

FICO Inbound Interface: IDC Interface to SAP



Applies to:

FICO Inbound Interface. For more information, visit the [Enterprise Resource Planning homepage](#).

Summary

1. SER software scans in the invoices received and formats into the file, then places this file on the secure directory depending on the PO number range (different range for each box).
2. GLOFFI polls directories for any new files.
3. GLOFFI wrapper program adds header and footer
4. GLOFFI places file on secure directory for MW to pick up automatically
5. MW picks up file and places on designated secure directory on SAP application server (zdata directories).
6. The SAP custom developed wrapper program /EUR/FGBFCI_SER_INV_POST_PARK is triggered by an RFC call made by the adapter script the moment the file is received into SAP. The file is then uploaded automatically.

Author: Aveek Ghose

Company: IBM India

Created on: 18 August 2008

Author Bio

Aveek Ghose has 12 years of IT experience and has worked across the globe in SAP Implementations. Aveek has an MS in Information Systems from George Mason University in USA and a MS in Economics from Virginia Tech in USA and is currently working for IBM India.

Table of Contents

Flat File Interface	3
Interface Identification	3
Client Interface Identification Table	5
Logical File Name	5
Physical file Path - Name.....	6
Programming Standards for File Handling	7
/EUR/X_INTERFACE_START_PROC	7
Exporting Parameters	7
Table Parameters	7
Functionality of the Function Module	8
/EUR/X_INTERFACE_END_PROC	9
Header / Footer Record	10
Header Record	10
Footer Record	10
Error Handling.....	11
Screen Shots	12
SER / IDC interface to SAP	15
SAP Components.....	15
Disclaimer and Liability Notice.....	19

Flat File Interface

To implement a flat file interface, the following standards are applicable:

- each interface needs to have an interface identification
- data of the interface is stored in a customer table (interface catalogue)
- the directory location is stored in a logical file, the logical file name is stored in the customer table
- when running the interface, we record the start and end date and time, status in a history table
- there are standards for file names, and these are generated automatically by the interface start function
- the middleware system does the file transfer based upon the filename
- the security/authorization can be determined on the logical file path

Interface Identification

Each batch interface will be uniquely identified by an interface key. The key consists of 3 parameters and logically describes the *source*, *destination* and *content* of any interface file.

- Organizational Unit : This represents the logical business unit within Client that is responsible for the interface. This could be the CLIENT center, a region, a market or a subsidiary within a market.

Example: 0000 CLIENT Center – Globally managed interface

 CH11 Client Suisse SA

 GB11 Client UK (tbc)

For outbound interfaces the organizational unit indicates the origin of the data. For inbound interfaces the organizational unit identifies the destination of the data.

For each organizational unit a separate batch schedule will be set up with the holiday and factory calendar specific to that region/market/subsidiary. For this reason we will make use of company codes to describe the organizational unit. In case the organizational unit is not a legal Client entity (e.g. the AMS region as a whole) we will make use of dummy company codes and maintain these in a separate table.

- External system / partner : This represents the name of the external partner or legacy system that receives/sends the interface file. This can correspond to an actual legacy system or can be the name of an EDI partner.

For the naming conventions of this external systems, see the document of the basis team for external systems (see also appendix).

The external system / partner needs to be stored in customer table /EUR/XEEXTPAR: CLIENT External Partner Master (see also appendix).

- Interface Identifier : This describes the type of message being sent to/from the external partner. It is effectively a unique name for a CLIENT standard plug-in.

Example : CSORD01 EDI order interface

 CSINV01 EDI invoice interface

 FICAR01 CARAT Common interface – POPL message

The interface identifier describes *what kind of data* is in the file.

The interface identifier needs to be stored in customer table /EUR/XEINTMAS: CLIENT Interface ID Master.

Naming convention for this field is now **PPMMMMMNN**

Where:

- **PP** - Process area e.g. 'FI'
- **D** - Direction (I - Inbound; O - Outbound)
- **MMMMM** - Message content (ORD - Orders, PRDPL - Product P&L, etc). Essentially 5 bytes to describe the message.
- **NN** - Sequence number. This is used to uniquely identify interface programs with similar content. Example - Both R/3 core and CRM have orders inbound interfaces.

For each interface the 3 parameters above answer the following questions :

- Who is the data from? (organizational unit for outbound / External system for inbound)
- Who is the data for? (organizational unit for inbound / External system for outbound)
- What data is in the file? (interface identifier)

These three identifiers will be used to save the specific information about the interface, to save the run history and it is also used in the filename, and consequently in the middleware system as routing mechanism.

When a new interface is created, you first have to check that there is not an existing interface. Therefore you need to contact *interface coordinator*. This person will check the interface catalogue and check for similar interfaces and he will give the name of the external system and the interface identifier to be used for the new interface.

Client Interface Identification Table

All interfaces are stored in this CLIENT Interface Identification table: see table /EUR/XEINTID in section 2.

For maintenance, see transaction SM30 for this table.

The interface coordinator is owner of this table but each development coordinator is for having the right entries in this table for his own interfaces.

Logical File Name

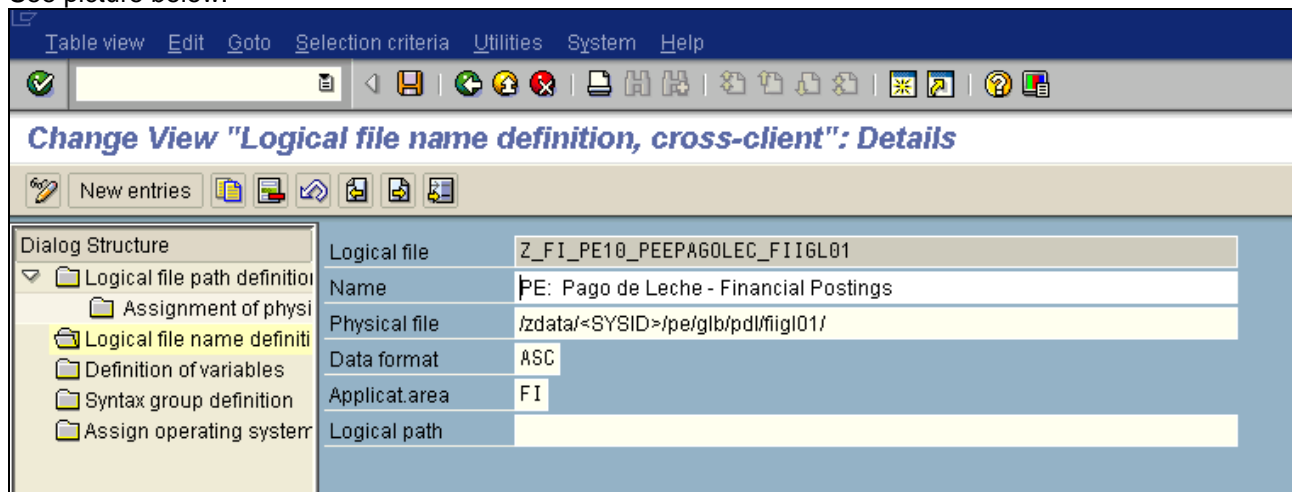
The standards of the logical file name are described in a document of the basis team (see also appendix). See above: shortcut to document: Standards-SIDLogicalSystemsClients.

Use transaction FILE to create the logical file.

The following set-up is used in CLIENT:

- no logical path definition is created
- all information is stored in the logical file name definition

See picture below.



Physical file Path - Name

The standards for the physical file path are stored in the following document: (See also appendix).

See above: shortcut to document : Standards-SIDLogicalSystemsClients.

The concept of the interface identification key is maintained also in the file name. This will reduce complexity within SAP and will give the adapter a key with which to determine routing information for the middleware.

The file name consists of :

[ORGANIZATIONAL UNIT] .
[EXTERNAL SYSTEM] .
[INTERFACE IDENTIFIER] .
[SEQUENCE NUMBER] . DAT

For example an interface with

- Organizational unit = Client Suisse (CH11)
- External system = CHEINFO+
- Interface identifier = FIOTRPRC01
- with file number 176

will have as file name : **CH11. CHEINFO+. FIOTRPRC01.000000000176.DAT**

The middleware system normally uses only the organizational unit, external system and interface identifier to identify the correct route for the file.

Programming Standards for File Handling

When the file starts, we have to call function /EUR/X_INTERFACE_START_PROC.

When ending the interface, we have to call function /EUR/X_INTERFACE_END_PROC.

We have a specific header and footer: see structure

/EUR/XINTERFACE_FOOTER and structure /EUR/XINTERFACE_HEADER.

/EUR/X_INTERFACE_START_PROC

The function has the following importing parameters:

BUKRS	: Organizational unit
INTID	: Interface identifier
PARNUM	: Partner number
INPUT_FILENAME	: Local file for upload/download
PARALLEL_RUNS_ALLOWED	: Parallel runs allowed = 'X', if the flag is set, the last interface run is not checked to not have status 'in process'.
NORMAL_RUN	: Run mode normal = 'X', if the flag is not set, the function assumes it is a rerun and it will get validated with the last sequence number and no new entry will be created in the history table
TEST_RUN	: Test run = 'X': if set, there is not updating of the history table
NO_SEQ_CHECK	: No sequence check = 'X', if set, there is no check on the sequence number for inbound files
NO_HEAD_FOOTER	: No header nor footer, if set, no checks are performed on the header or footer, and the header and footer are not removed

Exporting Parameters

XEINTID	: CLIENT Interface Identification table
PHYSICAL_FILENAME	: Logical file name
HEADER	: Interface header record

Table Parameters

INPUT_FILE	: EDI Interface file structure
------------	--------------------------------

Functionality of the Function Module

- Based upon the organizational unit, interface identifier and partner number, table /EUR/XEINTID is read. This gives the specific interface information.
- All interface information for this interface is selected by the history table /EUR/XEINTHIS and sorted by date and time
- If the flag PARALLEL_RUNS_ALLOWED is not set, then we check that the last specific interface run is not in status 'In process'.
- The header records is now filled using the information from /EUR/XEINTID, the date and time is set to the current date and time. The sequence number is the current sequence number of the last interface added by 1.
- We get the physical path using function FILE_GET_NAME using the logical file name stored in table /EUR/XEINTID. For inbound interfaces, the physical path is concatenated with the file name. For outbound interfaces, the same is done if the filename is not empty. If the filename INPUT_FILENAME is empty, the following filename is built:
physical path + XEINTID-BUKRS + '.' + XEINTID-PARNUM '.' + XEINTID-INTID + '.' + new sequence number + '.DAT'
- If it is inbound interface, some additional checks are done:
 - The file is opened and completely read into table parameter INPUT_FILE.
 - If the flag NO_HEAD_FOOTER is not set and NORMAL_RUN is set, the header is read and checked: record type, sequence number, partner number. The footer records is also read and checked: record type is checked and also the number of records
 - The header and the footer are removed from the table parameter INPUT_FILE.
- If the TEST_RUN parameter is empty, then the history table /EUR/XEINTHIS is updated with the current date and time if the status is not 'in process', otherwise the time stamp is not changed.

During Inbound interface processing the HEADER and FOOTER are provided in the file.

During Outbound interface processing

- The HEADER will be built by function module /EUR/X_INTERFACE_START_PROC, but will still need to be written into the file by the interface code

The FOOTER record needs to be built and written into the file by the interface code.

/EUR/X_INTERFACE_END_PROC

This function is to be called at the end of the interface processing.

The function has the following importing parameters:

HEADER : Interface header record
STATUS : Status
RECCT : Total records of the file
TRANS : Number of transactions
ERRCT : Number of incorrect transactions
TEST_RUN : Test run
MEMHOLD TYPE : Free Text (e.g. last record processed)

There are no export parameters nor table parameters.

The following statuses are valid:

'in process' : 1
'successfully finished' : 2
'failure': 3

The function updates the specific record using the HEADER record information in the history table /EUR/XEINTHIS. If the status is not in process, then the current date and time is set, otherwise the time stamp is not changed.

Header / Footer Record

Header Record

Regardless of the content of the file, the first record of any interface flat file should always be the HEADER record. It consists of the following fields :

Field	Description	Type	Size	Valueset / Example
RECTYPE	Record type	CHAR	10	'HEADER'
BUKRS	Organizational Unit	CHAR	4	e.g.: '0937' (Client Suisse)
PARNUM	External partner / system	CHAR	10	e.g.: 'CARAT'
INTID	Interface Identifier	CHAR	12	e.g.: 'CSINV01'
SEQNO	File sequence number (table XEINTHIS keeps log of previous sequence numbers for this interface)	NUMC	12	e.g.: '000000000005'
BEGDA	File creation date	DATS	8	YYYYMMDD
BEGTI	File creation time	TIMS	6	HHMMSS

This record is defined as structure **/EUR/XINTERFACE_HEADER** in the data dictionary.

Footer Record

Regardless of the content of the file, the last record of any interface flat file should always be the FOOTER record. It consists of the following fields :

Field	Description	Type	Size	Valueset / Example
RECTYPE	Record type	CHAR	10	'FOOTER'
RECORDS	Total number of records in file	NUMC	12	e.g.: '00000000100'
TRANS	Total number of LUWs in file	NUMC	12	e.g.: '00000000025'
CHKSUM	Control checksum value (is based on a control algorithm)	NUMC	25	e.g.: '00000000012586'

This record is defined as structure **/EUR/XINTERFACE_FOOTER** in the data dictionary.

Error Handling

The /EUR/X_INTERFACE_START_PROC function and the /EUR/X_INTERFACE_END_PROC function return specific exceptions back to the calling program whenever errors for these functions.

The calling program can decide how to handle this exceptions. However, if an entry is already made in the history table, the purpose is to set the status to 3:failure. If there were no errors, the status should be set to 2: success. This change is done when you call the /EUR/X_INTERFACE_END_PROC function with the proper import parameters.

If an interface has the status in process and a new interface run is requested with the option PARALLEL_RUNS_ALLOWED = space, then an the interface start function will return an exception. In this situation, the entry in the history table with the status in process has to be checked and if there is not interface running anymore and all reconciliation activities have taken place, the status should be changed in the table.

It is possible to code (within the interface program itself) sending the output report via SAPMail to designated users in the case of error. The fields /EUR/XEINTID-BUS_AGENT_TYPE, /EUR/XEINTID-BUS_AGENT, /EUR/XEINTID-IS_AGENT_TYPE, /EUR/XEINTID-IS_AGENT are used to determine where the output should be directed. The EDI Orders interface program /EUR/VGTRB EDI_ORDERS_INT is a good example of this functionality.

Screen Shots

Screen Shots of Interface Tables and Parameters of Function Modules:

Dictionary: Display Table

Transp. table: /NESGLB/XEINTID Active
Short Description: GLOBE Interface Identification table

Field	Key	Initi...	Data element	Data T...	Length	Deci...	Short Description	Group
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3		Client	
BUKRS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	/NESGLB/BUKRS	CHAR	4		Responsible Organization	
PARNUM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	/NESGLB/EXTPART	CHAR	10		External Partner	
INTID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	/NESGLB/INTID	CHAR	12		Interface ID	
DIRECT	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/EDI_DIR	CHAR	1		Interface Direction	
REPID	<input type="checkbox"/>	<input type="checkbox"/>	REPID	CHAR	40		ABAP Program Name	
INTDESC	<input type="checkbox"/>	<input type="checkbox"/>	EDI_DESCRP	CHAR	50		Description of process	
BUS_AGENT_TYPE	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/BUSAGEN	CHAR	10		Business Responsible Agent Type	
BUS_AGENT	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/BUSAGENT	CHAR	25		Business Agent	
IS_AGENT_TYPE	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/ISAGENT	CHAR	10		IS/IT Responsible Agent Type	
IS_AGENT	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/ISAGENT	CHAR	25		IS/IT Agent	
ERR_THRESHOLD	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/THRESHD	NUMC	5		Error threshold	
INTITYPE	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/INTITY	CHAR	1		Interface Type	
IDOCTP	<input type="checkbox"/>	<input type="checkbox"/>	EDI_IDOCTP	CHAR	30		Basic type	
MESTYP	<input type="checkbox"/>	<input type="checkbox"/>	EDI_MESTYP	CHAR	30		Message type	
CIMTYP	<input type="checkbox"/>	<input type="checkbox"/>	EDI_CIMTYP	CHAR	30		Extension	
PVRCPV	<input type="checkbox"/>	<input type="checkbox"/>	EDI_PVRCPV	CHAR	10		Port	
PARTYP	<input type="checkbox"/>	<input type="checkbox"/>	EDIPPARTYP	CHAR	2		Partner Type	

Dictionary: Display Table

Transp. table: /NESGLB/XEINTID Active
Short Description: GLOBE Interface Identification table

Field	Key	Initi...	Data element	Data T...	Length	Deci...	Short Description	Group
BITCJOB	<input type="checkbox"/>	<input type="checkbox"/>	BITCJOB	CHAR	32		Background job name	
FILENAME	<input type="checkbox"/>	<input type="checkbox"/>	FILEINTERN	CHAR	60		Logical file name	
DESTINATION	<input type="checkbox"/>	<input type="checkbox"/>	AD_LOGDEST	CHAR	32		RFC logical destination	
PRTEAM	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/PRTEAM	CHAR	4		Process Team	
INTEXEC	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/INTEX	CHAR	1		Interface Execution Method	
WORKREQ	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/YWREQUE	CHAR	9		Work Request number	
FREQUENCY	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/INTFR	CHAR	1		Interface Frequency	
WAITTIME	<input type="checkbox"/>	<input type="checkbox"/>	TIMER_I	NUMC	4		Updating Interval (in Seconds)	
HISTRETN	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/HISTRETN	NUMC	3		History Retention period (in runs)	
FILERETN	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/FILERETN	NUMC	3		File Retention period (in runs)	
ERRORFILE	<input type="checkbox"/>	<input type="checkbox"/>	/NESGLB/ERRORFI	CHAR	60		Error file	

Function Builder: Display /NESGLB/X_INTERFACE_START_PROC

Function module: /NESGLB/X_INTERFACE_START_PROC Active

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Type	Associated Type	Default value	Opt.	Pa...	Short text	Lo...
BUKRS	TYPE	T001 - BUKRS		<input type="checkbox"/>	<input checked="" type="checkbox"/>		
INTID	TYPE	/NESGLB/INTID		<input type="checkbox"/>	<input checked="" type="checkbox"/>		
PARNUM	TYPE	/NESGLB/EXTPAR...		<input type="checkbox"/>	<input checked="" type="checkbox"/>		
INPUT_FILENAME	TYPE	DRAW-FILEP		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Local file for upload/download	
PARALLEL_RUNS_ALLOW...	TYPE	/NESGLB/PARALL...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Parallel runs allowed = %'	
NORMAL_RUN	TYPE	BDC_DAMOD		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Run mode normal = %'	
TEST_RUN	TYPE	BDC_DAMOD		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Test run = %'	
NO_SEQ_CHECK	TYPE	BDC_DAMOD		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No sequence check = %'	
NO_HEAD_FOOTER	TYPE	BDC_DAMOD		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No header nor footer	

GD2 (4) (103) ddadb212 OVR

Function Builder: Display /NESGLB/X_INTERFACE_START_PROC

Function module: /NESGLB/X_INTERFACE_START_PROC Active

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Type spec.	Associated Type	Pass Va...	Short text	Long text
XEINTID	TYPE	/NESGLB/XEINTID	<input type="checkbox"/>	GLOBE Interface Identification table	
PHYSICAL_FILENAME	TYPE	DRAW-FILEP	<input type="checkbox"/>	Logical file name	
HEADER	TYPE	/NESGLB/XINTERFACE...	<input type="checkbox"/>	Interface header record	

GD2 (4) (103) ddadb212 OVR

Function Builder: Display /NESGLB/X_INTERFACE_START_PROC

Function module: /NESGLB/X_INTERFACE_START_PROC Active

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Type spec.	Associated Type	Optional	Short text	Long text
INPUT_FILE	LIKE	/NESGLB/XEINTFIL	<input checked="" type="checkbox"/>	EDI Interface file structure	

GD2 (4) (103) ddadb212 OVR

SAP

Structure Edit Goto Utilities Extras Environment System Help

Dictionary: Display Structure

Structure: /NESGLB/XEINTFIL Active
 Short Description: GLOBE Flat File Interface - File Container

Attributes Components Entry help/check Currency/quantity fields

Built-in type 1 / 2

Component	RT...	Component type	Data Type	Length	Deci...	Short Description	Group
RECTYPE	<input type="checkbox"/>	/NESGLB/RECTYP	CHAR	10	0	Record Type	
RECDETL	<input type="checkbox"/>	/NESGLB/RECDETL	CHAR	3000	0	Record Content	

GD2 (4) (103) ddadb212 OVR

SAP

Function module Edit Goto Utilities Environment System Help

Function Builder: Display /NESGLB/X_INTERFACE_START_PROC

Function module: /NESGLB/X_INTERFACE_START_PROC Active

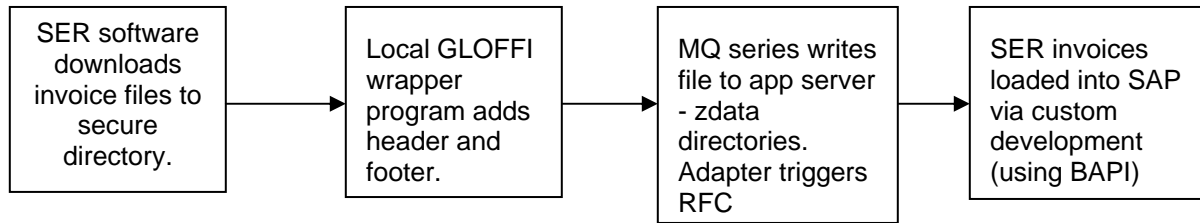
Attributes Import Export Changing Tables Exceptions Source code

Exceptn Classes

Exception	Short text	Long txt
INVALID_INTERFACE	Interface details not stored in /NESGLB/XEINTID	
LAST_RUN_EXECUTING	Previous interface execution incomplete	
FILE_NOT_FOUND	Logical filename not defined	
NO_HEADER_RECORD	Input file does not contain HEADER record as first ...	
NO_FOOTER_RECORD	Input file does not contain FOOTER record as last ...	
INCORRECT_NO_RECORDS	Incorrect number of records in FOOTER record	
INCORRECT_SEQUENCE	HEADER record contains incorrect file sequence n...	
CANNOT_OPEN_FILE	Unable to open input file	
INCORRECT_PARTNER	Partner number in header record incorrect	
HISTORY_UPDATE_FAILED	Update of history table failed	

GD2 (4) (103) ddadb212 OVR

SER / IDC interface to SAP



- SER software scans in the invoices received and formats into the file, then places this file on the secure directory depending on the PO number range (different range for each box).
- GLOFFI polls directories for any new files.
- GLOFFI wrapper program adds header and footer
- GLOFFI places file on secure directory for MW to pick up automatically
- MW picks up file and places on designated secure directory on SAP application server (zdata directories).
- The SAP custom developed wrapper program /EUR/FGBFCI_SER_INV_POST_PARK is triggered by an RFC call made by the adapter script the moment the file is received into SAP. The file is then uploaded automatically.

Local support team – needs to ensure that the relevant people have access to the secured directories

GLOFFI access - The polling job needs to have a user id set up

- For Test – which will be used for test and integration testing (MIT)
- For Production – which will be used for pre-production (MAT) and Production
- With a non-expiring password – this will only be used by the application to access the network directory.
- The User ids needs access to Read, Write access to Local server where the software runs and write access to the GLOFFI server.

SAP Components

Example below shows how the NPPE SER interface works on the SAP side.

File will arrive in SAP:

Directory: /zdata/EB7/gb/glb/ser/fiiser01/

Name: GB23.GBESER.FIISER01.041004135402.DAT

Directory: /zdata/EB1/gb/glb/ser/fiiser02/

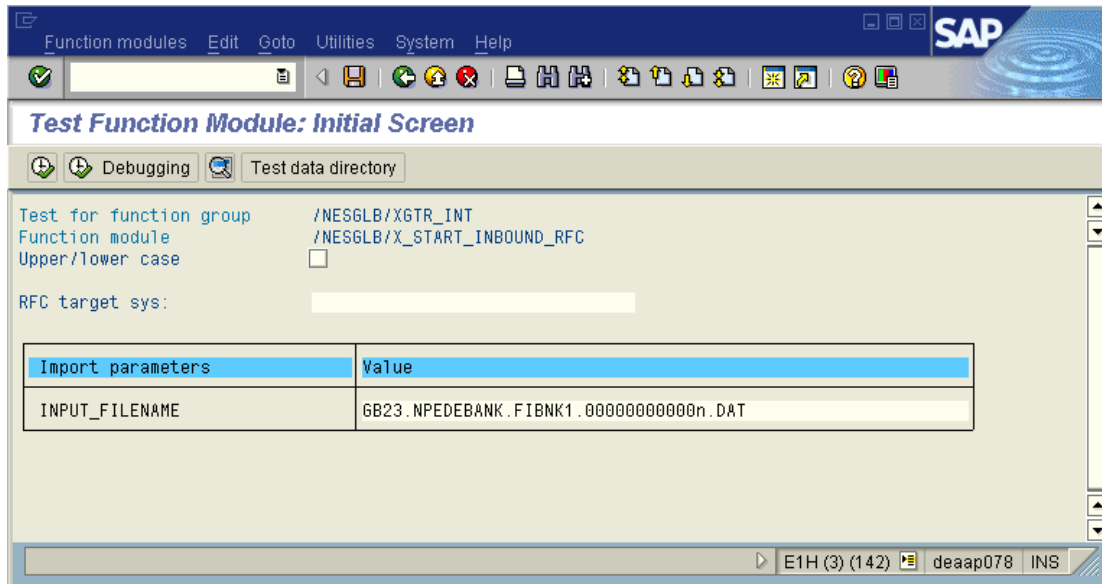
Name: GB23.GBESER.FIISER02.061004124402.DAT

Directory: /zdata/EBH/gb/glb/ser/fiiser03/

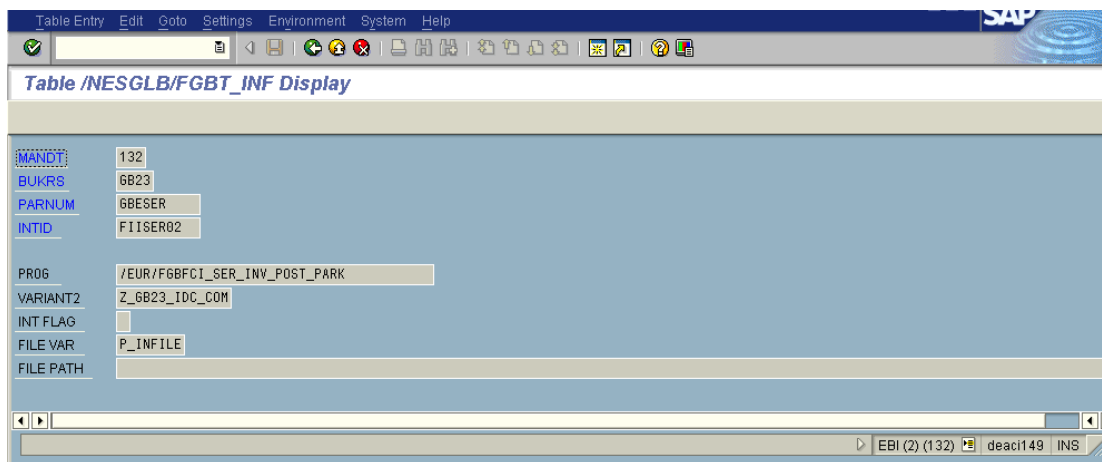
Name: GB23.GBESER.FIISER03.011004132402.DAT

The adapter on the SAP GAF will trigger an RFC on SAP, passing the filename.

RFC is:



The RFC will select all details from the RFC table, using the 3 interface control elements from the filename (Resp Org, Ext Partner and Interface ID):



The report name and variant name are selected from the table above and these are run, passing the filename into the SER variant. The name of the variant field for the filename can vary depending on how the program was coded. For bank statement, in program /EUR/FGBFCI_SER_INV_POST_PARK, the filename field in the selection screen was named 'P_INFILE'. As this can vary by interface, this field name is included in the table above and is used dynamically by the RFC.

The SER interface is run with the variant and the filename as passed over from the adapter program (with correct sequence number).

The screenshot shows the SAP 'SER Intelligent Data Capture - Invoice Posting and Parking' window. It contains two main sections: 'Interface Identification' and 'SER Data - Indirect material types for posting'.

Interface Identification	
Company Code	GB23
External System	GBESER
Interface Identifier	FI1SER02
UNIX Input Filename from SER	GB23.GBESER.FI1SER02.121004112700.DAT
Receiver	SHAREDSERVICECENTRE@PURINA.NESTLE.COM
Printer	UKEMAIL

SER Data - Indirect material types for posting	
Material Types	ERSA

The status bar at the bottom indicates 'EBI (1) (132) deaci149 INS'.

Details of the interface are selected from the interface control table /EUR/XEINTID (details vary by box/interface id):

The screenshot shows the SAP 'Table /NESGLB/XEINTID Display' window. It displays a list of fields and their corresponding values for the interface.

MANDT	132
BUKRS	GB23
PARNUM	GBESER
INTID	FI1SER02
DIRECT	2
REPID	/EUR/F6BFC1_SER_INV_POST_PARK
INTDESC	UK SER - IDC interface to COM box
BUS AGENT TYPE	
BUS AGENT	
IS AGENT TYPE	
IS AGENT	
ERR THRESHOLD	0
INTTYPE	F
IDOCTP	
MESTYP	
CIMTYP	
PVRCVP	
PARTYP	
BTCJOB	GB_GB23_FC_R_003995_SER02
FILENAME	Z_SER_GB23_COM
DESTINATION	
PRTEAM	FC
INTEXC	B
WORKREQ	3995
FREQUENCY	H
WAITTIME	1
UNTRFTH	10

The status bar at the bottom indicates 'EBI (2) (132) deaci149 INS'. The Windows taskbar at the bottom shows the Start button and several open applications including 'Inbox', 'My Music', 'SAP Log...', 'How to I...', 'Window...', 'ABAP Ed...', and 'Table /...'.

The wrapper program will remove the header and footer record and validate the contents before processing the individual invoices contained in the file.

When the job is complete, there are two places where the results can be seen:

- The first place is via SM36, where a job log can be found for job created by the RFC. The job name can be found in the interface id job above – for the SER interface the name is GB_GB23_FC_R_003995_SER01, 2 or 3.
- The job logs will give details of any errors encountered and are important if the RFC fails and the queue is blocked.
- The second place is via the interface controls history table /EUR/XEINTHIS. Use the interface controls to look up the last running of the interface (use the last run date and time). This will give details of the last sequence number, the success/failure code, number of records processed and the date/time the interface was run.

If there is an error with the sequence number, the interface controls table should be checked to identify which file was last processed and which sequence number is expected. This could either mean a missing file, a duplicate file or an error with the incrementing of the sequence number.

- Missing file: find the missing file and reason for failure and if relevant, resend and then resend the next one in correct order. If no file to resend (missing file is an error), reset the sequence number to the correct one on GLOFFI and resend correct file.
- Duplicate files: investigate if same file sent twice – if so, ignore second file. If error with sequence number, fix on GLOFFI and resend files with correct sequence numbers.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.