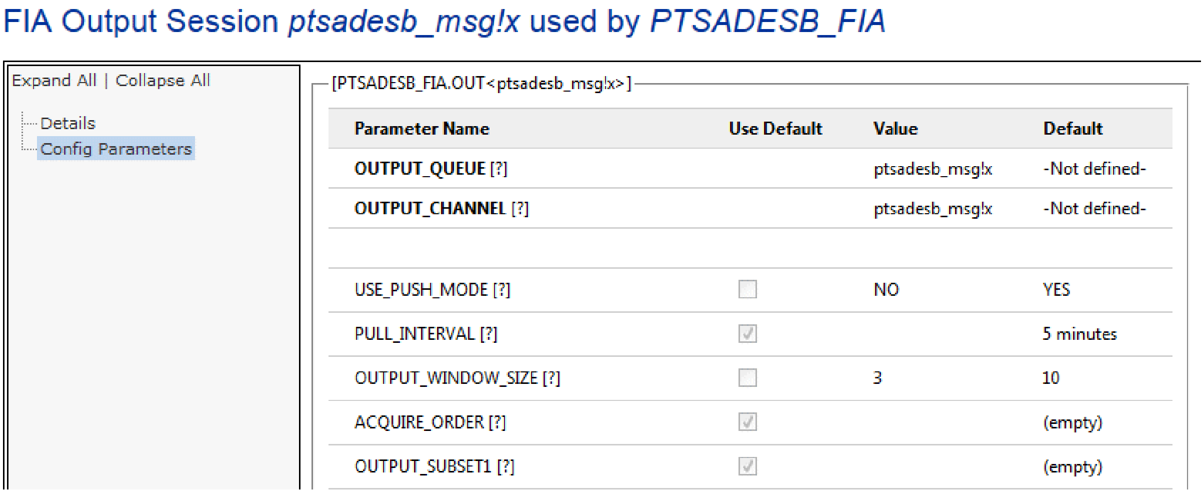
Parameter Name	Use Default	Value	Default
DEVICE_TYPE [?]		BOX-FIA	-Not defined-
CHANNEL_LIBRARY [?]	□	mcimf201	(mcim)
CHANNEL_LOAD [?]	□	20	20
CONTROL [?]	□	3	3

FIA Channel Output Sessions:



FIA Channel *PTSADESB_FIA*

	Output Session Name	Output Status
□	ptsadesb_file!x	Closed
□	ptsadesb_generic!x	Closed
□	ptsadesb_msg!x	Closed
□	**Select All**	



FIA Output Session *ptsadesb_msg!x* used by *PTSADESB_FIA*

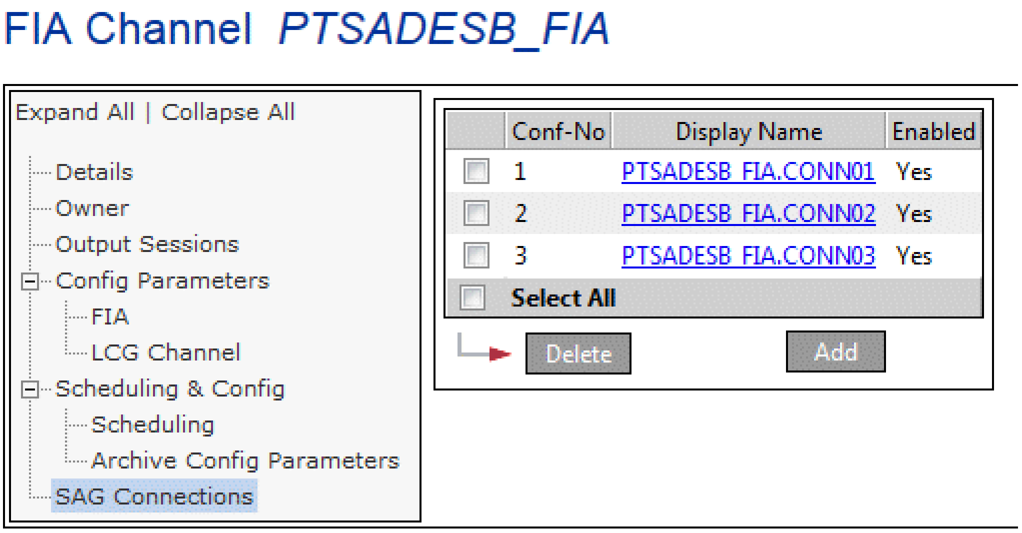
[PTSADESB_FIA.OUT<ptsadesb_msg!x>]

Parameter Name	**Use Default**	**Value**	**Default**
OUTPUT_QUEUE [?]		ptsadesb_msg!x	-Not defined-
OUTPUT_CHANNEL [?]		ptsadesb_msg!x	-Not defined-
USE_PUSH_MODE [?]	□	NO	YES
PULL_INTERVAL [?]	☑		5 minutes
OUTPUT_WINDOW_SIZE [?]	□	3	10
ACQUIRE_ORDER [?]	☑		(empty)
OUTPUT_SUBSET1 [?]	☑		(empty)

On the top of the right hand side you can see the configuration parameter section name **[PTSADESB_FIA.OUT<ptsadesb_msg!x>]** where the output session name (**ptsadesb_msg!x**) must match the name of the corresponding SWIFT SnF Queue (as specified with the parameter **OUTPUT_QUEUE**).

In the configuration there must be a dedicated section for each Output Session.

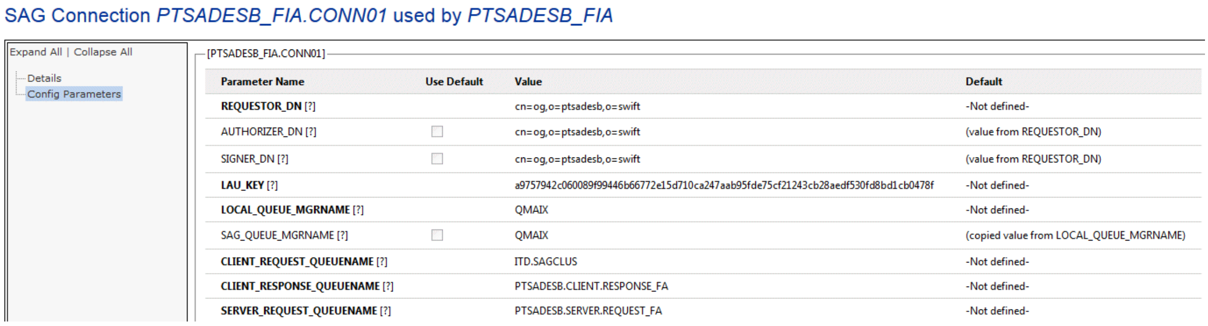
SAG Connection:



FIA Channel *PTSADESB_FIA*

	Conf-No	Display Name	Enabled
□	1	PTSADESB_FIA.CONN01	Yes
□	2	PTSADESB_FIA.CONN02	Yes
□	3	PTSADESB_FIA.CONN03	Yes
□	Select All		

SAG Connection:



SAG Connection *PTSADESB_FIA.CONN01* used by *PTSADESB_FIA*

[PTSADESB_FIA.CONN01]

Parameter Name	Use Default	Value	Default
REQUESTOR_DN [?]		cn=og,o=ptsadesb,o=swift	-Not defined-
AUTHORIZER_DN [?]	□	cn=og,o=ptsadesb,o=swift	(value from REQUESTOR_DN)
SIGNER_DN [?]	□	cn=og,o=ptsadesb,o=swift	(value from REQUESTOR_DN)
LAU_KEY [?]		a9757942c060089f99446b66772e15d710ca247aab95fde75cf21243cb28aedf530fd8bd1cb0478f	-Not defined-
LOCAL_QUEUE_MGRNAME [?]		QMAIX	-Not defined-
SAG_QUEUE_MGRNAME [?]	□	QMAIX	(copied value from LOCAL_QUEUE_MGRNAME)
CLIENT_REQUEST_QUEUENAME [?]		ITD.SAGCLUS	-Not defined-
CLIENT_RESPONSE_QUEUENAME [?]		PTSADESB.CLIENT.RESPONSE_FA	-Not defined-
SERVER_REQUEST_QUEUENAME [?]		PTSADESB.SERVER.REQUEST_FA	-Not defined-

7.2.1 Add, Delete, Modify Parameters

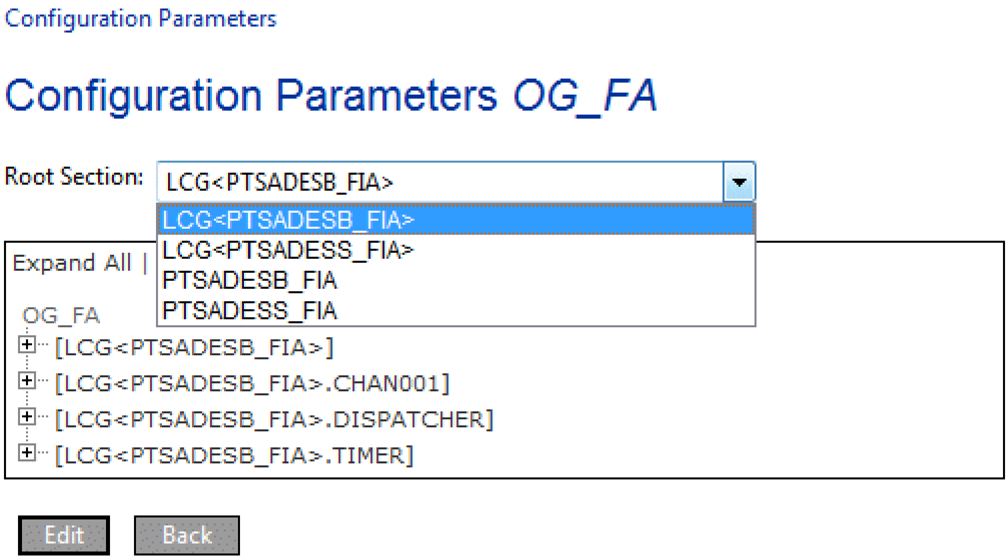
As stated above, the Messaging Interface related configuration panels show all parameters that are relevant for the configuration and normally there is no need to add further parameters or to delete existing ones.

However, if adding or deleting parameters becomes necessary for whatsoever reason, select **Administration / Configuration Parameters** from the main menu.

The **Configuration Parameters** page is displayed.

Click on the hyperlink (of the MI Module for which you want to make the configuration changes) in the **Module Name** column.

The **Configuration Parameters <Module Name>** page is displayed:



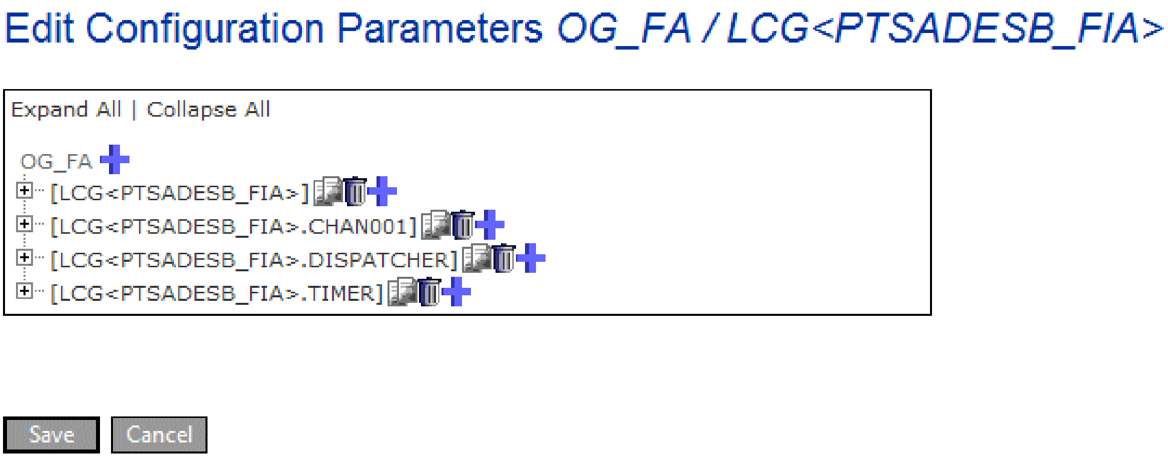
Configuration Parameters

Configuration Parameters OG_FA

Root Section: LCG<PTSADESB_FIA>

Select the configuration parameter section in which you wish to add or delete parameters from the Root Section list

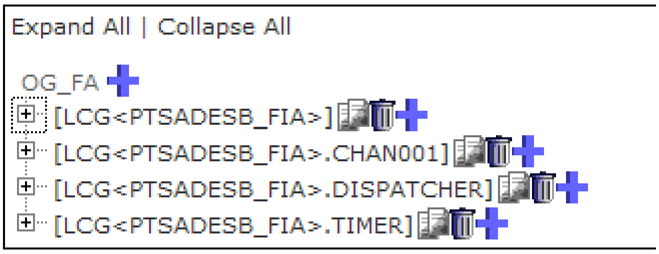
Click **Edit** to open the page in edit modus.



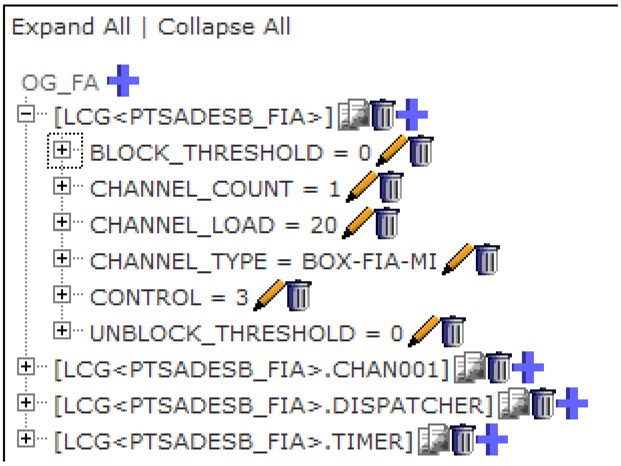
Edit Configuration Parameters OG_FA / LCG<PTSADESB_FIA>

Expand All | Collapse All
OG_FA
[LCG<PTSADESB_FIA>]
[LCG<PTSADESB_FIA>.CHAN001]
[LCG<PTSADESB_FIA>.DISPATCHER]
[LCG<PTSADESB_FIA>.TIMER]
Save
Cancel

Each section or parameter with a + symbol on the left hand side of the section/parameter can be expanded by clicking to the + symbol.



```
Expand All | Collapse All

OG_FA
⊞ [LCG<PTSADESB_FIA>]
⊞ [LCG<PTSADESB_FIA>.CHAN001]
⊞ [LCG<PTSADESB_FIA>.DISPATCHER]
⊞ [LCG<PTSADESB_FIA>.TIMER]
```



```
Expand All | Collapse All

OG_FA
⊟ [LCG<PTSADESB_FIA>]
    ⊞ BLOCK_THRESHOLD = 0
    ⊞ CHANNEL_COUNT = 1
    ⊞ CHANNEL_LOAD = 20
    ⊞ CHANNEL_TYPE = BOX-FIA-MI
    ⊞ CONTROL = 3
    ⊞ UNBLOCK_THRESHOLD = 0
⊞ [LCG<PTSADESB_FIA>.CHAN001]
⊞ [LCG<PTSADESB_FIA>.DISPATCHER]
⊞ [LCG<PTSADESB_FIA>.TIMER]
```

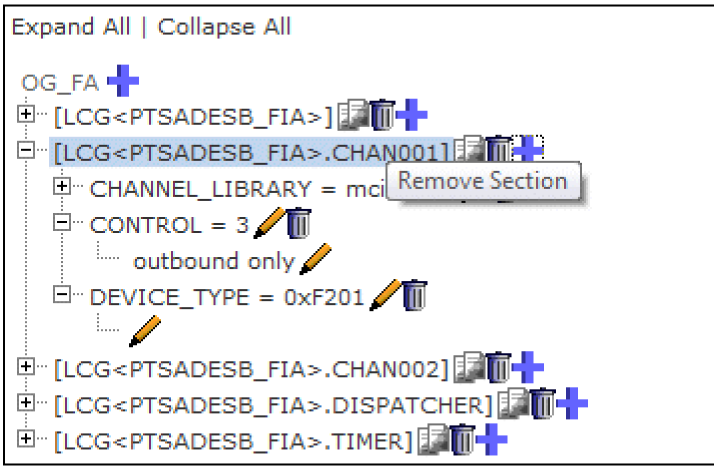
On the right hand side you find the following symbols:

✚	Click to this symbol to add a new configuration parameter or parameter section.
📄	Click to this symbol to copy the configuration parameter section. This function can be very time-saving because the copied section contains all the parameters (and the same parameter values) as the original section. The respective module must be created as described in 4.3 - 4.10.
🗑	Click to this symbol to delete the configuration parameter or parameter section. Before the parameter / section will be deleted you will be asked whether you really want to delete or not.
✏	Click to this symbol to modify the configuration parameter or parameter section.
✏	Click to this symbol to modify the parameter description.

If you use the tooltip, you can get a description of each of these symbols:

The **Unsaved changes**! notification indicates that you have made changes in the configuration. To have them become effective you must press **Save**.

Unsaved changes!


Expand All | Collapse All

OG_FA

[LCG<PTSADESB_FIA>]

[LCG<PTSADESB_FIA>.CHAN001]

CHANNEL_LIBRARY = mci

Remove Section

CONTROL = 3

outbound only

DEVICE_TYPE = 0xF201

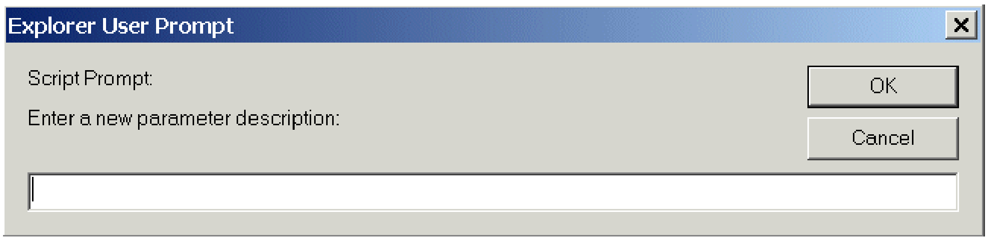
[LCG<PTSADESB_FIA>.CHAN002]

[LCG<PTSADESB_FIA>.DISPATCHER]

[LCG<PTSADESB_FIA>.TIMER]

When you press one of the symbols you will get an browser User Prompt telling you what to do next:

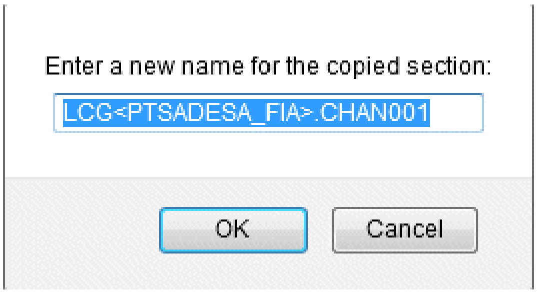
(Microsoft Internet Explorer)


Explorer User Prompt

Script Prompt:

Enter a new parameter description:

(Mozilla Firefox)


Enter a new name for the copied section:

LCG<PTSADESA_FIA>.CHAN001

Note that the Internet Explorer settings should a**llow websites to prompt for information using scripted windows**, otherwise the parameter cannot be edited.

For available parameters and parameter descriptions, please refer to the document BOX **Configuration Guide**.

7.2.2 Export Module Configuration

The configuration of modules can be exported to a file by selecting one or more modules (Module Name) and by pressing the **Export** button on the **Configuration Parameters** page.

The connection configuration will be displayed in XML format.

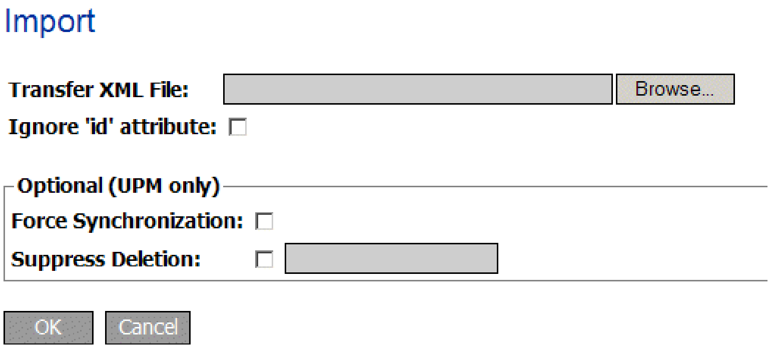
Save the page using the **Save as...** functionality of the browser.

Exporting of module configurations can also be done via command line, i.e. by using the mpoTransfer tool.

7.2.3 Import Module Configuration

The configuration of modules can be imported from file by pressing the **Import** button on the **Configuration Parameters** page.

The **Import** page is displayed:



Import

Transfer XML File: [Browse...]

Ignore 'id' attribute: □

Optional (UPM only)

Force Synchronization: □

Suppress Deletion: □

To import the configuration, either enter file name and path of the file you want to import or browse for the file you want to import.

If you check the **Ignore 'id' attribute** checkbox, all id attributes in the import file will be ignored during the import. Use this option for simple imports containing objects which are not related.
Note: Complex imports like workflow will not work with this option enabled.

If you check the **Force Synchronization** box, the default import mode is overridden and set to "synchronize" ("**sync**").

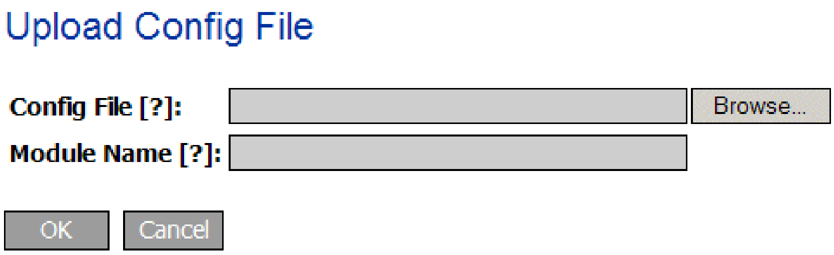
Activating the **Suppress Deletion** parameter represents an option for suppressing the deletion of UPM user objects from the import file. The activated parameter suppresses direct deletion of the user object and merely deactivates the user and sets his synchronization mode to "ToBeDeleted". The field to the right of the checkbox can be used for entering an optional comment.

Importing of module configurations can also be done via command line, i.e. by using the mpoTransfer tool.

7.2.4 Upload Module Configuration

The configuration of modules can be uploaded to the Web-client from a configuration file by pressing the **Upload** button on the **Configuration Parameters** page.

The Upload Config File page is displayed:



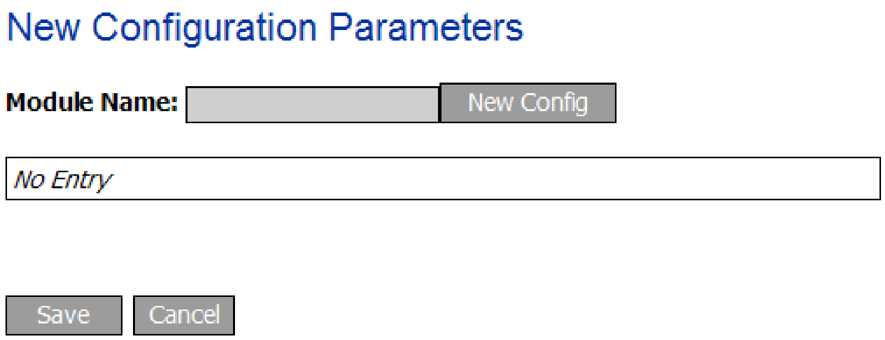
Either enter the name and path of the configuration file that you want to be uploaded or browse for the file and press **OK**.

The **Module Name** is optional. If no name is entered, the name of the file (without suffix) will be used.

7.2.5 Add Module Configuration

New configurations of Modules can be added by pressing the **Add** button on the **Configuration Parameters** page.

The New Configuration Parameters page is displayed:



Enter the desired Module Name.

Press **New Config**.

Enter new configuration parameter.

Repeat the above step for each new parameter.

Press **Save** after you have finished entering parameters.

7.2.6 Delete Module Configuration

The configuration of Modules can be deleted by selecting one or more modules (Module Name) and by pressing the **Delete** button on the **Configuration Parameters** page.

8 Configuration in SAG

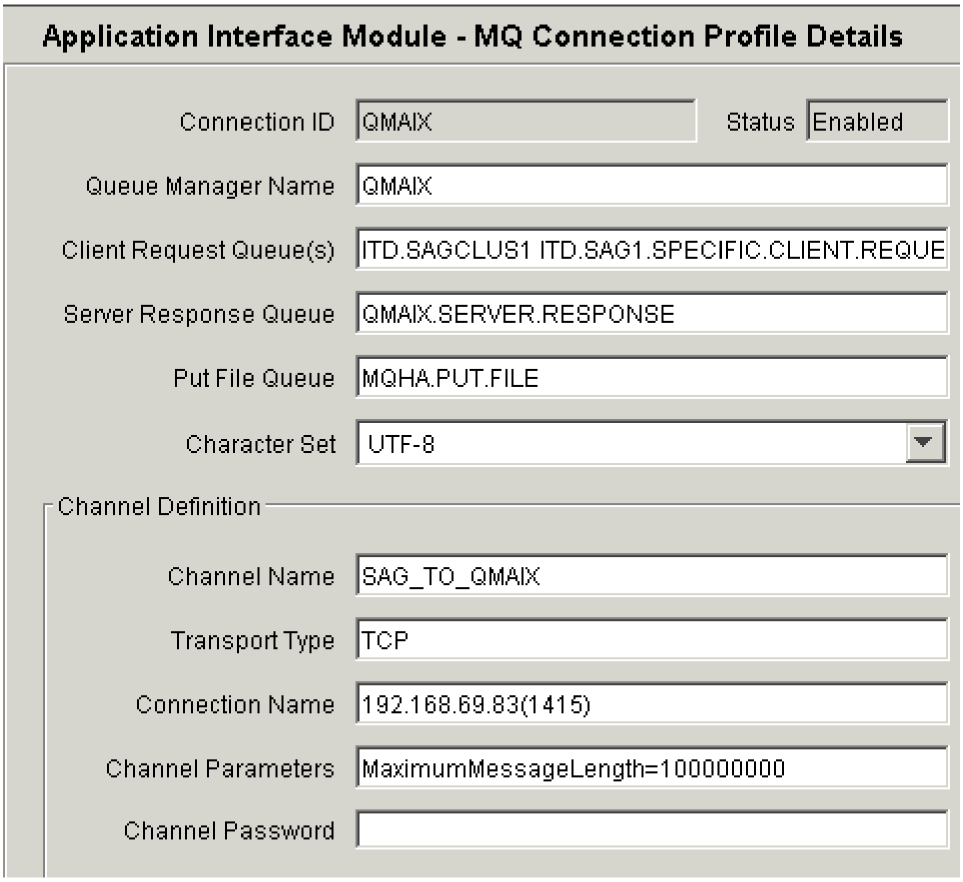
In SAG the following settings must be made:

8.1 General Configuration

8.1.1 Application Interface - MQ Connection Profile

If one MI Module uses several SnF Channels you need to configure only one MQ Connection Profile. This profile can be used for all Channels using different Message Partners.

The **Client Request Queue(s)** must match the BOX configuration parameter **CLIENT_REQUEST_QUEUENAME** in section [**<FIA-CHANNELNAME>.CONNXX]**.



Application Interface Module - MQ Connection Profile Details

Connection ID: QMAIX
Status: Enabled
Queue Manager Name: QMAIX
Client Request Queue(s): ITD.SAGCLUS1 ITD.SAG1.SPECIFIC.CLIENT.REQUE
Server Response Queue: QMAIX.SERVER.RESPONSE
Put File Queue: MQHA.PUT.FILE
Character Set: UTF-8

Channel Definition

Channel Name: SAG_TO_QMAIX
Transport Type: TCP
Connection Name: 192.168.69.83(1415)
Channel Parameters: MaximumMessageLength=100000000
Channel Password:

8.1.2 Application Interface - Message Partner Details

The **Name** of the Application Interface Module **must** match the BOX configuration parameter **MESSAGE_PARTNER** in section [**<FIA-CHANNELNAME>.CONNXX**].

The Message Partner details on the different tabs have to be set as follows:

8.1.2.1 Main

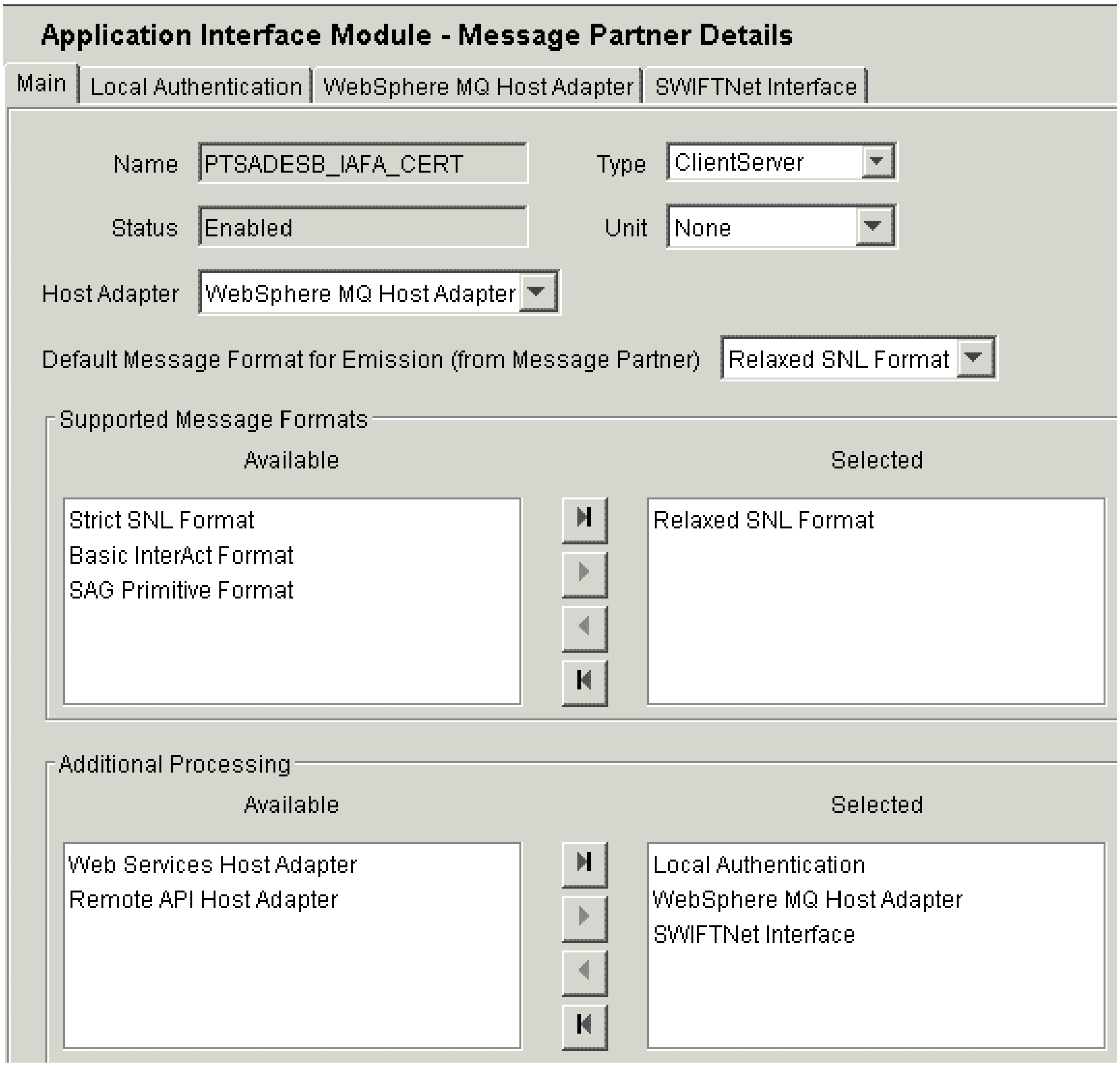
Select “ClientServer” as “Type”. (If PULL mode shall be used, select “Client” as “Type”.)

Select “Websphere MQ Host Adapter” as “Host Adapter”

Select “Relaxed SNL Format” as default message format

Select “Relaxed SNL Format” as message format under “Supported Message Formats”

Select “Local Authentication” “Websphere MQ Host Adapter” and “SWIFTNET Interface” under “Additional Processing”



Application Interface Module - Message Partner Details

Name PTSADESB_IAFA_CERT

Type ClientServer

Status Enabled

Unit None

Host Adapter WebSphere MQ Host Adapter

Default Message Format for Emission (from Message Partner) Relaxed SNL Format

Supported Message Formats

Available	Selected
Strict SNL Format Basic InterAct Format SAG Primitive Format	Relaxed SNL Format

Additional Processing

Available	Selected
Web Services Host Adapter Remote API Host Adapter	Local Authentication WebSphere MQ Host Adapter SWIFTNet Interface

8.1.2.2 Websphere MQ Host Adapter

The **Server Request Queue** must match the BOX configuration parameter **SERVER_REQUEST_QUEUENAME** in section **[<FIA-CHANNELNAME>.CONNXX]**.

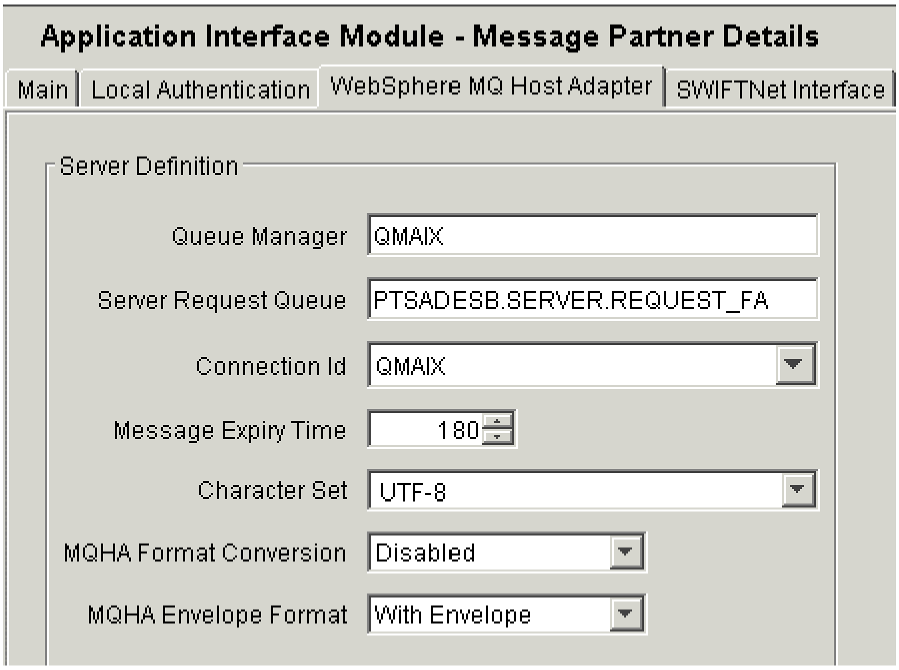
Select the MQ Connection Profile configured in chapter 8.1.1

Enter the MQ expiration time when the message will be removed from the MQ queue

Select UTF-8 as Character Set

Select “Disabled” for the “MQHA Format Convention”

Select “With Envelope” in the “MQHA Envelope Format”



Application Interface Module - Message Partner Details

Server Definition

Queue Manager: QMAIX

Server Request Queue: PTSADESB.SERVER.REQUEST_FA

Connection Id: QMAIX

Message Expiry Time: 180

Character Set: UTF-8

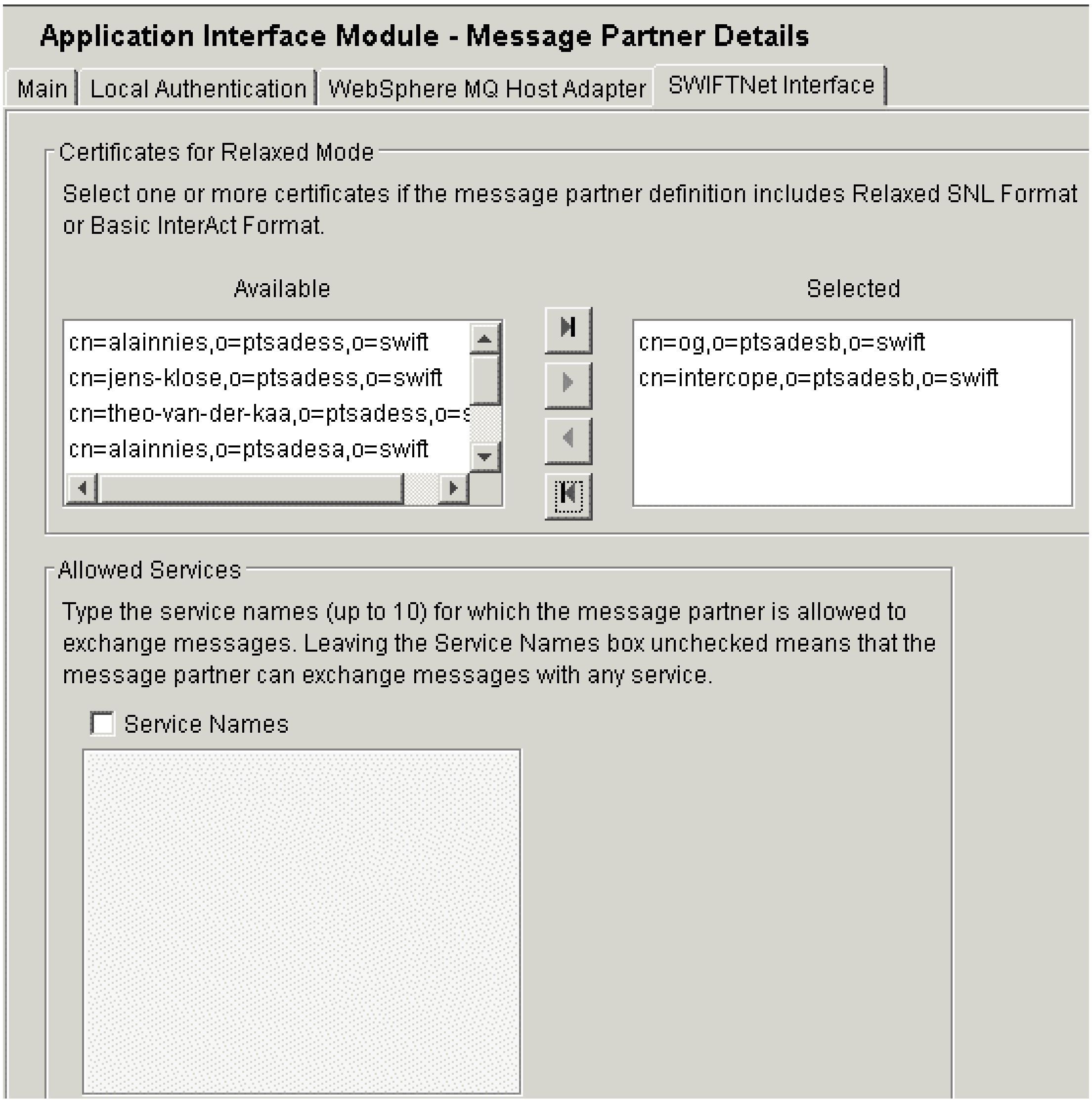
MQHA Format Conversion: Disabled

MQHA Envelope Format: With Envelope

(If PULL mode shall be used, you only need select UTF-8 as Character Set.

8.1.2.3 SWIFTNet Interface

All DNs configured in the MI Module (requestor, signer, authorizer...) must be selected for the used Message Partner.



Application Interface Module - Message Partner Details

Certificates for Relaxed Mode

Select one or more certificates if the message partner definition includes Relaxed SNL Format or Basic InterAct Format.

Available:
cn=alainnies,o=ptsadess,o=swift
cn=jens-klose,o=ptsadess,o=swift
cn=theo-van-der-kaa,o=ptsadess,o=s
cn=alainnies,o=ptsadesa,o=swift

Selected:
cn=og,o=ptsadesb,o=swift
cn=intercope,o=ptsadesb,o=swift

Allowed Services

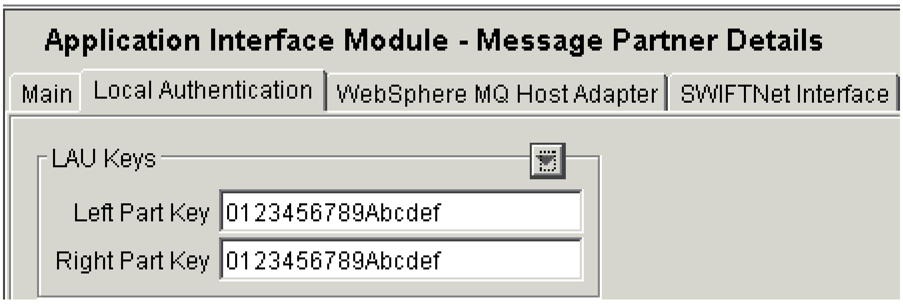
Type the service names (up to 10) for which the message partner is allowed to exchange messages. Leaving the Service Names box unchecked means that the message partner can exchange messages with any service.

□ Service Names

8.1.2.4 Local Authentication

The LAU Key values (Left Part Key and Right Part Key together in one string) must match the value of the BOX configuration parameter **LAU_KEY** in section **[<FIA-CHANNELNAME>.CONNXX]**

If no LAU KEY is used, leave the parameter **LAU_KEY** in section **[<FIA-CHANNELNAME>.CONNXX]**empty.



Application Interface Module - Message Partner Details

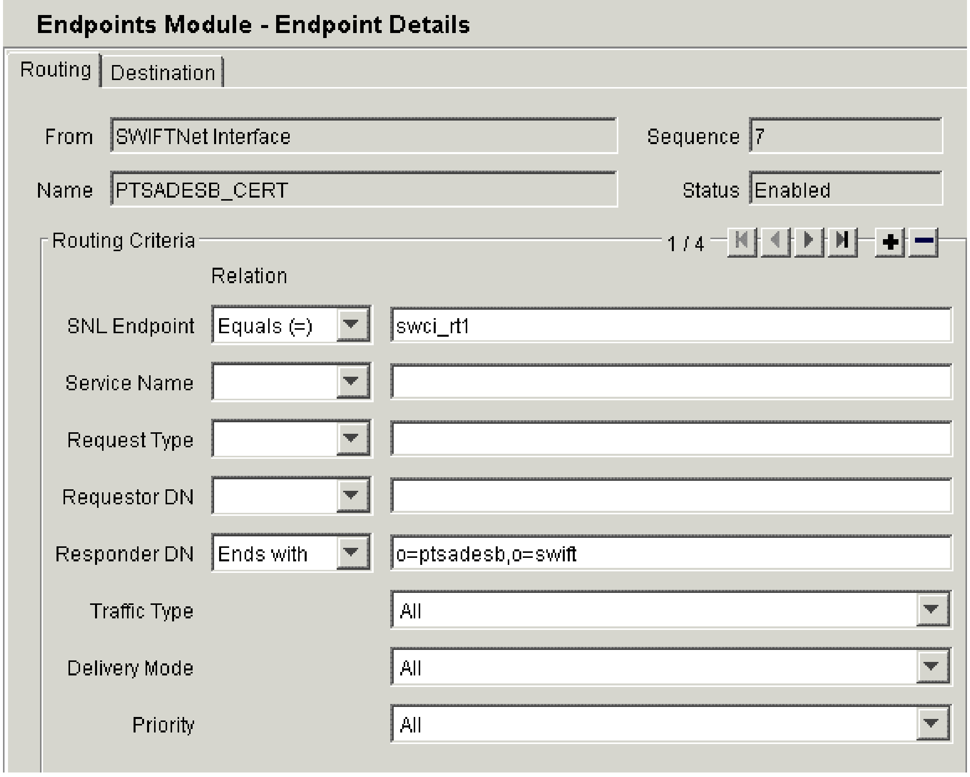
Left Part Key 0123456789Abcdef

Right Part Key 0123456789Abcdef

8.1.3 SNL Endpoints Module - Routing

The **SNL Endpoint** in the SAG configuration **must** match the BOX configuration parameter **SNL_ENDPOINT** in section **[<FIA-CHANNELNAME>.CONNXX]**.

This **SNL Endpoint** is used by all SnF Sessions of one and the same MI LCG in BOX (=SAG Message Partner)



Endpoints Module - Endpoint Details

From SWIFTNet Interface

Sequence 7

Name PTSADESB_CERT

Status Enabled

SNL Endpoint Equals (=) swci_rt1

Service Name

Request Type

Requestor DN

Responder DN Ends with o=ptsadesb,o=swift

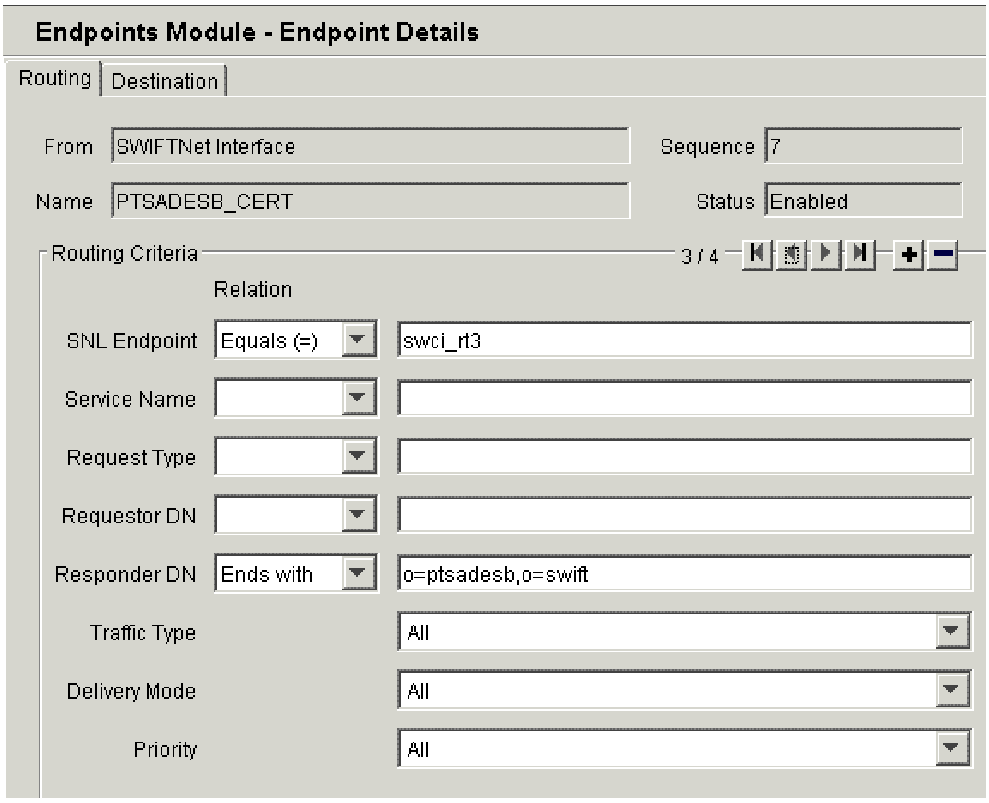
Traffic Type All

Delivery Mode All

Priority All

For SWIFT real-time services SWIFT assigns the SNL Endpoint to be used when a service is subscribed.

Therefor you must specify an **SNL Endpoint** separately for each SWIFT real-time service used by a certain **Connection** specified in the BOX configuration.



Endpoints Module - Endpoint Details

Routing | Destination

From: SWIFTNet Interface
Sequence: 7
Name: PTSADESB_CERT
Status: Enabled

Routing Criteria 3 / 4

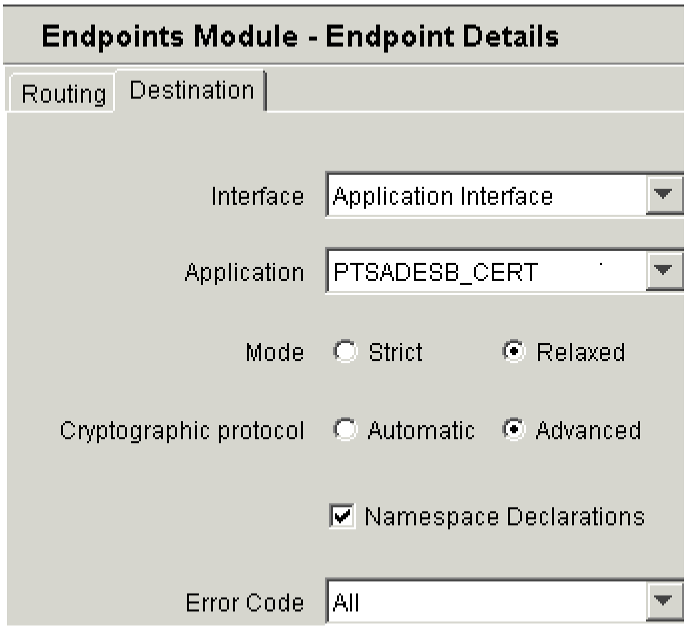
	Relation	
SNL Endpoint	Equals (=)	swci_rt3
Service Name		
Request Type		
Requestor DN		
Responder DN	Ends with	o=ptsadesb,o=swift
Traffic Type		All
Delivery Mode		All
Priority		All

8.1.4 SNL Endpoints Module - Destination

The SNL mode **must** be set to “Relaxed”.

The Cryptographic protocol **must** be “Advanced”.

The **Namespace Declarations** box **must** be checked.



Endpoints Module - Endpoint Details

Routing | Destination

Interface: Application Interface

Application: PTSADESB_CERT

Mode: ○ Strict ● Relaxed

Cryptographic protocol: ○ Automatic ● Advanced

☑ Namespace Declarations

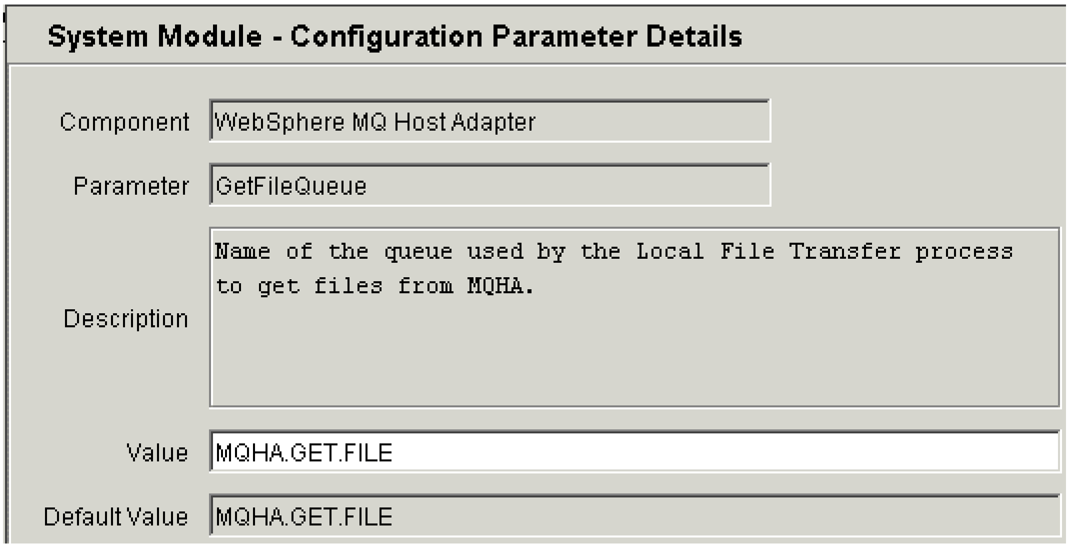
Error Code: All

8.2 Special Configuration for Local File Transfer

When the Local File Transfer (LFT) transfer mechanism shall be used the following configuration has to be done:

The Local File Transfer requires two MQ queues to be configured in SAG, the **GetFileQueue** (System parameter) and the **Put File Queue** (Application Interface Module - MQ Connection parameter).

The GetFileQueue:

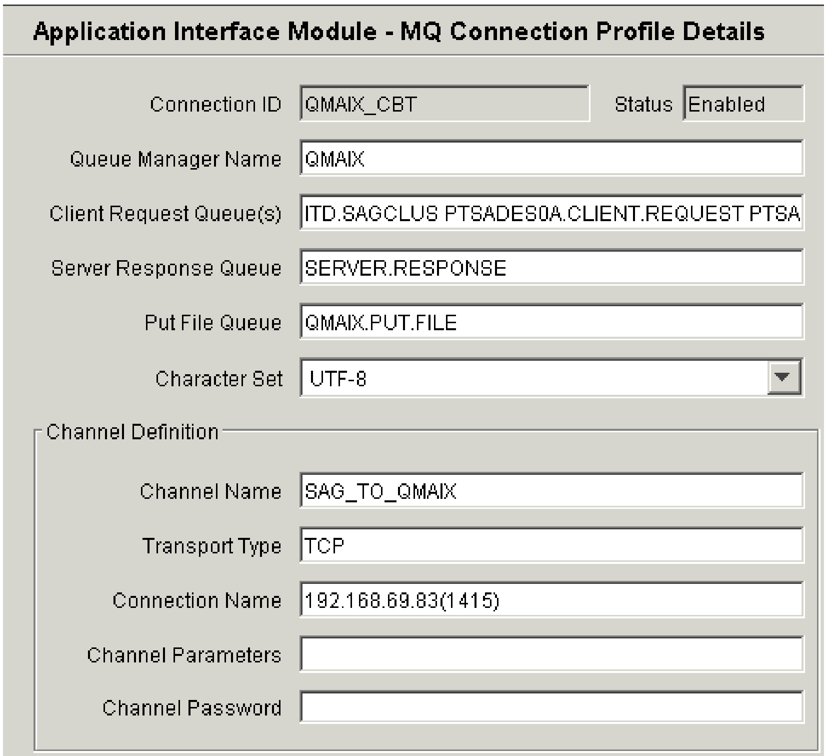


System Module - Configuration Parameter Details

Component	WebSphere MQ Host Adapter
Parameter	GetFileQueue
Description	Name of the queue used by the Local File Transfer process to get files from MQHA.
Value	MQHA.GET.FILE
Default Value	MQHA.GET.FILE

The **GetFileQueue** must match the queue specified by the BOX configuration parameter **LFT_GET_FILE_QUEUE** in section **[<FIA-CHANNELNAME>.CONNXX]**. As the **GetFileQueue** is an SAG System Parameter, SAG may write only to one queue. However, if multiple MQ Queue Managers are used, SAG may write to different queues (each one representing one BOX LFT_GET_FILE_QUEUE) as long as the names of the queues are the same.

The Put File Queue:



Application Interface Module - MQ Connection Profile Details

Connection ID	QMAIX_CBT	Status	Enabled
Queue Manager Name	QMAIX		
Client Request Queue(s)	ITD.SAGCLUS PTSADES0A.CLIENT.REQUEST PTSA		
Server Response Queue	SERVER.RESPONSE		
Put File Queue	QMAIX.PUT.FILE		
Character Set	UTF-8		

Channel Definition

Channel Name	SAG_TO_QMAIX
Transport Type	TCP
Connection Name	192.168.69.83(1415)
Channel Parameters	
Channel Password	

The **Put File Queue** name must match the queue specified by the BOX configuration parameter **LFT_PUT_FILE_QUEUE** in section **[<FIA-CHANNELNAME>.CONNXX]**. As the **Put File Queue** parameter is Application Interface specific, there may be multiple Put File Queues (each one representing one LFT_PUT_FILE_QUEUE) to which different MI Modules can write.

Additionally a **Client Request Queue** must be configured in SAG (see figure above).

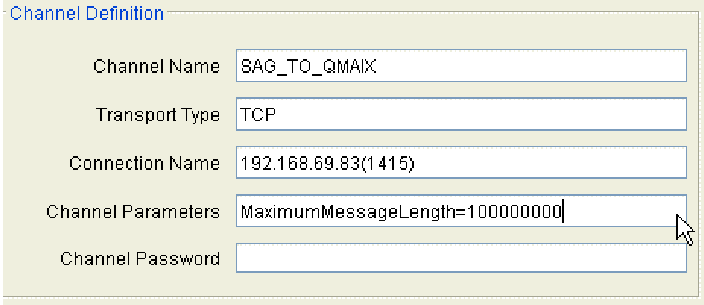
The name of this queue must match the value specified by the BOX configuration parameters (both parameters have the same value) **CLIENT_REQUEST_QUEUENAME** and **LFT_COMMAND_QUEUE** in section **[<FIA-CHANNELNAME>.CONNXX]**.

The Client Request Queue can be shared by all channels.

A Server Response Queue must be specified (one for each Server Request Queue [Message Partner configuration, see below]).

The Server Response Queue must be specified only in the SAG configuration and need not have a matching queue in the BOX configuration.

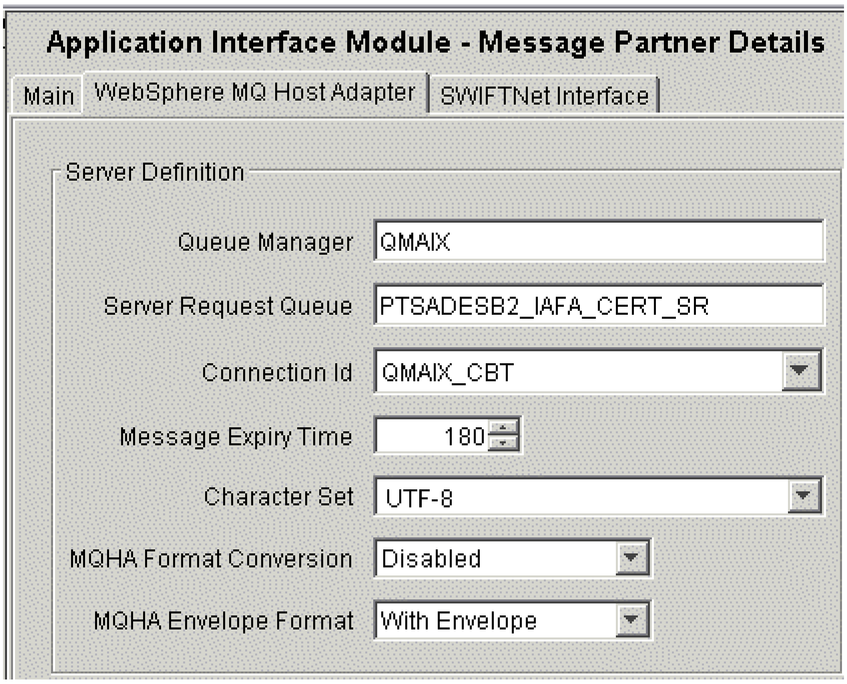
Note: The default maximum message size for MQ Queues is 4Mb (4194304 bytes). If an MQ Queue shall be able to handle messages > 4Mb, the channel definition parameter in the SAG MQ profile **MaximumMessageLength** must be set accordingly:



Channel Definition

Channel Name	SAG_TO_QMAIX
Transport Type	TCP
Connection Name	192.168.69.83(1415)
Channel Parameters	MaximumMessageLength=100000000
Channel Password	

Furthermore a **Server Request Queue** must be configured for the SAG Message Partner.



Application Interface Module - Message Partner Details

Server Definition

Queue Manager	QMAIX
Server Request Queue	PTSADESB2_IAFA_CERT_SR
Connection Id	QMAIX_CBT
Message Expiry Time	180
Character Set	UTF-8
MQHA Format Conversion	Disabled
MQHA Envelope Format	With Envelope

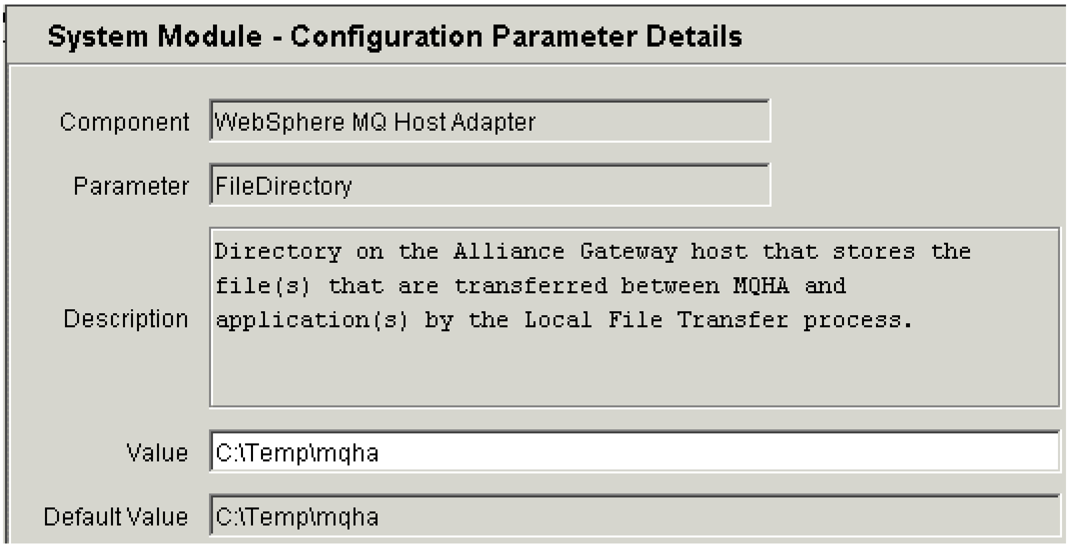
This queue must match the queue specified in the BOX configuration by the parameter **SERVER_REQUEST_QUEUENAME** in section **[<FIA-CHANNELNAME>.CONNXX]**.

There must be a separate Server Request Queue for each BOX FIA channel.

Each Server Request Queue must have a Server Response Queue (Application Interface Module – MQ Connection configuration, see above).

Finally subdirectories named 'BOXCHN_<FIA_CHANNELNAME>/Outgoing' and 'BOXCHN_<FIA_CHANNELNAME>/Receiving' used for storing the files for individual channels and directions **must be manually created**. The ‘Outgoing’ directory will store the files transferred from BOX to SAG. The ‘Receiving’ directory will store the files that shall be transferred from SAG to BOX.

The location of these subdirectories has to be specified with the SAG parameter **FileDirectory** in the **"Websphere MQ Host Adapter**" component configuration. The value of the parameter **FileDirectory** must match the BOX configuration parameter **FACT_TRANSFER_SAGDIR** in section **[<FIA-CHANNELNAME>.CONNXX]** (e.g.: C:\Temp\mqha, as in the figure below):



System Module - Configuration Parameter Details

Component	WebSphere MQ Host Adapter
Parameter	FileDirectory
Description	Directory on the Alliance Gateway host that stores the file(s) that are transferred between MQHA and application(s) by the Local File Transfer process.
Value	C:\Temp\mqha
Default Value	C:\Temp\mqha

9 Special Configurations

9.1 Real-Time Delivery Notifications / Multiple MI Channels

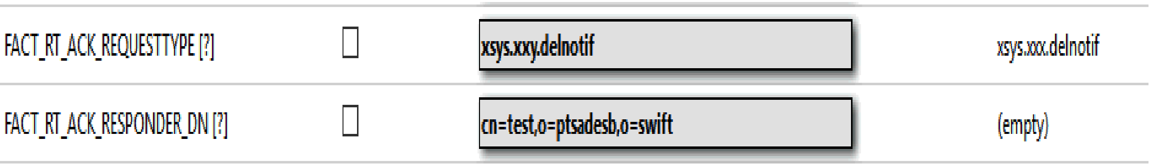
If multiple FileAct MI Channels have been defined for one and the same BIC, it must be ensured that Real-Time Delivery Notifications will be received via the very same channel that was used for sending the Input message that the Notification belongs to.

This requires some configuration modifications:

9.1.1 MI SAG Connection

In the SAG Connection configuration of the MI Channel you can specify which "Request Type" and/or which "Responder DN" shall be used for the Delivery Notifications.

In order to allocate the Delivery Notifications unambiguously to the corresponding MI Channel these values must be unique for each SAG Connection.

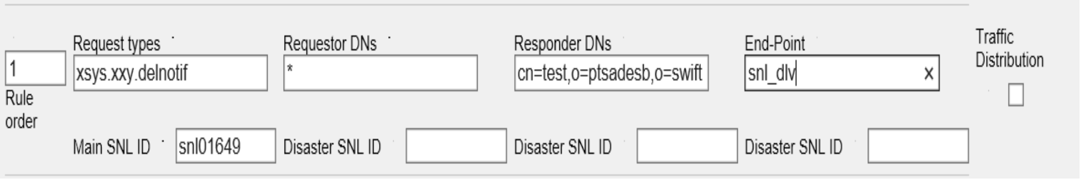


FACT_RT_ACK_REQUESTTYPE [?]	□	xsys.xxy.delnotif	xsys.xxx.delnotif
FACT_RT_ACK_RESPONDER_DN [?]	□	cn=test,o=ptsadesb,o=swift	(empty)

9.1.2 SWIFT Routing Rule when using Multiple SNLs

If multiple SNLs are used, it must be ensured that the Real-Time Delivery Notifications will be routed to the very same SNL that was used for sending the Input message that the Notification belongs to.

This is achieved by specifying one SWIFT Routing Rule for Real-Time Notifications for each MI Channel, e.g.:



Rule order: 1

Request types: xsys.xxy.delnotif

Requestor DNs: *

Responder DNs: cn=test,o=ptsadesb,o=swift

End-Point: snl_dlv

Traffic Distribution: □

Main SNL ID: snl01649

Disaster SNL ID:

Disaster SNL ID:

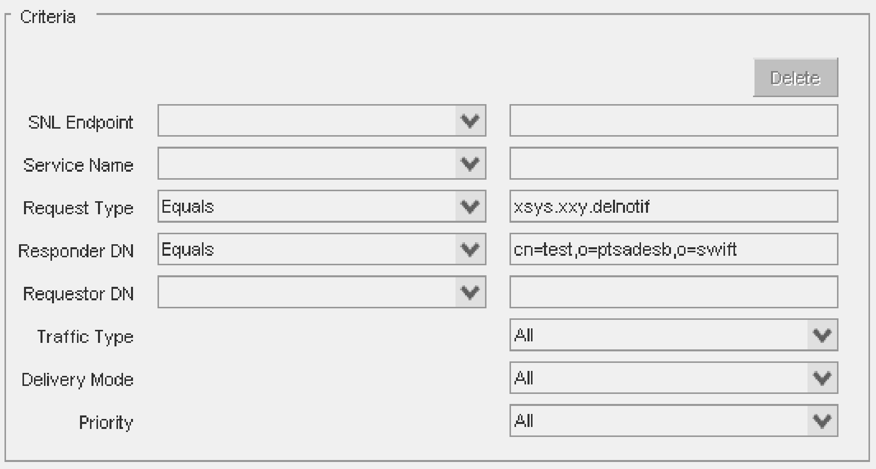
Disaster SNL ID:

If only one SNL is used, this is not necessary as all messages will be routed to this SNL.

9.1.3 SAG Endpoint Configuration

When the Delivery Notifications are routed to the SNL, they must then be routed to the corresponding Message Partner.

This is achieved by specifying the SAG Endpoint Routing accordingly, e.g. by using the “Request Type” and “Responder DN” parameters (see above) as “Routing Criteria":

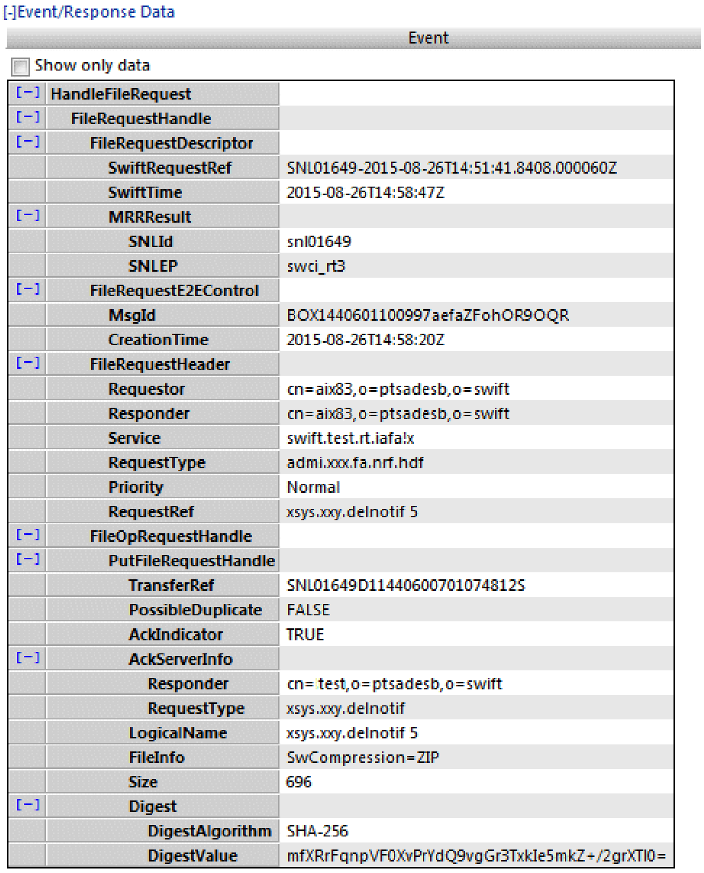


Criteria

SNL Endpoint		
Service Name		
Request Type	Equals	xsys.xxy.delnotif
Responder DN	Equals	cn=test,o=ptsadesb,o=swift
Requestor DN		
Traffic Type		All
Delivery Mode		All
Priority		All

9.1.4 Example

Below you can see a Put File Request with the information (AckServerInfo) concerning the delivery notification (the delivery notification will be sent back with the req type (xsys.xxy.delnotif) and the responder DN (cn=test.xxxy.delnotif)):



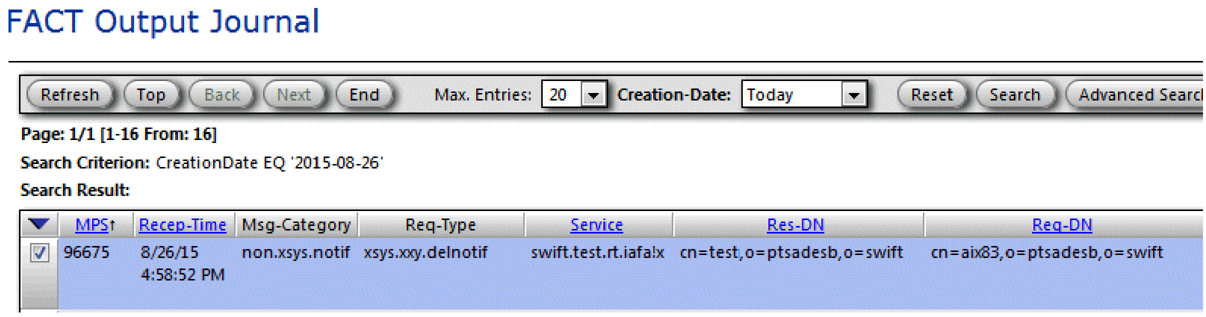
[-]Event/Response Data

Event

□ Show only data

[−]	HandleFileRequest	
[−]	FileRequestHandle	
[−]	FileRequestDescriptor	
	SwiftRequestRef	SNL01649-2015-08-26T14:51:41.8408.000060Z
	SwiftTime	2015-08-26T14:58:47Z
[−]	MRRResult	
	SNLId	snl01649
	SNLEP	swci_rt3
[−]	FileRequestE2EControl	
	MsgId	BOX1440601100997aefaZFohOR9OQR
	CreationTime	2015-08-26T14:58:20Z
[−]	FileRequestHeader	
	Requestor	cn=aix83,o=ptsadesb,o=swift
	Responder	cn=aix83,o=ptsadesb,o=swift
	Service	swift.test.rt.iafa!x
	RequestType	admi.xxx.fa.nrf.hdf
	Priority	Normal
	RequestRef	xsys.xxy.delnotif 5
[−]	FileOpRequestHandle	
[−]	PutFileRequestHandle	
	TransferRef	SNL01649D11440600701074812S
	PossibleDuplicate	FALSE
	AckIndicator	TRUE
[−]	AckServerInfo	
	Responder	cn= test,o=ptsadesb,o=swift
	RequestType	xsys.xxy.delnotif
	LogicalName	xsys.xxy.delnotif 5
	FileInfo	SwCompression=ZIP
	Size	696
[−]	Digest	
	DigestAlgorithm	SHA-256
	DigestValue	mfXRrFqnpVF0XvPrYdQ9vgGr3TxkIe5mkZ+/2grXTI0=

The figure below shows the corresponding entry in the FACT Output Journal:



Max. Entries: 20 Creation-Date: Today

Page: 1/1 [1-16 From: 16]

Search Criterion: CreationDate EQ '2015-08-26'

Search Result:

▼	MPS↑	Recep-Time	Msg-Category	Req-Type	Service	Res-DN	Req-DN
☑	96675	8/26/15 4:58:52 PM	non.xsys.notif	xsys.xxy.delnotif	swift.test.rt.iafa!x	cn=test,o=ptsadesb,o=swift	cn=aix83,o=ptsadesb,o=swift

10 Operation

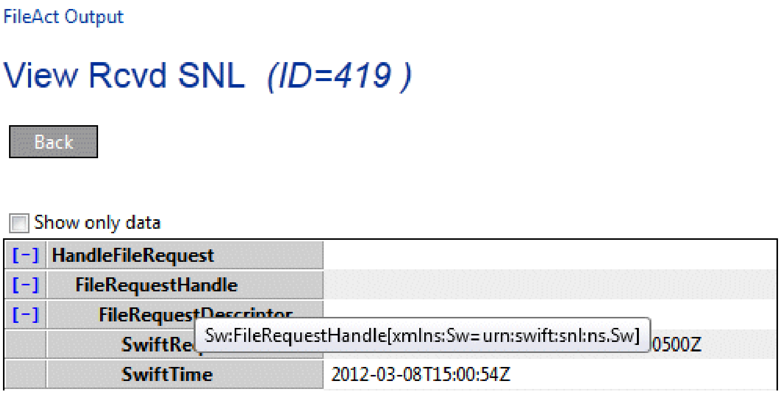
On the **MI Operations / FileAct/InterAct Channels** page the BOX web-client offers the possibility for operators and/or administrators to perform numerous operations by pressing the respective button (see below).

The logged in user performing these operations may be attached to either Enterprise (level 5) or Company (level 4) Nodes of the UPM (User Profile Management)

The visibility of the different buttons depends on the configuration (Messaging Interface Access in the UPM).

In the following sections we have not described all parameters. For detailed descriptions please refer to the corresponding SWIFT documentation (e.g. SWIFTNet 7.0 Service Design Guide and SWIFTNet 7.0 Developers Toolkit For SWIFTNet Link - Interface Specifications).

In many cases parameter descriptions or the exact name of the corresponding SWIFT primitive (tag) can be seen by holding the mouse cursor on the parameter field:



FileAct Output

View Rcvd SNL (ID=419)

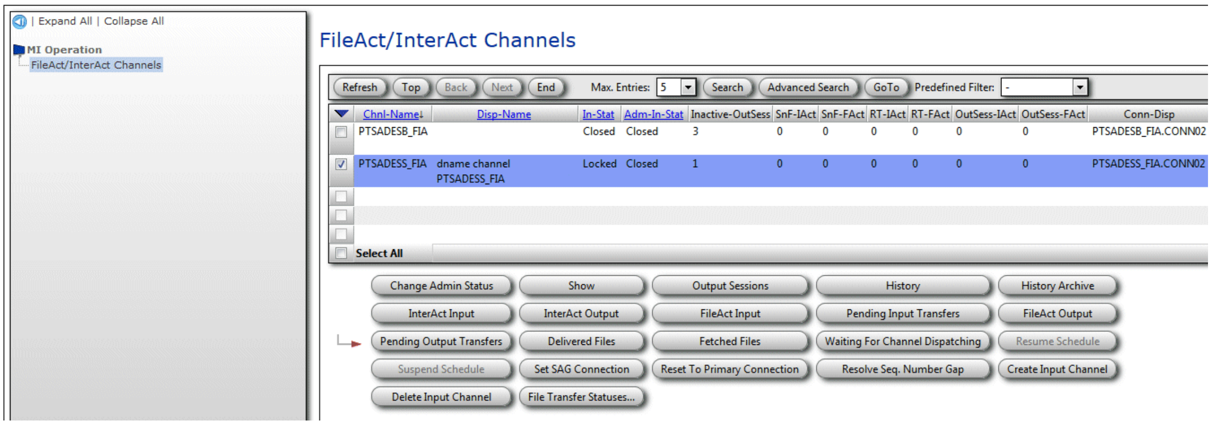
□ Show only data

[-]	HandleFileRequest	
[-]	FileRequestHandle	
[-]	FileRequestDescriptor	
	SwiftRe	0500Z
	SwiftTime	2012-03-08T15:00:54Z

Sw:FileRequestHandle[xmlns:Sw=urn:swift:snl:ns.Sw]

10.1 View Messaging Interface Channels

By selecting **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view the user can view all activities of all Channels that he is allowed to see (as configured in the User Profile Management):



FileAct/InterAct Channels

	Chnl-Name↓	Disp-Name	In-Stat	Adm-In-Stat	Inactive-OutSess	SnF-IAct	SnF-FAct	RT-IAct	RT-FAct	OutSess-IAct	OutSess-FAct	Conn-Disp
□	PTSADESB_FIA		Closed	Closed	3	0	0	0	0	0	0	PTSADESB_FIA.CONN02
☑	PTSADESS_FIA	dname channel PTSADESS_FIA	Locked	Closed	1	0	0	0	0	0	0	PTSADESS_FIA.CONN02
□												
□												
□												
□	Select All											

The table shows the following columns:

Column	Description
Chnl-Name	Name of the FileAct / InterAct channel as specified at creation time.
Disp-Name	Name of the FileAct / InterAct channel as displayed in the web-client.
In-Stat	Operative Input status of the channel.
Admin-Stat	Administrative Input status of the channel.
Inactive-OutSess	Number of inactive output messages (within current Session).
SnF-IAct	Number of InterAct Store and Forward Input messages (within current Session).
SnF-FAct	Number of FileAct Store and Forward Input messages (within current Session).
RT-IAct	Number of InterAct Real-time Input messages (within current Session).
RT-FAct	Number of FileAct Real-time Input messages (within current Session).
OutSess-IAct	Number of InterAct Real-time Output messages (within Output Session).
OutSess-FAct	Number of FileAct Real-time Output messages (within current Session).
Conn-Disp	Display name of used SAG Connection.

Additionally the user can perform selected actions on the channels.

This includes:

- Viewing details of the Channel(s)
- Viewing Channel history entries
- Viewing details on InterAct Input and Output messages handled by each Channel
- Viewing pending messages
- Viewing messages that wait for dispatching
- Changing administrative status
- Setting an SAG Connection
- Creating and deleting Input Channels
- Creating and deleting Output Channels
- Resending InterAct Input messages
- Resolving Sequence Number Gaps

In the following sections we have described some of the available action buttons.

10.2 Show Details of FileAct Channel

To view details of a selected Channel select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select a channel from the table and press **Show**.

The **FIA Channel <Channel Name>** page is displayed.

The **Details** page in the tree view shows detailed information on this Channel.

On the **Server Config** page you can view the current server configuration of this Channel.

On the **Scheduling** page you can view the currently configured scheduling options of this Channel. The parameter descriptions are described in section 4.7 of this document.

On the **Owner** page you can view the configured ownership options of this Channel.

If the logged-in user has the respective access rights specified in the UPM, he may modify the above configurations.

10.3 View Details of Output Sessions

To view details of a selected Output Session select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select a channel from the table and press **Output Sessions**.

Select an output session from the table and press **Show**.

The **Details** page in the tree view shows detailed information on this Output Session.

10.4 View Channel History

To view history entries of events, i.e. of PDUs that have passed the Messaging Interface via the selected channel, select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select a Channel from the table and press **History**.

The **FIA Channel History** page is displayed.

Select an event and press **Show**.

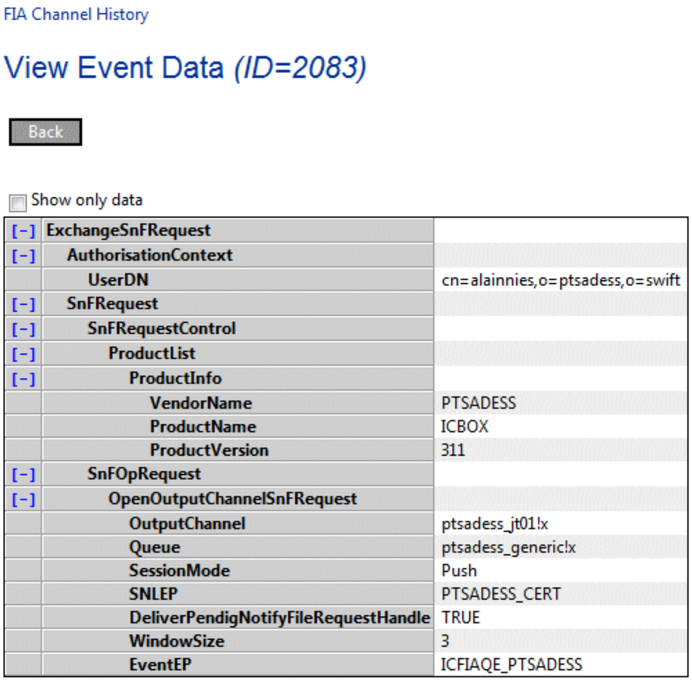
The **FIA Channel History <Channel Name>** page is displayed.

This page shows detailed information on the selected event.

10.4.1 View Event Data Content

To view only the data of a selected event press **View Event Data Content**.

The **View Event Data** showing details of the selected event is displayed.



FIA Channel History

View Event Data *(ID=2083)*

□ Show only data

[-]	ExchangeSnFRequest	
[-]	AuthorisationContext	
	UserDN	cn=alainnies,o=ptsadess,o=swift
[-]	SnFRequest	
[-]	SnFRequestControl	
[-]	ProductList	
[-]	ProductInfo	
	VendorName	PTSADESS
	ProductName	ICBOX
	ProductVersion	311
[-]	SnFOpRequest	
[-]	OpenOutputChannelSnFRequest	
	OutputChannel	ptsadess_jt01!x
	Queue	ptsadess_generic!x
	SessionMode	Push
	SNLEP	PTSADESS_CERT
	DeliverPendigNotifyFileRequestHandle	TRUE
	WindowSize	3
	EventEP	ICFIAQE_PTSADESS

10.4.2 View Response Data Content

To view only the data of a selected event press **View Response Data Content**.

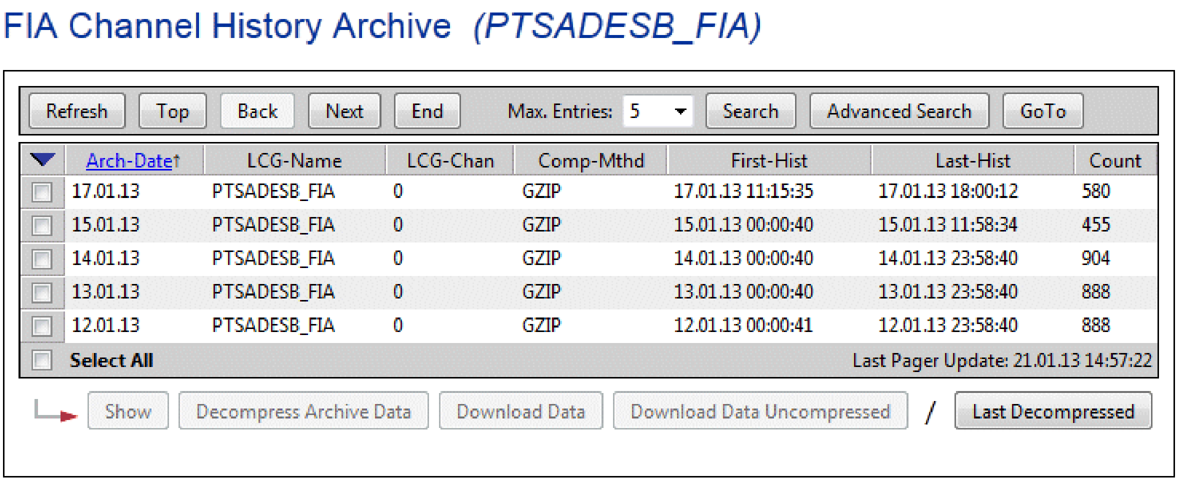
The **View Event Data** page showing details of the response to the selected event is displayed.

10.5 View Archived Channel History Entries

To view archived history entries select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select a Channel from the table and press **History Archive**.

The **FIA Channel History Archive** page is displayed:



FIA Channel History Archive *(PTSADESB_FIA)*

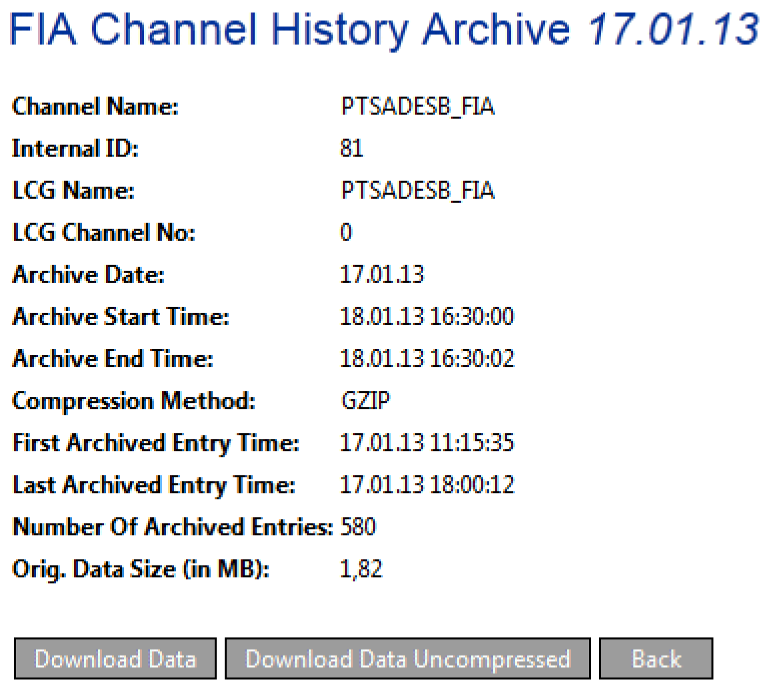
	Arch-Date↑	LCG-Name	LCG-Chan	Comp-Mthd	First-Hist	Last-Hist	Count
□	17.01.13	PTSADESB_FIA	0	GZIP	17.01.13 11:15:35	17.01.13 18:00:12	580
□	15.01.13	PTSADESB_FIA	0	GZIP	15.01.13 00:00:40	15.01.13 11:58:34	455
□	14.01.13	PTSADESB_FIA	0	GZIP	14.01.13 00:00:40	14.01.13 23:58:40	904
□	13.01.13	PTSADESB_FIA	0	GZIP	13.01.13 00:00:40	13.01.13 23:58:40	888
□	12.01.13	PTSADESB_FIA	0	GZIP	12.01.13 00:00:41	12.01.13 23:58:40	888

□ Select All

Last Pager Update: 21.01.13 14:57:22

Select the desired archive (the column **Arch-Date** shows the date on which the data was archived) and press **Show**.

The details of the selected archive are displayed:



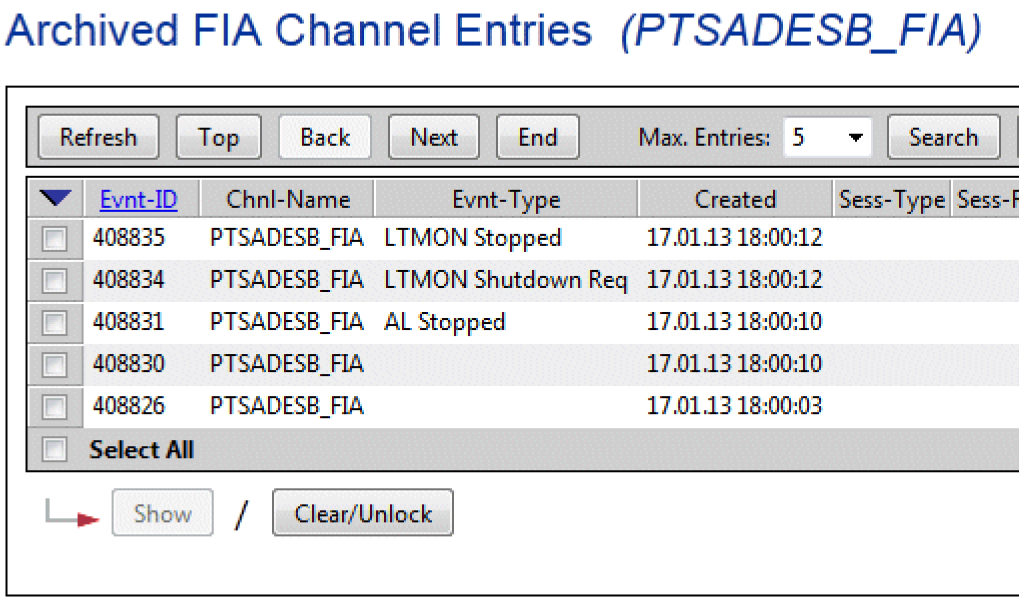
FIA Channel History Archive *17.01.13*

Channel Name:	PTSADESB_FIA
Internal ID:	81
LCG Name:	PTSADESB_FIA
LCG Channel No:	0
Archive Date:	17.01.13
Archive Start Time:	18.01.13 16:30:00
Archive End Time:	18.01.13 16:30:02
Compression Method:	GZIP
First Archived Entry Time:	17.01.13 11:15:35
Last Archived Entry Time:	17.01.13 18:00:12
Number Of Archived Entries:	580
Orig. Data Size (in MB):	1,82

If you press the **Download Data** button the data will be downloaded as stored in the database, i.e. as CLOB (Character Large Object) data. The format of the download file is ZLIB or GZIP.

If you press the **Download Data Uncompressed** button the data will be downloaded uncompressed as an xml file.

If you press the **Decompress Archive Data** (you may select multiple archiving dates) button on the **FIA Channel History Archive** page, the archived data will be loaded into an Archive Table and displayed:



Archived FIA Channel Entries *(PTSADESB_FIA)*

Max. Entries: 5

	Evnt-ID	Chnl-Name	Evnt-Type	Created	Sess-Type	Sess-F
☐	408835	PTSADESB_FIA	LTMON Stopped	17.01.13 18:00:12		
☐	408834	PTSADESB_FIA	LTMON Shutdown Req	17.01.13 18:00:12		
☐	408831	PTSADESB_FIA	AL Stopped	17.01.13 18:00:10		
☐	408830	PTSADESB_FIA		17.01.13 18:00:10		
☐	408826	PTSADESB_FIA		17.01.13 18:00:03		
☐	**Select All**					

To view details of a channel history entry select an entry and press **Show**.

10.6 View Details of File Transfer

To view details of a transferred message select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select the respective Channel and then press the **FileAct Input** or the **FileAct Output** the button, depending on whether the message you want to see the details of was an SWIFT Input or SWIFT Output message.

The **FileAct Input <channel name>** page (or **FileAct Output <channel name>** page) is displayed.

Select the respective message and press the **Show** button.

10.7 View Transfer Parameters

To view the transfer parameters of a FileAct Input message select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select the respective Channel and then press the **FileAct Input** button. The **FileAct Input <channel name>** page is displayed.

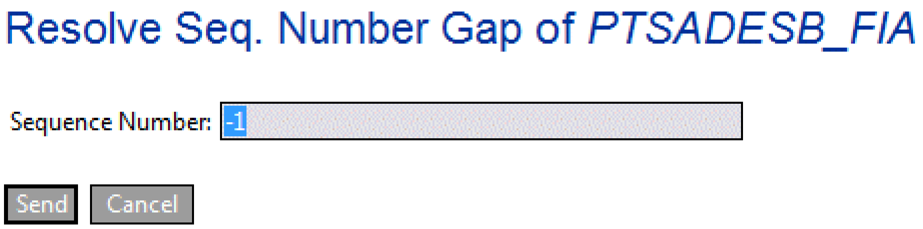
Select the respective message and press the **Transfer Parameters** button. The **Transfer Parameters (<Message ID>)** page is displayed.

10.8 Resolve Sequence Number Gap

A Sequence Number Gap occurs when no acknowledgement for a message is received (either the acknowledgement was never created by SWIFT because the message did not reach SWIFT's store-and-forward database or the acknowledgement was created by SWIFT but did not reach the Sender of the message).

In order to be able to send consecutive messages the gap has to be resolved.

To send such a resolve request to SWIFT press the **Resolve Seq. Number Gap** button.



Resolve Seq. Number Gap of *PTSADESB_FIA*

Sequence Number: -1

Enter the Input Sequence Number of the messages that was not acknowledged and press **Send**.

10.9 View File Transfer Status

To view the file transfer status of FileAct Input or FileAct Output messages, select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select the respective Channel and then press the **FileAct Input** or **FileAct Output** button.

The table on the displayed page shows the FileAct Input (or FileAct Output) messages that have been transferred via the selected channel but for which no ACK / Nak has been received yet. Among other columns, there are two status columns, **Status** and **Lst-Tr-Stat** (last transfer status).

Status describes the status reported by the Messaging Interface and can have the following values:

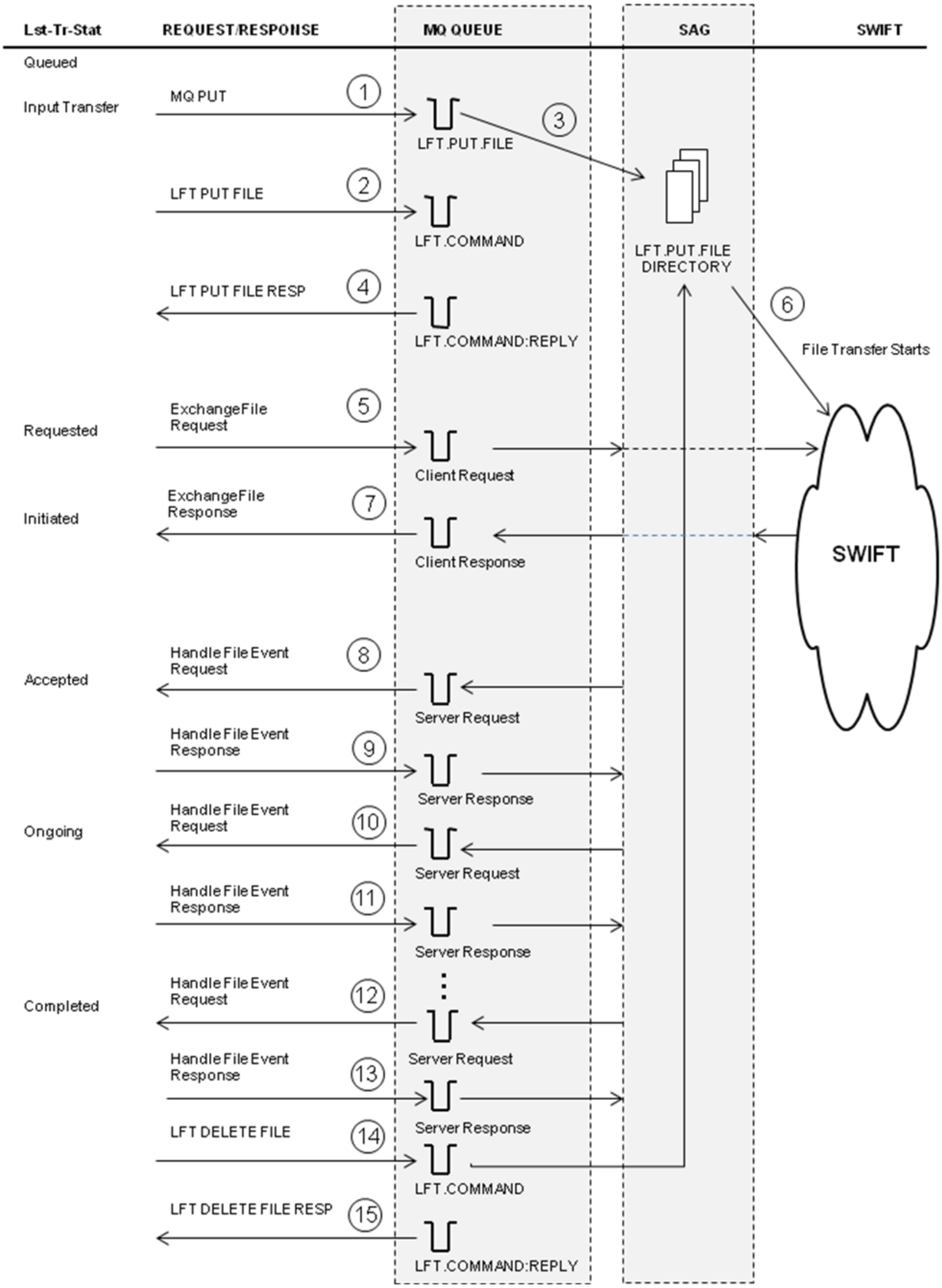
Status	Description
Queued	File transfer order queued
Dispatched	File transfer dispatched
Input Transfer	File data transferring to SAG/RFH (input messages only)
Requested	File transfer request sent to SWIFTNET
Initiated	File transfer initiated
Accepted	File transfer has been accepted
Ongoing	File data transfer in progress
Output Transfer	File data transferring from SAG/RFH (output messages only)
Problem	File transfer status in error but still pending
Completed	File transfer completed
Aborted	File transfer has been aborted
Failed	File transfer did not complete successfully
Unknown	File transfer did not complete successfully due to an unknown reason (from MI point of view).
Rejected	File transfer rejected by counterparty
Duplicated	File transfer aborted, would have been duplicate
Digest Mismatch	File transfer finished, but file digest mismatch reported

Lst-Tr-Stat describes the status reported by SWIFT and can have the following values:

Lst-Tr-Stat	Description
Initiated	File transfer initiated
Accepted	File transfer has been accepted
Ongoing	File data transfer in progress
Completed	File transfer completed
Aborted	File transfer has been aborted
Failed	File transfer did not complete successfully
Unknown	File transfer did not complete successfully due to an unknown reason (from SNL point of view).
Rejected	File transfer rejected by counterparty
Duplicated	File transfer aborted, would have been duplicate

10.9.1 File Transfer Status Overview

The figure below shows an overview of a successful file transfer using File Events and the respective status (Lst-Tr-Stat) values during the transfer. The numbering indicates the sequence of the different processing steps.



10.9.2 View Details

Press the **View Last Status** button to see the details of the file transfer status. The **View Last Status (<Message ID>)** page is displayed.

10.10 Query File Transfer Status

To send a file transfer status query select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select the respective Channel and then press the **FileAct Input** or **FileAct Output** button.

Select the message you want to query the status of and press the **Query File Transfer Status** button.

10.11 View Pending Transfers

To view pending transfers select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select a channel from the table and press **Pending Output Transfers** or **Pending Input Transfers,** respectively.

10.12 Abort File Transfer

To abort a file transfer of FileAct Input message, select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select the respective Channel and then press the **FileAct Input** button.

Select the message you want to abort and press the **Abort File Transfer** button.

10.13 Show List of Messages Waiting for Channel Dispatching

To view messages that wait for channel dispatching, select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select the respective Channel and then press the **Waiting for Channel Dispatching** button. The table showing the messages that wait for dispatching is displayed:

Sometimes there may be the need to cancel the processing of such messages.

For instance, the connection to SAG may be down (due to whatsoever reason) for this channel session and the message shall be sent via another channel. In such a case the message processing must be canceled and the message must be returned to the server in order to have it be sent by an alternative channel.

This can be achieved by selecting the respective message and then pressing the **Cancel Message** button. In this case the automatic processing is canceled and any further processing of the message may be done manually.

10.14 Change Administrative Input / Output Status

It is possible to specify an Admin Input or Output Status that will then be the 'set value' whenever the Channel is running.

If the connection to SWIFT is interrupted (due to any reason), the system will automatically try to re-establish the same status when the connection is up again without operator interference being necessary.

To change the Admin Status, select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select the Channel for which you want to change the status and then press the **Change Admin Status** button.

Select the desired status and press **Save**.

10.15 Set (Next) SAG Connection

To set an active SAG connection, select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

This function is available both for one Channel and for multiple Channels.

Select the Channel(s) for which you want to set the SAG connection and then press the **Set SAG Connection** button.

Select the desired connection from the pull-down menu and press **Save**.

10.16 Reset to Primary Connection

In cases where the primary SAG Connection of a Channel has failed, this Channel will use another SAG connection configured for this Channel (the one with the next lowest order number). This also applies to the next time the system is started, regardless of whether the Primary Connection has been fixed or not.

By pressing the **Reset to Primary Connection** button you can have the Primary Connection be used again the next time the system is started.

10.17 Create Input Channel

To create an Input Channel select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select a channel from the table and press **Create Input Channel**.

A **Create Input Channel** signal will be sent to SWIFT.

10.18 Delete Input Channel

To delete an Input Channel select **MI Operation** from the main menu and then **FileAct/InterAct Channels** from the tree-view.

Select a channel from the table and press **Delete Input Channel**.

A **Delete Input Channel** signal will be sent to SWIFT.

10.19 InterACT/FileACT Command Line Interface

The InterACT/FileACT Command Line Interface (iafacli) is a tool which can be used to interact with the IAFA channels on a command line basis.

10.19.1 Commandline Options

There are only a few command line options for the tool itself.

When no COMMAND is given the tool will run in interactive mode.

This is a subscript of the output you will get when running the cbtcli with the option '-?'.

Usage:

```
Main [OPTIONS] [COMMAND]
```

Options:

```
 -?,--help                   print this help
 -api,--configfile <file>    API configuration file, default:
                             cbtcli-api-configuration.properties
```

Commands:

```
CONNECT
SET ADMIN INPUT STATUS
SET ADMIN OUTPUT STATUS
SET CURRENT FORMAT
SET CURRENT CHANNEL
SET CURRENT SESSION
QUERY
LIST CHANNEL
LIST SESSION
PROMPT
HELP
EXIT
```

The default run batch/shell script already specifies the API config file: '-api config/iafacli-api-configuration.properties'

10.19.2 Configuration

The only configuration is done within the api configuration file iafacli-api-configuration.properties.

10.19.3 Interactive Mode

When running in interactive mode the tool will process commands until the 'EXIT' command ist given.

By default you will get a prompt.

```
IAFA-CLI >
```

When a user is connected it will change to

```
IAFA-CLI:testuser@Client1/OperatorRole >
```

10.19.4 Usage Examples

```
> run.bat
IAFA-CLI > connect client demo user testuser using secret
connected
IAFA-CLI:testuser@Client1/OperatorRole > query channel PTSADESS_FIA
State of FIA channel 'PTSADESS_FIA'
   Last Status Update Time: 2012-03-16T10:46:16Z
   Active Channel Connection: 2
   Input Status operational: CLOSED
   Input Status admin:       OPEN
   Last Replication Time: 0141:2012-03-16T08:31:29
SNF Input:
   Input Window Size: 10
   Last Input Session ID:
   Last Input Session Type: SNF_INPUT
   First ISN:      000000
   Last ISN:       000000
   Last ACKed ISN: 000000
   Input IACT Count: 0
   Input FACT Count: 0
RT Input:
   Input IACT Count: 0
   Input FACT Count: 0
FIA Output Session 'dname'
   Session Name: 'FOS-name-1340721992767'
   Last Status Update Time: 2013-03-11T16:27:33Z
   Output Status Operational: CLOSED
   Output Status Admin:       ACTIVE
   Output Window Size: 0
   Last Output Session ID:
   Last Output Session Type: UNDEFINED
   First OSN:      000000
   Last OSN:       000000
   last ACKed OSN: 000000
   Output IACT Count: 0
   Output FACT Count: 0
FIA Output Session 'dname'
   Session Name: 'FOS-name-1340722288568'
   Last Status Update Time: 2013-03-11T16:27:33Z
   Output Status Operational: CLOSED
   Output Status Admin:       ACTIVE
   Output Window Size: 0
   Last Output Session ID:
   Last Output Session Type: UNDEFINED
   First OSN:      000000
   Last OSN:       000000
   last ACKed OSN: 000000
   Output IACT Count: 0
   Output FACT Count: 0
IAFA-CLI:testuser@Client1/OperatorRole > x
bye
>
```

```
> run.bat
IAFA-CLI > connect client demo user testuser using secret; query channel
PTSADESS_FIA; x
connected
State of FIA channel 'PTSADESS_FIA'
   Last Status Update Time: 2012-03-16T10:46:16Z
   Active Channel Connection: 2
   Input Status operational: CLOSED
   Input Status admin:       OPEN
   Last Replication Time: 0141:2012-03-16T08:31:29
SNF Input:
   Input Window Size: 10
   Last Input Session ID:
   Last Input Session Type: SNF_INPUT
   First ISN:      000000
   Last ISN:       000000
   Last ACKed ISN: 000000
   Input IACT Count: 0
   Input FACT Count: 0
RT Input:
   Input IACT Count: 0
   Input FACT Count: 0
FIA Output Session 'dname'
   Session Name: 'FOS-name-1340721992767'
   Last Status Update Time: 2013-03-11T16:27:33Z
   Output Status Operational: CLOSED
   Output Status Admin:       ACTIVE
   Output Window Size: 0
   Last Output Session ID:
   Last Output Session Type: UNDEFINED
   First OSN:      000000
   Last OSN:       000000
   last ACKed OSN: 000000
   Output IACT Count: 0
   Output FACT Count: 0
FIA Output Session 'dname'
   Session Name: 'FOS-name-1340722288568'
   Last Status Update Time: 2013-03-11T16:27:33Z
   Output Status Operational: CLOSED
   Output Status Admin:       ACTIVE
   Output Window Size: 0
   Last Output Session ID:
   Last Output Session Type: UNDEFINED
   First OSN:      000000
   Last OSN:       000000
   last ACKed OSN: 000000
   Output IACT Count: 0
   Output FACT Count: 0
bye
>
```

```
> run.bat connect client demo user testuser using secret; set current
format m; query channel PTSADESS_FIA; x
connected
OK:
QUERY CHANNEL=PTSADESS_FIA
LAST STATUS UPDATE TIME=2012-03-16T10:46:16Z
ACTIVE CHANNEL CONNECTION=2
INPUT STATUS OPERATIONAL=CLOSED
INPUT STATUS ADMIN=OPEN
LAST REPLICATION TIME=0141:2012-03-16T08:31:29
SNF INPUT:
INPUT WINDOW SIZE=10
LAST INPUT SESSION ID=
LAST INPUT SESSION TYPE=SNF_INPUT
FIRST ISN=000000
LAST ISN=000000
LAST ACKED ISN=000000
INPUT IACT COUNT=0
INPUT FACT COUNT=0
RT INPUT:
INPUT IACT COUNT=0
INPUT FACT COUNT=0
FIA OUTPUT SESSION:
SESSION NAME=FOS-name-1340721992767
SESSION DISPLAY NAME=dname
LAST STATUS UPDATE TIME=2013-03-11T16:27:33Z
OUTPUT STATUS OPERATIONAL=CLOSED
OUTPUT STATUS ADMIN=ACTIVE
OUTPUT WINDOW SIZE=0
LAST OUTPUT SESSION ID=
LAST OUTPUT SESSION TYPE=UNDEFINED
FIRST OSN=000000
LAST OSN=000000
LAST ACKED OSN=000000
OUTPUT IACT COUNT=0
OUTPUT FACT COUNT=0
FIA OUTPUT SESSION:
SESSION NAME=FOS-name-1340722288568
SESSION DISPLAY NAME=dname
LAST STATUS UPDATE TIME=2013-03-11T16:27:33Z
OUTPUT STATUS OPERATIONAL=CLOSED
OUTPUT STATUS ADMIN=ACTIVE
OUTPUT WINDOW SIZE=0
LAST OUTPUT SESSION ID=
LAST OUTPUT SESSION TYPE=UNDEFINED
FIRST OSN=000000
LAST OSN=000000
LAST ACKED OSN=000000
OUTPUT IACT COUNT=0
OUTPUT FACT COUNT=0
>
```

10.19.5 Commands

Some command support a FORCE option which disables the command pending check.

Otherwise the command will fail if there is a command pending for the given LT.

```
CONNECT
SET ADMIN INPUT STATUS
SET ADMIN OUTPUT STATUS
SET CURRENT FORMAT
SET CURRENT CHANNEL
SET CURRENT SESSION
QUERY
LIST CHANNEL
LIST SESSION
PROMPT
HELP
EXIT
```

Note: **Multiple commands** can be entered using `';'` as separator (as can be seen in one of the examples above).

10.19.5.1 CONNECT

Used to connect a user to the database. This user is used to check access to the underlying objects (LTs, connections etc).

If the given user has multiple context paths, the default context is used.

Syntax:

```
CONNECT [CLIENT | CLT | C] <clt prefix> [USER | U] <username> [USING |
P] [ENCRYPTED] <password>
```

Examples:

```
CONNECT CLIENT demo USER testuser USING ENCRYPTED a67074b10cc2d676

CONNECT C demo U testuser P secret
```

10.19.5.2 SET ADMIN INPUT STATUS

Changes the administrative INPUT status of the given channel.

Syntax:

```
SET ADMIN INPUT STATUS [CHANNEL <name>] [OPEN | FORCED | CLOSED |
ABORTED]
```

Examples:

```
SET ADMIN INPUT STATUS CHANNEL PTSADESS_FIA OPEN

SET ADMIN STATUS OPEN
```

10.19.5.3 SET ADMIN OUTPUT STATUS

Changes the administrative OUTPUT status of one or all of the output sessions of the current or given channel.

If the SESSION is omitted the status will be set for all available session of the current or given channel.

If you want to change the status of a particular session you must specify both CHANNEL and SESSION.

Syntax:

```
SET ADMIN OUTPUT STATUS [CHANNEL <name> [SESSION <name>]] [ACTIVE |
FORCED | CLOSED | ABORTED]
```

Examples:

```
SET ADMIN OUTPUT STATUS CHANNEL PTSADESS_FIA ACTIVE

SET ADMIN OUTPUT STATUS CLOSED
```

10.19.5.4 SET CURRENT FORMAT

Changes the current output format.

This is primarily used with the QUERY command and specifies whether to print a more machine readable output rather than human readable.

Syntax:

```
SET CURRENT FORMAT [[MACHINE | M] | [HUMAN | H]]
```

Example:

```
SET CURRENT FORMAT M
```

10.19.5.5 SET CURRENT CHANNEL

Sets the current channel field within the current session.

When set the current channel will used in all commands unless a channel is provided.

Syntax:

```
SET CURRENT CHANNEL <name>
```

Example:

```
SET CURRENT CHANNEL PTSADESS_FIA
```

10.19.5.6 QUERY

Queries status information for the given channel and its output sessions.

Syntax:

```
QUERY [CHANNEL <name>]
```

Examples:

```
QUERY CHANNEL PTSADESS_FIA

QUERY
```

10.19.5.7 LIST CHANNEL

Displays a list of all channels available to the current user.

Syntax:

```
LIST CHANNEL
```

Example:

```
LIST CHANNEL
```

10.19.5.8 LIST SESSION

Displays a list of all output sessions of the current or given channel.

Syntax:

```
LIST SESSION [CHANNEL <name>]
```

Examples:

```
LIST SESSION CHANNEL PTSADESS_FIA
```

```
LIST SESSION
```

10.19.5.9 PROMPT

Enables or disables the prompt.

If no option is given the value will be toggled.

Syntax:

```
PROMPT [ON | OFF]
```

Examples:

```
PROMPT
```

```
PROMPT OFF
```

10.19.5.10 EXIT

Quits the tool.

Syntax:

```
[EXIT | X]
```

Examples:

```
EXIT
```

```
X
```

10.19.5.11 HELP

Prints help messages. If no specific command is given a list of available commands is printed out.

Syntax:

```
HELP [COMMAND]
```

Examples:

```
HELP

HELP CONNECT
```

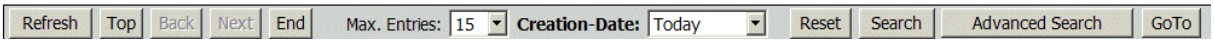
11 FileAct Journals

The BOX FileAct Implementation offers dedicated journals for sent and received FileAct messages. These journals can be found by selecting **FileAct** from the main menu and then **Journals** from the tree-view.

Additionally it is possible to create own Journal views, either by users themselves (these views will only be available for the user who created them) or by administrators, which can create views for whole groups or departments, etc.

The Journals provide detailed information for each message, search capabilities to find specific messages quickly and message handling functions (e.g. routing, retransmit, interrupt)

Journals show the following buttons and filters:



Max. Entries: 15

Creation-Date: Today

The following table describes the functions of the available buttons and fields:

Button / Filter	Description
Refresh	Pressing this button refreshes the view.
Top	Pressing this button displays the entries on the top of the list as specified by the view.
Back	Pressing this button displays the previous entries in the list. If the value for **Max. Entries:** is **10**, the previous 10 entries in the list would be displayed.
Next	Pressing this button displays the next entries in the list. If the value for **Max. Entries:** is **10**, the next 10 entries in the list would be displayed.
End	Pressing this button displays the users on the bottom of the list as specified by the filter. The number of entries displayed depends on the value for **Max. Entries:**.
Max. Entries:	This menu allows you to select how many entries shall be displayed.
Creation Date	Date Filter. You must select the creation date (or time range) to have the entries be displayed.
Search	This button is for retrieval purposes. How to use the Search option is described in the document **BOX Message Entry & Repair User Manual**.
Advanced Search (optional)	This button is for retrieval purposes. As this feature is optional the button may not be visible in your individual installation.
Reset	The **Reset** button is shown only if a search is active. Be very careful in such a case as you usually will see only the messages that fit to the search criteria and not all messages in the queue. Press the **Reset** button to disable the search immediately. All messages in the queue will be shown again.
GoTo (optional)	This button also is for retrieval purposes. As this feature is optional the button may not be visible in your individual installation.

11.1 FACT Input Journal

This journal displays all SWIFT Input messages from the Web-Client or from other applications that have been sent via BOX to SWIFT after all processing.

The FACT Input journal columns provide the following information:

Column	Description
MPS	Unique BOX MPS (message) ID.
Recep-Time	Reception time. The time when the message was received by BOX (created from client or received from a backend application).
Del-Stat-SDD	Delivery status of a Single Delivery Destination (SDD)
Msg-Category	Message category
Service	SWIFT Service
Req-DN	Requestor DN
Res-DN	Responder DN
Logic-File-Name	Logical file name
File-Size	File size
Res-Code-SDA	Result code of delivery to a Single Destination Address (SDA)
Status-Description	Status description. Information is set by workflow, e.g. by a TGI. Displays information from the **External Status Description** field of the Instruction.
Appl-Q-ID	Application Queue ID. ID of Application Queue where the message is currently located.
Stat-Val	Status value
PD-Flag	Possible duplicate flag
PDE	Possible duplicate emmission
PDM	Possible duplicate message
Comment	Comment
Chnl-Type	Channel type used for message transfer to SWIFT.
Details-Code	Device-specific information, e.g. result data provided by Analysis Module.
Deliv-Start-Time	Time when delivery to SWIFT started
DeliveryResultCode	Delivery result code
Deliv-End-Time	Time when delivery to SWIFT ended
Msg-ID	Message ID. The value is taken from the **Message Identifier** field of the End-to-End Control Data in the **FileAct - Put File** Submission Profile.
Source-Appl	Source application.

11.2 FACT Output Delivery

This journal shows all SWIFT Output messages that have been received by BOX and that have been delivered to other applications.

The FACT Output Delivery journal columns provide the following information:

Column	Description
MPS	Unique BOX MPS (message) ID
Recep-Time	Reception time. The time when the message was received by BOX (from SWIFT)
Msg-Category	Message category
Req-Type	Request type
Service	SWIFT Service
Res-DN	Responder DN
Req-DN	Requestor DN
Logic-File-Name	Logical file name
Out-Sess-ID	Output Session ID
Out-Seq-No	Output Sequence Number
Comment	Comment
File-Size	File size
Del-Stat-SDD	Delivery status of a Single Delivery Destination (SDD)
Rec-Disp-Addr	Recipient display address
Orig-Mode	Origination mode
Del-Stat-SDA	Delivery status of delivery to a Single Destination Address (SDA).
Addr-Type	Address type used for delivery (to other application).
Chnl-Type	Channel type used for delivery (to other application).
DeliveryResultCode	Delivery result code.
Deliv-Start-Time	Time when delivery to other application started
Deliv-End-Time	Time when delivery to other application ended
Creation-Date	Creation date. The date the MPS (containing the data of the SWIFT Output message) was created by BOX.
Creat-Time	Creation time. The time the MPS (containing the data of the SWIFT Output message) was created by BOX.
PD-Flag	Possible duplicate flag
Msg-ID	Message ID. Value is taken from the Message Reference parameter of the SWIFT Output message.
PDE	Possible duplicate emmission.
PDM	Possible duplicate message.
Last-Trans-Stat	Last transmission status.

11.3 FACT Output Journal

This journal displays all SWIFT Output messages that have been received by BOX.

The FACT Output journal columns provide the following information:

Column	Description
MPS	Unique BOX MPS (message) ID.
Recep-Time	The time when the message was received by BOX (from SWIFT).
Msg-Category	Message category
Req-Type	Request type
Service	SWIFT Service
Res-DN	Responder DN
Req-DN	Requestor DN
Logic-File-Name	Logical file name
Status-Description	Status description. Information is set by workflow, e.g. by a TGI. Displays information from the **External Status Description** field of the Instruction.
Appl-Q-ID	Application Queue ID. ID of Application Queue where the message is currently located.
Out-Sess-ID	Output Session ID
Out-Seq-No	Output Sequence Number
Comment	Comment
File-Size	File size
Del-Stat-SDD	Delivery status of a Single Delivery Destination (SDD)
Stat-Val	Status value
PD-Flag	Possible duplicate flag
Rec-Disp-Addr	Recipient display address
Deliv-Start-Time	Delivery start time. Time when delivery to backend application started.
Deliv-End-Time	Delivery end time. Time when delivery to backend application ended.
Del-Stat-SDA	Delivery status of delivery to a Single Destination Address (SDA).
Creat-Time	Creation time. The time the MPS (containing the data of the SWIFT Output message) was created by BOX.
Msg-ID	Message ID. Value is taken from the Message Reference parameter of the SWIFT Output message.

11.4 FACT SWIFT-ACK Delivery

This journal displays all SWIFT ACKs that have been received by BOX and then been routed to a backend application.

The FACT SWIFT ACK Delivery journal columns provide the following information:

Column	Description
MPS	Unique BOX MPS (message) ID.
Recep-Time	Reception time. The time when the original message was received by BOX (from client or backend application).
Res-DN	Responder DN
Req-DN	Requestor DN
Service	SWIFT Service
Rec-Disp-Addr	Recipient display address
Comment	Comment
Orig-Mode	Origination mode
Del-Stat-SDA	Delivery status of delivery to a Single Destination Address (SDA).
Addr-Type	Address type used for delivery (to other application)
Chnl-Type	Channel type used for delivery (to other application)
DeliveryResultCode	Delivery result code
Out-Seq-No	Output Sequence Number
Out-Sess-ID	Output Session ID
Deliv-Start-Time	Delivery start time. Time when delivery of ACK started
Deliv-End-Time	Delivery end time. Time when delivery of ACK ended
Creation-Date	Creation date. The date the original MPS (SWIFT Input message) was created.
Creat-Time	Creation time. The time the original MPS (SWIFT Input message) was created.
Msg-ID	Message ID. Value is taken from the Message Reference parameter of the SWIFT Output message.
PD-Flag	Possible duplicate flag
PDE	Possible duplicate emmission
PDM	Possible duplicate message

12 Cold Start

Under exceptional circumstances, SWIFT may have to restore the SWIFT Main Services from an empty or zeroed state (cold start).

After a cold start:

- No messages or message data that were in the SWIFTNet systems before the cold start are available.
  SWIFT will not recover these messages when it restores the SWIFTNet service.
  The information that is unavailable and unrecoverable includes the following messages and data:
  - All positively acknowledged messages that were in queue and awaiting delivery to the intended recipient. SWIFT makes no further delivery attempts for these messages.
  - All historic data related to message receipt by SWIFT before the cold start, or message delivery by SWIFT before the cold start.
- SWIFT resets all Input Sequence Numbers (ISNs) and Output Sequence Numbers (OSNs) to zero, for all Logical Terminals (LTs).

12.1 Required User Actions

Output channels that were created must be re-created, except for the generic output channels that SWIFT automatically makes available.

Input channels must be re-created, except for the generic input channels that SWIFT automatically makes available.

12.1.1 Identify Possible Duplicate SWIFT Output Messages

Make sure that the messages received with a “Possible Duplicate Indication” (PDI) are only acted upon once as per standard processing of PDI.

12.1.2 Identify and Resend Affected SWIFT Input Messages/Files

Identify any messages/files that have been sent, but for which the delivery status to the recipient is uncertain. This also applies to messages/files which were initially acknowledged by SWIFT, because they may not have been delivered yet at the time of the incident.

The starting point for the identification of affected traffic is the system message Unsolicited Undelivered Traffic Report (xsys.005.002), with CS tag, which contains information about the situation before the cold start.

SWIFT generates this report, in the form of one or more system messages, automatically after a cold start and makes it available in the user's generic queue of each BIC 8 (pilot queues for the test).

The time (T) mentioned in the report should be used to identify the messages/files that need to be retransmitted.

Resend the identified messages/files with a PDI indication (and if possible the MsgId and CreationTime of the original transmission).

The messages/files to be resent with “Possible Duplicate Indication” are:

- Messages/files sent at or after the xsys.005.002 Unsolicited Undelivered Traffic Report Date/Time indicated (last capture) which have been acknowledged by SWIFT.
- Messages/files listed in the xsys.005.002 report for the BIC 8 and which have not yet been delivered to the correspondent.
- Messages/files for which the delivery status was unclear – for example, sent to SWIFT but for which a network acknowledgement is still to be received.

There are some exceptions:

- Messages/files sent in the context of a SWIFTNet Copy service, for which the service administrator gave instructions not to resend.
- Messages/files for which the sender has selected the delivery notification option, and for which the sender has received a Delivery Notification from SWIFT indicating a successful delivery to the intended recipient.
- Messages/files for which the sender can be sure that the intended recipient received successfully (for example, because it has resulted in the sender receiving a message).

If there is any reasonable doubt about an exception, then it is always safe to resend a message/file with a possible duplicate indicator.

If a Reconciliation System as described below has been set up, the xsys.005.002 messages can be found in a queue specially created for these messages.

12.2 Set up of the SWIFT Coldstart Reconciliation for FileAct

For setting up a Coldstart Reconciliation system for FileAct messages, the workflow configuration was modified as described below.

Note: For testing purposes we configured the system to handle xsys.005.00x.0x messages and named queues and IPS accordingly. In real life xsys.005.002 messages will be received and the configuration must be modified accordingly.

Two new IPS, **IACT_ToReceived_xsys.005.00x.0x** and **IAFA_ToColdStartResend** were created.

The IPS **IAFA_ToColdStartResend** consists of a TGI pointing to the application queue **FACT Cold Restart Resend** (see below).

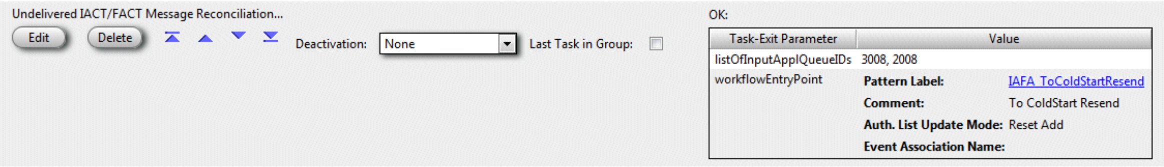
The IPS **IACT_ToReceived_xsys.005.00x.0x** consists of a TGI pointing to the application queue **Received-xsys.005.00x.0x** (see below)

Two new Application Queue Definitions were created, **FACT-ColdstartResend** and **Received-xsys.005.00x.0x**.

These are the application queues for received xsys.005.00x.0x messages (MX Received-xsys.005.001.01) and for reconciliated FileAct messages that shall be resent (FACT Cold Restart Resend).

Two new Views and Tasks definitions were created.

In the queue **Received-xsys.005.00x.0x** the Selection Task **Undelivered IACT/FACT Message Reconciliation** was configured.
The Pattern Label of the Workflow Entry Point points to the IPS **IAFA_ToColdstartResend**.



Undelivered IACT/FACT Message Reconciliation...

Deactivation: None

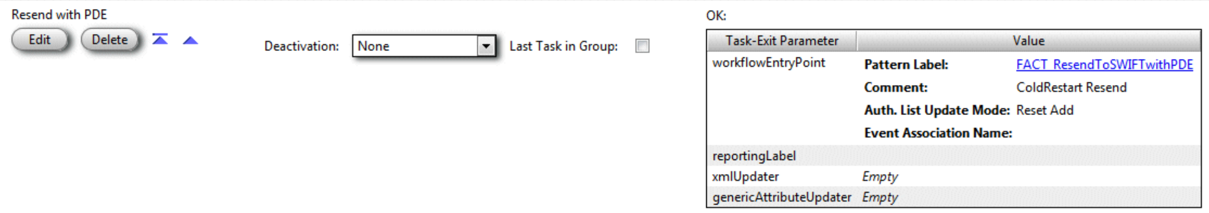
Last Task in Group: □

OK:

Task-Exit Parameter	Value	
listOfInputApplQueueIDs	3008, 2008	
workflowEntryPoint	**Pattern Label:**	IAFA_ToColdStartResend
	Comment:	To ColdStart Resend
	Auth. List Update Mode:	Reset Add
	Event Association Name:	

The list of queues in which the undelivered messages to be reconciliated shall be searched for (parameter listOfInputApplQueueIDs) was amended. This is because the task is used for the reconciliation of both InterAct (ApplQueueID 2008) and FileAct messages (ApplQueueID 3008).

In the queue **FACT ColdRestart Resend** the Selection Task **Resend with PDE** was configured. The Pattern Label of the Workflow Entry Point points to the IPS **FACT_ResendToSWIFTwithPDE**.



Resend with PDE

Deactivation: None

Last Task in Group: ☐

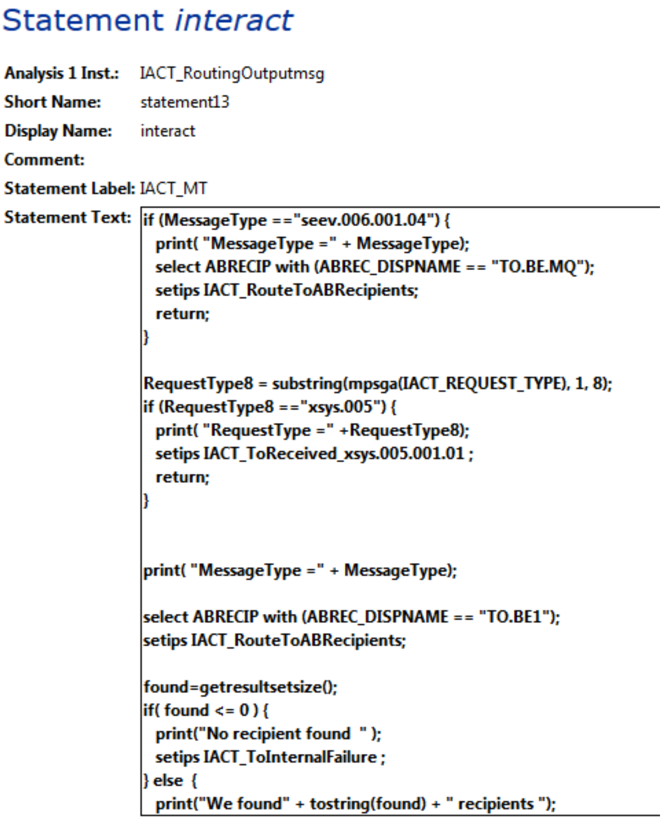
OK:

Task-Exit Parameter	Value	
workflowEntryPoint	**Pattern Label:**	FACT_ResendToSWIFTwithPDE
	Comment:	ColdRestart Resend
	Auth. List Update Mode:	Reset Add
	Event Association Name:	
reportingLabel		
xmlUpdater	*Empty*	
genericAttributeUpdater	*Empty*	

Routing Definition for the SWIFT Output messages was modified.

All xsys.005.00x.0x messages shall be routed to the application queue **MX Received-xsys.005.00x.0x** (which is the Reconciliation Queue). For this purpose the Statement was modified by inserting the following section into the Statement:

```
RequestType8 = substring(mpsga(IACT_REQUEST_TYPE), 1, 8);
if (RequestType8 =="xsys.005") {
   print( "RequestType =" +RequestType8);
   setips IACT_ToReceived_xsys.005.00x.0x  ;
   return;
}
```



Statement *interact*

Analysis 1 Inst.: IACT_RoutingOutputmsg

Short Name: statement13

Display Name: interact

Comment:

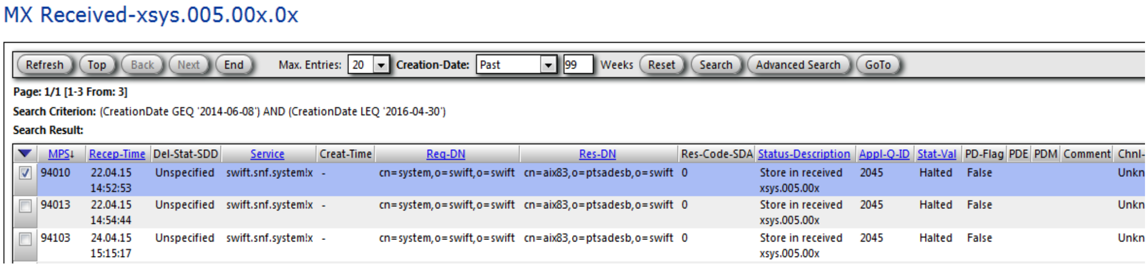
Statement Label: IACT_MT

Statement Text:

```
if (MessageType =="seev.006.001.04") {
  print( "MessageType =" + MessageType);
  select ABRECIP with (ABREC_DISPNAME == "TO.BE.MQ");
  setips IACT_RouteToABRecipients;
  return;
}

RequestType8 = substring(mpsga(IACT_REQUEST_TYPE), 1, 8);
if (RequestType8 =="xsys.005") {
  print( "RequestType =" +RequestType8);
  setips IACT_ToReceived_xsys.005.001.01 ;
  return;
}


print( "MessageType =" + MessageType);

select ABRECIP with (ABREC_DISPNAME == "TO.BE1");
setips IACT_RouteToABRecipients;

found=getresultsetsize();
if( found <= 0 ) {
  print("No recipient found  ");
  setips IACT_ToInternalFailure ;
} else  {
  print("We found" + tostring(found) + " recipients ");
```

12.3 Processing Reconciliation Messages.

All xsys.005.00x.0x messages are now kept in the queue **MX Received-xsys.005.00x.0x** (Menu: MX/Miscellaneous Queues/…).



Page: 1/1 [1-3 From: 3]

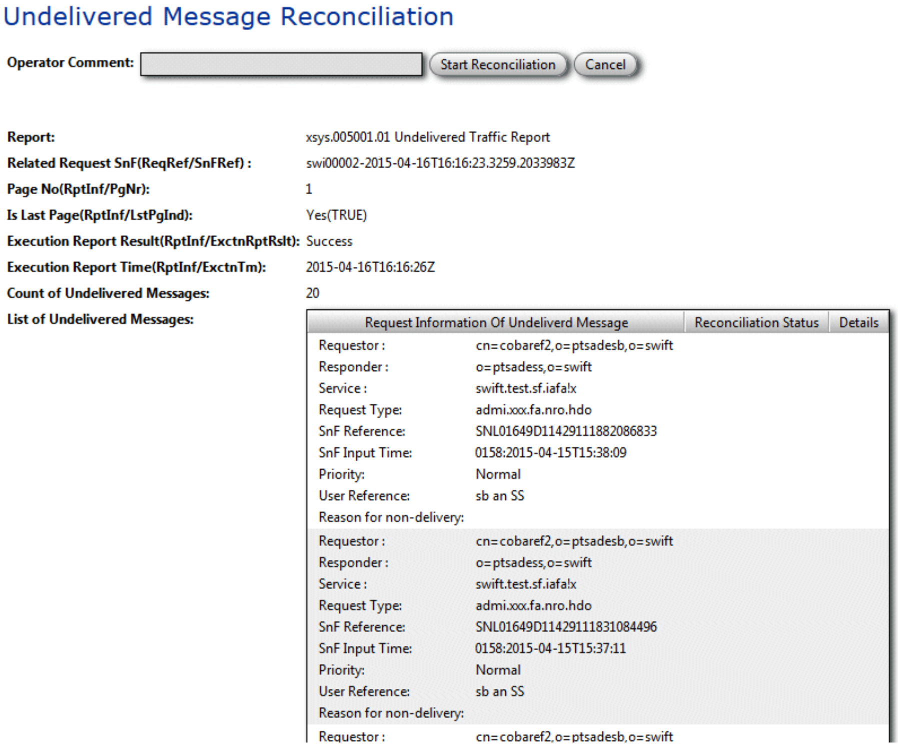
Search Criterion: (CreationDate GEQ '2014-06-08') AND (CreationDate LEQ '2016-04-30')

Search Result:

	MPS↓	Recep-Time	Del-Stat-SDD	Service	Creat-Time	Req-DN	Res-DN	Res-Code-SDA	Status-Description	Appl-Q-ID	Stat-Val	PD-Flag	PDE	PDM	Comment	Chnl-
☑	94010	22.04.15 14:52:53	Unspecified	swift.snf.system!x	-	cn=system,o=swift,o=swift	cn=aix83,o=ptsadesb,o=swift	0	Store in received xsys.005.00x	2045	Halted	False				Unkn
☐	94013	22.04.15 14:54:44	Unspecified	swift.snf.system!x	-	cn=system,o=swift,o=swift	cn=aix83,o=ptsadesb,o=swift	0	Store in received xsys.005.00x	2045	Halted	False				Unkn
☐	94103	24.04.15 15:15:17	Unspecified	swift.snf.system!x	-	cn=system,o=swift,o=swift	cn=aix83,o=ptsadesb,o=swift	0	Store in received xsys.005.00x	2045	Halted	False				Unkn

In the queue **MX Received-xsys.005.00x.0x** you can select a xsys.005.00x.0x message to process and click on **Undelivered IACT/FACTMessage Reconciliation….**

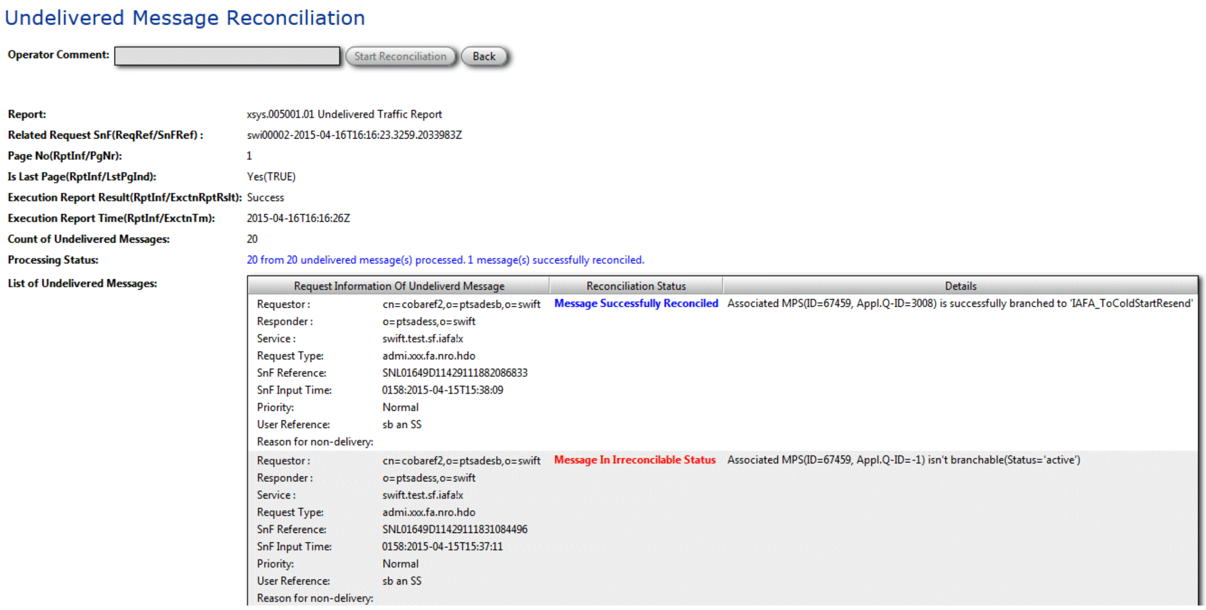
The **Undelivered Message Reconciliation** page is displayed:



Report:	xsys.005001.01 Undelivered Traffic Report
Related Request SnF(ReqRef/SnFRef) :	swi00002-2015-04-16T16:16:23.3259.2033983Z
Page No(RptInf/PgNr):	1
Is Last Page(RptInf/LstPgInd):	Yes(TRUE)
Execution Report Result(RptInf/ExctnRptRslt):	Success
Execution Report Time(RptInf/ExctnTm):	2015-04-16T16:16:26Z
Count of Undelivered Messages:	20
List of Undelivered Messages:	

Request Information Of Undeliverd Message		Reconciliation Status	Details
Requestor :	cn=cobaref2,o=ptsadesb,o=swift		
Responder :	o=ptsadess,o=swift		
Service :	swift.test.sf.iafa!x		
Request Type:	admi.xxx.fa.nro.hdo		
SnF Reference:	SNL01649D11429111882086833		
SnF Input Time:	0158:2015-04-15T15:38:09		
Priority:	Normal		
User Reference:	sb an SS		
Reason for non-delivery:			
Requestor :	cn=cobaref2,o=ptsadesb,o=swift		
Responder :	o=ptsadess,o=swift		
Service :	swift.test.sf.iafa!x		
Request Type:	admi.xxx.fa.nro.hdo		
SnF Reference:	SNL01649D11429111831084496		
SnF Input Time:	0158:2015-04-15T15:37:11		
Priority:	Normal		
User Reference:	sb an SS		
Reason for non-delivery:			
Requestor :	cn=cobaref2,o=ptsadesb,o=swift		

The content of the xsys.005.00x.0x and a list of all messages that were not delivered is shown. Now you can enter a comment and press the **Start Reconciliation** button.

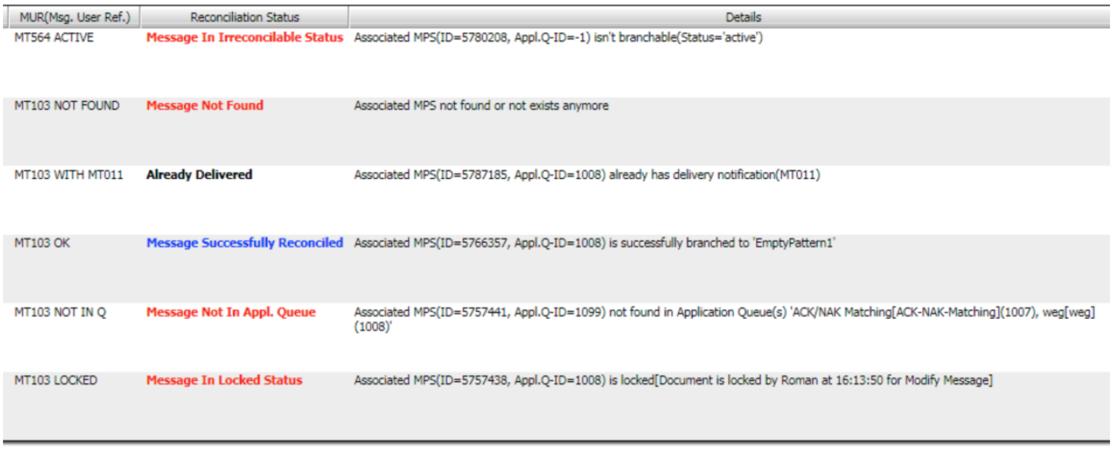


Undelivered Message Reconciliation

Report:	xsys.005001.01 Undelivered Traffic Report
Related Request SnF(ReqRef/SnFRef) :	swi00002-2015-04-16T16:16:23.3259.2033983Z
Page No(RptInf/PgNr):	1
Is Last Page(RptInf/LstPgInd):	Yes(TRUE)
Execution Report Result(RptInf/ExctnRptRslt):	Success
Execution Report Time(RptInf/ExctnTm):	2015-04-16T16:16:26Z
Count of Undelivered Messages:	20
Processing Status:	20 from 20 undelivered message(s) processed. 1 message(s) successfully reconciled.
List of Undelivered Messages:	

Request Information Of Undeliverd Message		Reconciliation Status	Details
Requestor : Responder : Service : Request Type: SnF Reference: SnF Input Time: Priority: User Reference: Reason for non-delivery:	cn=cobaref2,o=ptsadesb,o=swift o=ptsadess,o=swift swift.test.sf.iafa!x admi.xxx.fa.nro.hdo SNL01649D11429111882086833 0158:2015-04-15T15:38:09 Normal sb an SS	**Message Successfully Reconciled**	Associated MPS(ID=67459, Appl.Q-ID=3008) is successfully branched to 'IAFA_ToColdStartResend'
Requestor : Responder : Service : Request Type: SnF Reference: SnF Input Time: Priority: User Reference: Reason for non-delivery:	cn=cobaref2,o=ptsadesb,o=swift o=ptsadess,o=swift swift.test.sf.iafa!x admi.xxx.fa.nro.hdo SNL01649D11429111831084496 0158:2015-04-15T15:37:11 Normal sb an SS	**Message In Irreconcilable Status**	Associated MPS(ID=67459, Appl.Q-ID=-1) isn't branchable(Status='active')

The following states are possible:



MUR(Msg. User Ref.)	Reconciliation Status	Details
MT564 ACTIVE	**Message In Irreconcilable Status**	Associated MPS(ID=5780208, Appl.Q-ID=-1) isn't branchable(Status='active')
MT103 NOT FOUND	**Message Not Found**	Associated MPS not found or not exists anymore
MT103 WITH MT011	**Already Delivered**	Associated MPS(ID=5787185, Appl.Q-ID=1008) already has delivery notification(MT011)
MT103 OK	**Message Successfully Reconciled**	Associated MPS(ID=5766357, Appl.Q-ID=1008) is successfully branched to 'EmptyPattern1'
MT103 NOT IN Q	**Message Not In Appl. Queue**	Associated MPS(ID=5757441, Appl.Q-ID=1099) not found in Application Queue(s) 'ACK/NAK Matching[ACK-NAK-Matching](1007), weg[weg](1008)'
MT103 LOCKED	**Message In Locked Status**	Associated MPS(ID=5757438, Appl.Q-ID=1008) is locked[Document is locked by Roman at 16:13:50 for Modify Message]

All successfully reconciliated messages are routed to the queue **FACT ColdRestart Resend** (Menu: FileAct/Miscellaneous Queues/…).

After selecting a message the **Resend with PDE** is activated. By pressing this button, the dialog to re-send the message(s) is opened.

Now you can check the message(s) and send them again.

All messages are enriched with a qualified PDI.

After having resent the affected messages, you may resume normal operation.

13 Disclaimer

INTERCOPE International Communication Products Engineering GmbH (Intercope) and the stylized logo is the registered trademark of Intercope and its subsidiaries, in Germany and certain other countries. All other trademarks mentioned in this document are the acknowledged property of their respective owners.

Intercope provides this publication "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of non-infringement, merchantability or fitness for a particular purpose.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Intercope may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information may contain sample application programs in source language, which illustrate programming and implementation techniques. You may copy, modify, and distribute these samples programs in any form without payment to Intercope, for the purposes of developing, using, marketing or distributing application programs for which the sample programs are written. These examples have not been thoroughly tested under all conditions. Intercope, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.
The sample programs are provided "AS IS", without warranty of any kind. Intercope shall not be liable for any damages arising out of use of the sample programs.

Intercope grants the right to reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. Intercope does not allow derivative works of these publications, or to reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of Intercope.

Without written permission of Intercope no part of this publication may be modified and/or reproduced in any way.

INTERCOPE GmbH
Himmelstrasse 12-16,
22299 Hamburg,
Germany

+49 40 514 52 0
info@intercope.com

https://www.intercope.com