Extractor Technical Design Specification

Qlik Connector for SAP Version: 6.3.2 - November 2016

Contents
General.................................................................................................................. 3

Configuration ........................................................................................................ 3
  Transaction /N/QTQVC/EXTRACTOR_ADM ......................................................... 3
  Option Create....................................................................................................... 3
  Option Verify...................................................................................................... 4
  Option Delete..................................................................................................... 5
  Option Modify..................................................................................................... 5
  Regarding option Create and Modify............................................................... 5

Useful transactions ............................................................................................... 7
  Transaction /N/QTQVC/DEACTIVATE ................................................................. 7
  Transaction /N/QTQVC/DELETE_INIT ................................................................. 8
  Transaction /N/QTQVC/DELETE ........................................................................ 9

Activate / generate Extractor ............................................................................. 10
  Transaction /N/QTQVC/ACTIVATE .................................................................. 10

Extractor database tables .................................................................................... 18
  /QTQVC/STATUS .............................................................................................. 18
  /QTQVC/CONVERT ........................................................................................ 19

Lookup activated Extractors, Hierarchies, Logical systems and Language codes ......... 19
  Function module /QTQVC/GET_EXTRACTOR_OBJECTS ..................................... 19

Get data of activated Extractor (datasource) ....................................................... 20
  Function module /QTQVC/GET_ACTIVATED_EXTRACTOR ............................... 20

Create Request IDoc ............................................................................................ 21
  Function module /QTQVC/CREATE_REQUEST_IDOC ....................................... 21

Update Delta Init database tables the first time .................................................. 24
  Function module /QTQVC/DELTAINIT_START ............................................... 24

Update Delta Init database tables the second time .............................................. 24
  Function module RSC1_DELTAINIT_FINISH .................................................. 25

Qlik Connector for SAP
IDOC content........................................................................................................................................... 25
IDOC Structure ............................................................................................................................................ 25
Unpack data from an IDOC segment ........................................................................................................ 28
Check if Delta Init has been performed previously ........................................................................... 29
  Function module /QTQVC/EXTRACTION_STATUS ........................................................................... 29
Get Message text ....................................................................................................................................... 29
  Function module /QTQVC/EXTRACTION_STATUS ........................................................................... 29
Check Extraction status ............................................................................................................................ 30
  Function module /QTQVC/EXTRACTION_STATUS ........................................................................... 30
Update status table when a job is finished or cancelled ...................................................................... 32
  Function module /QTQVC/UPDATE_STATUS ................................................................................... 32
Resend an Extraction job .......................................................................................................................... 32
  Function module /QTQVC/RESEND_EXTRACTION_JOB .................................................................... 33
Get Hierarchy details .................................................................................................................................. 34
  Function module RSAP_REMOTE_HIERARCHY_CATALOG ................................................................. 34
Timeouts ...................................................................................................................................................... 35
Migrate activated Extractors ...................................................................................................................... 36
Migrate Extractor Environment .................................................................................................................. 40
General
Please note that this document is not a manual although some sections can be found in the manual as well. The intention with this document is to get a deeper knowledge about how the Extractor connector is working.

Configuration

Transaction /N/QTQVC/EXTRACTOR_ADM

Maintain the Extractor environment.

![Maintain Extractor environment](image)

Option Create
Creates the Extractor environment.
It is necessary to unlock the client during the execution of the ‘create’ option. Use transaction SCC4. Below is a summary of how the updates are made when creating the Extractor environment. Only SAP standard function modules are used. Remember to lock the client again after creation.

Create new RFC connection of type ABAP (call SAP function module ‘RFC_MODIFY_R3_DESTINATION’) RFC-Connection gets the same name as the Logical system. See transaction SM59. RFC-Connection points to a Program-ID (same name as Logical system) which is used from the Windows side to establish a connection with SAP.

Configure the ALE parameters for the new connection (Call SAP function module ‘RSAP_BIW_CONNECT)
Creates Logical system of receiver. See transaction SALE.
Creates Partner profile of type LS (same name as Logical system). See transaction WE20. Contains the used EDI Logical message types which are RSRQST, RSINFO and RSSEND (see transaction WE81).
Creates Basic IDoc type for data transfer. The name is hardcoded like ZSQAQTVCEXTR1. The name is a concatenation of three parts. The first part is ZS. The second part is a prefix. We will use the first available in a series like QA, QB, QC, QD and so on. To be available, the prefix (TSPREFIX) must not exist in the database table RSBASIDOC, where it will be stored when the environment is created. There is a one to one relation between Logical system and prefix. The third part is the name of the logical system. See transaction WE30.

Delete our RFC-connection of type ABAP (call SAP function module ‘RFC_MODIFY_R3_DESTINATION’) The connection will be re-created in the next step.

Create new RFC connection of type TCP/IP instead (call SAP function module ‘RFC_MODIFY_TCP/IP_DESTINATION’)
Re-create the connection.

Modify the new logical system to set the 'send immediately' flag (call SAP function module ‘EDI_AGREE_IN_MESSTYPE_UPDATE’) We want the 'send immediately' flag to be default.

Copy old datasource entries from ROIGEN and ROCHABGEN to ROOSGEN for 30B compliance (call SAP function module ‘RSA2_ROOSGEN_COPY’) Copy old entries if available.

Fix source system communication release information (call SAP function module ‘RSAS_RWBCRL_STORE’) Update release information.

A record in the database table /QTQVC/STATUS (part of the SAP connector package) is created:

Data Browser: Table /QTQVC/STATUS:

Table: /QTQVC/STATUS
Displayed Fields: 11 of 11 Fixed Columns:

<table>
<thead>
<tr>
<th>MANDT</th>
<th>CONNECTOR</th>
<th>JOBDATE</th>
<th>JOBSTIME</th>
<th>JOBSTATUS</th>
<th>JOI</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>QTVCEXTR1</td>
<td>19.04.2011</td>
<td>09:21:57</td>
<td>K</td>
<td></td>
</tr>
</tbody>
</table>

The jobstatus ‘K’ stands for ‘Keep’. This record must not be deleted. It will be deleted in case the whole Extractor environment for the Logical system QTVCEXTR1 is deleted.

Option Verify
Verifies that all necessary components of the Extractor environment are configured.

Qlik Connector for SAP
Option Delete
Deletion of all components of the Extractor environment.
It is necessary to unlock the client during the execution of the ‘delete’ option. Use transaction SCC4.
Remember to lock the client again after deletion.
In the ‘delete’ option there is a check that there is a record in the database table /QTQVC/STATUS for the Logical system entered. The jobstatus should be ‘K’. If no record is found then the Deletion is not executed.

Option Modify
Modifies the RFC Destination.

Regarding option Create and Modify
A server has to be selected that the Extractor connector will use for the RFC communication. The user can look up possible servers in the SAP system cluster (from transaction SM51) and select one of them from the field Server Name:

If the Central instance is part of the list, it can also be selected. The selected server should have dialog processes.

In this example the Gateway Host (SAP1BW73LD.RDLund.qliktech.com) and Gateway Service (sapgw00) of the selected server are used when the RFC Destination is created or modified. Check the result in transaction SM59 under TCP/IP connections:
The Windows part of the connector will always use the Gateway Host and Gateway Service of the selected server when registering the Program ID of the RFC Destination (starting up the ‘listener’).
It is possible to change to another server anytime. The RFC Destination can easily be updated with the Gateway Host and Gateway Service of the new server and the connector will automatically use the new settings when starting up the ‘listener’.

The RFC Destination can be updated in the transaction /n/QTQVC/EXTRACTOR_ADM by using the option **Modify** or by using the SAP transaction SM59.

If the server is changed in SAP, the Service ‘Qlik SAP Network Server’ has to be restarted.

The SAP parameter SAPLOCALHOSTFULL has to contain the correct value for each server. Log on directly to each SAP server and check the value in transaction RZ11. SAPLOCALHOSTFULL should contain the Fully Qualified Domain Name of the server.

**Useful transactions**

**Transaction /N/QTQVC/DEACTIVATE**

Deactivate previously activated Extractors. Enter Logical system name, Language code and Transfer Method and press the ‘Get Extractors’ button. Select Extractor(s) to deactivate and press button ‘Deactivate Extractors’. 
Transaction /N/QTQVC/DELETE_INIT

Delete Delta Init for an Extractor. Necessary if a new Init should be performed for the Extractor.
Delete Delta Init

Delete Delta Init for:

<table>
<thead>
<tr>
<th>Logical system of receiver</th>
<th>Extractor name</th>
</tr>
</thead>
<tbody>
<tr>
<td>QTQVC/CEXT1</td>
<td>OCUSTOMER_ATTR</td>
</tr>
</tbody>
</table>

Transaction /N/QTQVC/DELETE

Use it to delete a single record from the database table /QTQVC/STATUS. Could be necessary if there has been a communication error between SAP and the Windows part of the connector. Use transaction SE16 to display the key values of the table record. Then copy and paste them into the delete transaction.

Note, do not use the four delete options for complete table deletion on the top of the screen!!

Delete Database Table Records

Delete records from table:

/QTQVC/CONTROL
/QTQVC/JOB_SQL
/QTQVC/TRACE
/QTQVC/STATUS

Delete records from year: 2011

Delete single record from table:

/QTQVC/STATUS
  CONNECTOR: QTQVC/CEXT1
  JOBDATE: 29.06.2011
  JOBTIME: 12:55:11

Simulate Deletion
Delete Records

Qlik Connector for SAP
Activate / generate Extractor

Transaction /N/QTQVC/ACTIVATE

Enter the ‘Logical system of receiver’, ‘Language code’ and ‘Extractor name’ and press the button ‘Get Extractor’.

Select Transfer Method (tRFC or IDoc). If the Extractor has previously been activated for the IDoc transfer method and should be activated for tRFC instead, it has to be deactivated first in transaction
Then select the fields that should be part of the extraction by putting ‘X’ in the column ‘SELECT’.
If the Extractor is previously activated, the fields that were selected then are automatically marked with ‘X’. It is possible to activate the Extractor again with new fields selected.
Finally press the button ‘Activate Extractor’.

When the Extractor is activated for the IDoc transfer method, the transfer structure is created under the Basic IDoc type with a name like ZSQAQTQVCEXTR1 (depending on the used logical system, see transaction WE30):

**Display basic type: ZSQAQTQVCEXTR1**

<table>
<thead>
<tr>
<th>ZSQAQTQVCEXTR1 Basis IDoc type</th>
<th>QTVCEXTR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1RSHIE Data transfer Idoc: Hierarchy header information</td>
<td></td>
</tr>
<tr>
<td>E1RSTTX Data transfer Idoc: Texts</td>
<td></td>
</tr>
<tr>
<td>/BIC/QAOCOMP_CODE_TEXT001 Transfer struct. for Infosource OCOMP_CODE_TEXT</td>
<td></td>
</tr>
<tr>
<td>/BIC/QAOSL_ACCOUNT_ATTR001 Transfer struct. for Infosource OSL_ACCOUNT_ATTR</td>
<td></td>
</tr>
<tr>
<td>/BIC/QAOSL_ACCOUNT_TEXT001 Transfer struct. for Infosource OSL_ACCOUNT_TEXT</td>
<td></td>
</tr>
<tr>
<td>/BIC/QAOCO_PC_ACT_02001 Transfer struct. for Infosource OCG_PC_ACT_02</td>
<td></td>
</tr>
<tr>
<td>/BIC/QAGFI_GL_4001 Transfer struct. for Infosource OFI_GL_4</td>
<td></td>
</tr>
<tr>
<td>/BIC/QAGEC_PCA_3001 Transfer struct. for Infosource OEC_PCA_3</td>
<td></td>
</tr>
<tr>
<td>/BIC/QAGMATERIAL_TEXT001 Transfer struct. for Infosource OMATERIAL_TEXT</td>
<td></td>
</tr>
<tr>
<td>/BIC/QAGCUSTOMER_TEXT001 Transfer struct. for Infosource OCUSTOMER_TEXT</td>
<td></td>
</tr>
<tr>
<td>/BIC/QAGCOMP_CODE_ATTR001 Transfer struct. for Infosource OCOMP_CODE_ATTR</td>
<td></td>
</tr>
<tr>
<td>/BIC/QAGFI_AR_3001 Transfer struct. for Infosource OFI_AR_3</td>
<td></td>
</tr>
<tr>
<td>/BIC/QAGCOSTCENTER_TEXT001 Transfer struct. for Infosource OCOSTCENTER_TEXT</td>
<td></td>
</tr>
</tbody>
</table>

To check the fields of a transfer structure, double-click the segment type name and press the button ‘Segment editor’.
### Attribute Display

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segm.type</td>
<td>/BIC/QA6FI_GL_4901</td>
</tr>
<tr>
<td>Mandatory seg</td>
<td></td>
</tr>
<tr>
<td>Minimum number</td>
<td>1</td>
</tr>
<tr>
<td>Maximum number</td>
<td>9999999999</td>
</tr>
<tr>
<td>Parent segment</td>
<td>E1RSSH</td>
</tr>
<tr>
<td>Hier.level</td>
<td>2</td>
</tr>
</tbody>
</table>

View the fields:
When the Extractor is activated for the tRFC transfer method, the transfer structure can be found in transaction SE11:
The name of the transfer structure is a concatenation of `/BIC/`, `Transfer structure prefix`, Extractor name (first 17 characters) and a serial number.
The Transfer structure prefix can be found in table RSBASIDOC (transaction SE16).
The Extractor name (first 17 characters) and serial number can be found in table /QTQVC/CONVERT.
Transfer structure layout in SE11:
An Extraction program for the used Extractor is also generated. To get the name of that program, look in table ROOSGEN (transaction SE16):
Take the content of the field GENUID and prefix it with ‘GP’. Go to transaction SE38 and display the program:

**ABAP Editor: Initial Screen**

Program: `GPD9ZK1BK2Z2ZISIE6S4B2HJ7`
Create

Subobjects:
- © Source Code
- ○ Variants
- ○ Attributes
- ○ Documentation
- ○ Text elements

Display  Change
Do not change this program!

## Extractor database tables

### /QTQVC/STATUS

<table>
<thead>
<tr>
<th>Field</th>
<th>Key/Init</th>
<th>Data element</th>
<th>Data Ty</th>
<th>Length</th>
<th>Decim</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MANDT</td>
<td></td>
<td>MANDT</td>
<td>CLNT</td>
<td>3</td>
<td></td>
<td>Client</td>
</tr>
<tr>
<td>CONNECTOR</td>
<td></td>
<td>/QTQVC/CONNECTOR</td>
<td>CHAR</td>
<td>10</td>
<td></td>
<td>Connector (like Logical system name, REPORT, QUERY)</td>
</tr>
<tr>
<td>JOBDATE</td>
<td></td>
<td>/QTQVC/JOBDATE</td>
<td>DATS</td>
<td>8</td>
<td></td>
<td>Date of QVC Job</td>
</tr>
<tr>
<td>JOBTIME</td>
<td></td>
<td>/QTQVC/JOBTIME</td>
<td>TIMS</td>
<td>6</td>
<td></td>
<td>Time of QVC Job</td>
</tr>
<tr>
<td>JOBSTATUS</td>
<td></td>
<td>/QTQVC/JOBTSTATUS</td>
<td>CHAR</td>
<td>1</td>
<td></td>
<td>Status of QVC Job</td>
</tr>
<tr>
<td>ENDATE</td>
<td></td>
<td>/QTQVC/ENDATE</td>
<td>DATS</td>
<td>8</td>
<td></td>
<td>Date of QVC Job</td>
</tr>
<tr>
<td>ENTIME</td>
<td></td>
<td>/QTQVC/ENTIME</td>
<td>TIMS</td>
<td>6</td>
<td></td>
<td>Time of QVC Job</td>
</tr>
<tr>
<td>JONAME</td>
<td></td>
<td>JONAME</td>
<td>CHAR</td>
<td>32</td>
<td></td>
<td>Background job name</td>
</tr>
<tr>
<td>JOBCOUNT</td>
<td></td>
<td>JOBCOUNT</td>
<td>CHAR</td>
<td>8</td>
<td></td>
<td>Job ID</td>
</tr>
<tr>
<td>OBJECT</td>
<td></td>
<td>/QTQVC/OBJECT</td>
<td>CHAR</td>
<td>40</td>
<td></td>
<td>Object like Extractor name, Report, Query etc.</td>
</tr>
<tr>
<td>UPCODE</td>
<td></td>
<td>/QTQVC/UPCODE</td>
<td>CHAR</td>
<td>2</td>
<td></td>
<td>Mode of data update (Full, Delta, etc.)</td>
</tr>
<tr>
<td>USERNAME</td>
<td></td>
<td>USERNAME</td>
<td>CHAR</td>
<td>12</td>
<td></td>
<td>User Name</td>
</tr>
<tr>
<td>WNUSER</td>
<td></td>
<td>/QTQVC/WNUSER</td>
<td>CHAR</td>
<td>50</td>
<td></td>
<td>Windows user</td>
</tr>
<tr>
<td>WNIPMR</td>
<td></td>
<td>/QTQVC/WNIPMR</td>
<td>CHAR</td>
<td>10</td>
<td></td>
<td>Windows IP number</td>
</tr>
<tr>
<td>TEXT</td>
<td></td>
<td>TEXT</td>
<td>CHAR</td>
<td>255</td>
<td></td>
<td>Text, 255 Characters</td>
</tr>
</tbody>
</table>

This table is used to control the Extraction jobs.
This table is used to create a unique serial number which is part of the Basic IDoc type segment name or transfer structure for the tRFC method. It is necessary to distinguish between long Extractor names (that have the same leading 17 characters) for the Basic IDoc type segments or tRFC transfer structure.

**Lookup activated Extractors, Hierarchies, Logical systems and Language codes**

**Function module /QTQVC/GET_EXTRACTOR_OBJECTS**

Use parameters:
- RLOGSYS (char 10)
- LANGU (char 1)

Set parameters:
- RLOGSYS = name of Logical system or SPACE
- LANGU = selected Language code (supplied when RLOGSYS has a value)

If the parameter RLOGSYS contains SPACE, then the function module performs a lookup of available Logical systems. They are returned in a table called CONNECTORTABLE:

- **CONNECTOR** CHAR 10 Connector (like Logical system name, REPORT, QUERY)

Another table called LANGUAGETABLE is also returned. It contains available language codes:

- **SPRAS** LANG 1 Language Key
- **SPTXT** CHAR 16 Language description

Qlik Connector for SAP
If the parameters RLOGSYS and LANGU contains values, then the function module returns two tables. One table contains activated Extractor objects and one table contains available Hierarchy objects.

EXTRACTORTABLE:

| OLTPSOURCE  | CHAR 30 | Extractor (DataSource) |
| TXTMD       | CHAR 40 | Extractor description   |
| DELTA       | CHAR 4  | Delta Process for a DataSource |
| DELTA_DESCR | CHAR 60 | Delta description       |

HIERARCHYTABLE:

| OLTPSOURCE  | CHAR 30 | Hierarchy (DataSource) |
| TXTMD       | CHAR 40 | Hierarchy description   |

### Get data of activated Extractor (datasource)

**Function module /QTQVC/GET_ACTIVATED_EXTRACTOR**

Use parameters:

- OBJECT (char 30)
- RLOGSYS (char 10)
- TRANSFER_METHOD (CHAR 1)
- LANGU (CHAR 1)

Set parameters:

- OBJECT = Extractor name
- RLOGSYS = name of Logical system
- TRANSFER_METHOD = 'T' for tRFC or 'I' for IDoc.
- LANGU = Input value from Windows

If the Extractor is not activated, the function module returns an error message.

The segment type name is returned in the parameter SEGMENT (CHAR 30).

The function module returns a table FIELDS with the selected fields of the activated Extractor:

| SEGMENTTYP  | CHAR 30 |
| FIELDDATE   | CHAR 30 |
| INTLEN      | NUMC 6  |
| EXTLEN      | NUMC 6  |
| FIELD_POS   | NUMC 6  |

Qlik Connector for SAP
If the field SELECTION contains a value > SPACE, the field in FIELDNAME can have conditions. Possible values are (for the moment we are only allowing SPACE, M, P and X):

A  Field in OLTP and BW Hidden by SAP
M  Selection Required, Visible
SPACE No Selection Possible, Visibility Set
P  Selection Adjustable, Visibility Set
X  Selection Adjustable, Visibility Set
1  Pure Selection Field, Selection Set
2  Pure Selection Field, Selection Set
3  Selection Adjustable, Visibility Adjustable
4  No Selection Possible, Visibility Adjustable

Use the fields from the table FIELDS above to create script and then call the function module below.

The name of the Transfer structure (Basic IDoc type segment) is returned in the parameter SEGNAME (char 30)

If not successful the function module will return an error message.

Create Request IDoc

This step requires that the connection to SAP is established before calling the function module.

Function module /QTQVC/CREATE_REQUEST_IDOC

Use parameters
OBJECT (char 30)
UPDMODE (char 2)
RLOGSYS (char 10)
WINUSER (char 50)
WINIPNR (char 16)
TIMEOUT_SAP (INT 10)
TRANSFER_METHOD (CHAR 1)
LANGU (CHAR 1)

If the OBJECT is a hierarchy, also add the parameters:

HIENM (CHAR 30)
VERSION (CHAR 3)

If Conditions should be used for a field then use table SELECT_CRITERIA with layout:
FIELDNM CHAR 30
SIGN CHAR 1
OPTION CHAR 2
LOW CHAR 45
HIGH CHAR 45

Set parameters:
OBJECT = Extractor name
RLOGSYS = name of Logical system.
UPDMODE = see explanation below
WINUSER = Windows user
WINIPNR = Windows IP number
TIMEOUT_SAP = default value or value from connection string (in seconds)
TRANSFER_METHOD = ‘T’ for tRFC or ‘I’ for IDoc.
LANGU = input value from Windows

For Hierarchies:
HIENM = value from function module RSAP_REMOTE_HIERARCHY_CATALOG
VERSION = value from function module RSAP_REMOTE_HIERARCHY_CATALOG

Possible values for parameter UPDMODE:
C : Initialization of the delta transfer
F : Transfer of all requested data
D : Transfer of the deltas since the last request
H : Hierarchies

If Conditions should be used, set table SELECT_CRITERIA (one row for each condition of a field):
FIELDNM = fieldname
SIGN = I (include) or E (exclude)
OPTION = see some examples below
LOW = single value or low value in a range
HIGH = high value in a range

Qlik Connector for SAP
Possible values for parameter OPTION:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>=, EQ</td>
<td>Equal: True, if the content of operand1 matches the content of operand2.</td>
<td>FIELDNM = BUKRS SIGN =I OPTION = BT LOW = 1000 HIGH = 2000</td>
</tr>
<tr>
<td>&lt;&gt; , NE</td>
<td>Not Equal: True, if the content of operand1 does not match the content of operand2.</td>
<td></td>
</tr>
<tr>
<td>&lt;, LT</td>
<td>Lower Than: True, if the content of operand1 is smaller than the content of operand2.</td>
<td></td>
</tr>
<tr>
<td>&gt;, GT</td>
<td>Greater Than: True, if the content of operand1 is greater than the content of operand2.</td>
<td></td>
</tr>
<tr>
<td>&lt;=, LE</td>
<td>Lower Equal: True, if the content of operand1 is lower than or equal to the content of operand2.</td>
<td></td>
</tr>
<tr>
<td>&gt;=, GE</td>
<td>Greater Equal: True, if the content of operand1 is greater than or equal to the content of operand2.</td>
<td></td>
</tr>
<tr>
<td>BT</td>
<td>Between operand1 and operand2</td>
<td></td>
</tr>
</tbody>
</table>

Example:
FIELDNM = BUKRS
SIGN =I
OPTION = BT
LOW = 1000
HIGH = 2000

The above condition means: select the Company codes (BUKRS) that is included (I) in the interval (BT) between 1000 and 2000.

Check the return parameter JOBSTATUS (CHAR 1). If JOBSTATUS = ‘S’ (Started), then there is another job running in SAP for the selected Logical system.
If the transfer method is IDoc only one job can be executed at the same time for one logical system.
If the transfer method is tRFC many jobs can be executed simultaneously for one logical system. However the same Extractor can not be executed twice simultaneously for one logical system.

The function module returns the ‘data transfer request number’ in a field called: INITRNR (CHAR 30)

Use that number as input in steps below.

The function module also returns values in the following fields:
JOBNAME (CHAR 32)
JOBDATE (CHAR 8)
JOBTIME (CHAR 6)

Use these values in steps below.

If not successful the function module will return an error message.

**Update Delta Init database tables the first time**

*Note: Perform this step only if UPDMODE = ‘C’ (Init) is sent to the function module /QTQVC/CREATE_REQUEST_IDOC*

**Function module /QTQVC/DELTAINIT_START**

Use parameters:

- I_OLTPSOURCE (CHAR 30)
- I_RLOGSYS (CHAR 10)
- I_INITRNR (CHAR 30)

Set parameters:

- I_OLTPSOURCE = Extractor name
- I_RLOGSYS = name of Logical system
- I_INITRNR = INITRNR (returned from SAP when calling Function module /QTQVC/CREATE_REQUEST_IDOC)

If not successful the function module will return an error message.

**Update Delta Init database tables the second time**

*Note: Perform this step only if UPDMODE = ‘C’ is sent to the function module /QTQVC/CREATE_REQUEST_IDOC*

It should be performed when the last data IDoc or tRFC package has been received successfully.
Function module RSC1_DELTAINIT_FINISH

Use parameters:
I_OLTPSOURCE (CHAR 30)
I_RLOGSYS (CHAR 10)
I_INITRNR (CHAR 30)

Set parameters:
I_OLTPSOURCE = Extractor name
I_RLOGSYS = name of Logical system
I_INITRNR = INITRNR (returned from SAP when calling FM /QTQVC/CREATE_REQUEST_IDOC)

If not successful the function module will return an error message.

IDOC content

IDOC Structure

An IDoc consists of 3 parts (record types):
- Control record
- Data record(s)
- Status records (not used by the connector)

The control record always consists of 1 record and comes first. It contains header information about the IDoc. The content of the Control record has the structure called IDOC_CONTROL_REC_40 which is used in the SAP function module IDOC_INBOUNDASYNCHRONOUS:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABNAM</td>
<td>CHAR</td>
<td>10</td>
<td>Name of Table Structure</td>
</tr>
<tr>
<td>MANDT</td>
<td>CLNT</td>
<td>3</td>
<td>Client</td>
</tr>
<tr>
<td>DOCNUM</td>
<td>CHAR</td>
<td>16</td>
<td>IDoc number</td>
</tr>
<tr>
<td>DOCREL</td>
<td>CHAR</td>
<td>4</td>
<td>SAP Release for IDoc</td>
</tr>
<tr>
<td>STATUS</td>
<td>CHAR</td>
<td>2</td>
<td>Status of IDoc</td>
</tr>
<tr>
<td>DIRECT</td>
<td>CHAR</td>
<td>1</td>
<td>Direction</td>
</tr>
<tr>
<td>OUTMOD</td>
<td>CHAR</td>
<td>1</td>
<td>Output mode</td>
</tr>
<tr>
<td>EXPRSS</td>
<td>CHAR</td>
<td>1</td>
<td>Overriding in inbound processing</td>
</tr>
<tr>
<td>TEST</td>
<td>CHAR</td>
<td>1</td>
<td>Test flag</td>
</tr>
<tr>
<td>IDOCTYP</td>
<td>CHAR</td>
<td>30</td>
<td>Name of basic type</td>
</tr>
<tr>
<td>CIMTYP</td>
<td>CHAR</td>
<td>30</td>
<td>Extension (defined by customer)</td>
</tr>
<tr>
<td>MESTYP</td>
<td>CHAR</td>
<td>30</td>
<td>Message type</td>
</tr>
</tbody>
</table>

Qlik Connector for SAP
The field MESTYP (bold font) tells us which IDoc Message type that is sent. In our case we will receive the types RSINFO and RSSEND.

The next part contains one or many Data records. A data record has the structure called IDOC_DATA_REC_40 from the SAP function module IDOC_INBOUNDASYNCHRONOUS:

- **SEGNAM** CHAR 30 Segment type
- **MANDT** CLNT 3 Client
- **DOCNUM** CHAR 16 IDoc number
- **SEGNUM** CHAR 6 Number of SAP segment
- **PSGNUM** NUMC 6 Number of the hierarchically higher SAP segment
- **HLEVEL** CHAR 2 Hierarchy level
- **SDATA** LCHR 1000 Application data
The last field SDATA has a variable content. The SDATA part of a Data record can contain different segments (one at a time). The segment type name of the current Data record is found in the field SEGNAM (bold font).

The message type RSINFO can contain three different segment types. We will only look for the one below:

Segment type E1RSHIN has the structure:

```
REQUEST       CHAR       30
INFOIDOCNR    CHAR       6
SELDATE       CHAR       8
SELTIME       CHAR       6
RQSTATE       CHAR       1
RQRECORD      CHAR       12
```

The value of the field RQSTATE (can be found in position 51 of the field SDATA) tells us the current status of the transfer of data IDocs:

<table>
<thead>
<tr>
<th>Value</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Data request received</td>
</tr>
<tr>
<td>1</td>
<td>Data selection started</td>
</tr>
<tr>
<td>2</td>
<td>Data selection running</td>
</tr>
<tr>
<td>5</td>
<td>Error in data selection</td>
</tr>
<tr>
<td>6</td>
<td>Transfer structure obsolete, transfer rules regeneration</td>
</tr>
<tr>
<td>8</td>
<td>No data available, data selection ended</td>
</tr>
<tr>
<td>9</td>
<td>Data selection ended</td>
</tr>
</tbody>
</table>

We will receive a number of IDocs of the type RSINFO during the transfer of data. When the status of an RSINFO IDoc is ‘9’, then the data transfer is finished.

The message type RSSEND is used to send the data. It can contain two different segment types. A header segment called E1RSSH:

```
REQUEST       CHAR       30
REQOBJTYPE    CHAR       1
REQDATE       CHAR       8
REQTIME       CHAR       6
REQUSER       CHAR       12
DATAPAKID     CHAR       6
LOGSYS        CHAR       10
IOBJECT       CHAR       30
SELDATE       CHAR       8
SELTIME       CHAR       6
```
The content of the field REQUEST (position 1-30 of the SDATA field) is useful to link the data IDocs that are connected to the same extraction.

Then comes one or many data segments which are specific to the Extractor used. The segment type name is returned in the parameter SEGNAME (CHAR 30) when calling the function module /QTQVC/GET_ACTIVATED_EXTRACTOR.

**Unpack data from an IDOC segment**

To get the data of the segment type used, call function module /QTQVC/GET_ACTIVATED_EXTRACTOR as described earlier in this document.

The Extractor 0FI_GL_4 contains the following fields:

BUKRS
FISCPER
BELNR
BUZEI
KTOPL
HKONT
WRSOL
WRHAB

The first seven columns of the result table from the function module /QTQVC/GET_ACTIVATED_EXTRACTOR looks like:

<table>
<thead>
<tr>
<th>/BIC/QA0FI_GL_4001</th>
<th>BUKRS</th>
<th>4</th>
<th>4</th>
<th>1</th>
<th>64</th>
<th>67</th>
</tr>
</thead>
<tbody>
<tr>
<td>/BIC/QA0FI_GL_4001</td>
<td>FISCPER</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td>68</td>
<td>74</td>
</tr>
<tr>
<td>/BIC/QA0FI_GL_4001</td>
<td>BELNR</td>
<td>10</td>
<td>10</td>
<td>3</td>
<td>75</td>
<td>84</td>
</tr>
<tr>
<td>/BIC/QA0FI_GL_4001</td>
<td>BUZEI</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>85</td>
<td>87</td>
</tr>
<tr>
<td>/BIC/QA0FI_GL_4001</td>
<td>KTOPL</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>88</td>
<td>91</td>
</tr>
<tr>
<td>/BIC/QA0FI_GL_4001</td>
<td>HKONT</td>
<td>10</td>
<td>10</td>
<td>6</td>
<td>92</td>
<td>101</td>
</tr>
<tr>
<td>/BIC/QA0FI_GL_4001</td>
<td>WRSOL</td>
<td>13</td>
<td>15</td>
<td>7</td>
<td>102</td>
<td>116</td>
</tr>
<tr>
<td>/BIC/QA0FI_GL_4001</td>
<td>WRHAB</td>
<td>13</td>
<td>15</td>
<td>8</td>
<td>117</td>
<td>131</td>
</tr>
</tbody>
</table>

To unpack a data segment of type /BIC/QA0FI_GL_4001, look in the SDATA field:

The content of the field BUKRS (column 2) has the length of 4 (look in column 4) and starts in position 1 and ends in position 4 of SDATA.
The content of the field FISCPER (column 2) has the length of 7 (look in column 4) and starts in position 5 and ends in position 11 of SDATA.
And so on...
Check if Delta Init has been performed previously

Function module /QTQVC/EXTRACTION_STATUS

Use parameters
MODE (CHAR 1)
RLOGSYS (CHAR 10)
OBJECT (CHAR 30)

Set parameters:
MODE = I (check if Init has been performed previously)
RLOGSYS = name of Logical system
OBJECT = Extractor name

If the field STATUS (CHAR 1), which is returned from the function module above contains ‘Y’, then the Init has been performed previously. The function module also returns the existing ‘data transfer request number’ (if there is any) in the field E_INITRNR (CHAR 30). Display warning message.

If the field STATUS contains ‘N’, then the Init has not been performed previously for the Extractor.

Get Message text

Function module /QTQVC/EXTRACTION_STATUS

Use parameters
MODE (CHAR 1)
MSGID (CHAR 20)
MSGNO (CHAR 3)

Set parameters:
MODE = M (get message text)
MSGID = Message ID
MSGNO = Message number
The field MESSAGE_TEXT (CHAR 73) containing the message text is returned. If the field STATUS contains ‘N’, then no text was found. STATUS = ‘Y’ means a successful fetch of the text.

**Check Extraction status**

Call when more than 5 minutes has passed without receiving a new IDoc (not while waiting for the first IDoc) or when the status (RQSTATE) of an RSINFO IDoc is ‘9’ (IDoc creation batch job is finished):

**Function module /QTQVC/EXTRACTION_STATUS**

Used to check the status of the IDoc creation batch job. Repeat calling the module until the returned status = ‘A’ or all IDocs are received.

Use parameters:
- **MODE (CHAR 1)**
- **RLOGSYS (CHAR 10)**
- **JOBNAME (CHAR 32)**
- **JOBDATE (CHAR 8)**
- **JOBTIME (CHAR 6)**
- **INITRNR (CHAR 30)**

Set parameters:
- **MODE** = C (check status of an Extraction job)
- **RLOGSYS** = name of Logical system.
- **JOBNAME** = value from function module /QTQVC/CREATE_REQUEST_IDOC
- **JOBDATE** = value from function module /QTQVC/CREATE_REQUEST_IDOC
- **JOBTIME** = value from function module /QTQVC/CREATE_REQUEST_IDOC
- **INITRNR** = value from function module /QTQVC/CREATE_REQUEST_IDOC

If the field STATUS (CHAR 1), which is returned from the function module above contains ‘A’, then the Extraction job has failed. Display error message.

**STATUS** = N (The Batch job producing IDocs has not started). Repeat calling the module.

**STATUS** = B (The Batch job producing IDocs is ongoing). Repeat calling the module.

If the field STATUS contains ‘S’ then the IDoc table below is filled and the IDoc creation batch job has finished successfully. The function module will return a table called IDOCTABLE containing all IDocs produced for this data transfer request (INITRNR and REQUNR are the same):
If the fields DATAPAKID and INFOPAKID contain zero, this is the IDoc of type RSRQST which was created in SAP to trigger the Extractor job in function module /QTQVC/CREATE_REQUEST_IDOC. Hence this IDoc will not be received.

Check that all the other IDocs in the table are received. If the field ERROR is not equal to zero for some IDoc then that IDoc contains error(s). Display error message.

We do not use the other fields in the table for the moment. The two last fields do not seem to be updated by SAP.

If all IDocs are received, then the Extraction job is finished and there is no need to continue calling the function module /QTQVC/EXTRACTION_STATUS.

If not all IDocs are received, continue to call the function module /QTQVC/EXTRACTION_STATUS.

Below are explanations of possible RFC errors that can be found in database table ARFCSSTATE. If one of those errors are found, the function module deletes the corresponding record in the table ARFCSSTATE and then performs a resend of the corresponding IDoc.

**Calls w/ Execution Errors – SYSFAIL**
Number of tRFC calls that could not be executed due to a problem with the execution in the target system or an external component.

**Calls w/o Server Resources – SYSLOAD**
Number of tRFC and qRFC calls with outbound queue with status SYSLOAD; these are calls that could not be executed due to a lack of resources in the target system.

**Calls w/ Communication Errors – CPICERR**
Number of tRFC calls that were not executed due to problems creating the connection or the communication with the target system or an external component; depending on the settings (transaction SM59), the attempt may be repeated a number of times.

If the field STATUS contains ‘R’, then a Resend of one or many IDocs has been performed.

If the field STATUS contains ‘F’, then no RFC transfers can be found on the SAP side. Check again that all IDocs are received. Otherwise try to call the function module again.
When all IDocs are received:
Check the number of records (RQRECORD) in the last RSINFO record where the status (RQSTATE) is ‘2’.
The number of records should be equal to the number of received records in the RSSEND IDocs.

When all IDocs have been received successfully or the Extraction job has aborted, perform the step ‘Update status table when a job is finished or cancelled’ below.

**Update status table when a job is finished or cancelled**

When all IDocs for a job has been received or the tRFC transfer is finished and the handling on the Windows side is completed. Or when there is a failure in receiving IDocs or tRFC packages, set the JOBSTATUS to finished, aborted or cancelled. The status Finished (F) is used when the job has finished successfully. The status aborted (A) is used for a system or technical error. The status cancelled (C) is used if the job is cancelled by a QlikView user.

**Function module /QTQVC/UPDATE_STATUS**

Used to update the status table /QTQVC/STATUS.

Use parameters
- MODE (CHAR 1)
- CONNECTOR (CHAR 10)
- JOBDATE (CHAR 8)
- JOBTIME (CHAR 6)
- JOBSTATUS (CHAR 1)

Set parameters:
- MODE = U (update status table)
- CONNECTOR = name of Logical system.
- JOBDATE = value from function module /QTQVC/CREATE_REQUEST_IDOC
- JOBTIME = value from function module /QTQVC/CREATE_REQUEST_IDOC
- JOBSTATUS = (A = Aborted, C = cancelled by user or F = finished)

If the operation is not successful, the function module returns an error message.

**Resend an Extraction job**
The used function module below can resend any Extraction job for the IDoc transfer method, but for the moment there is a check against the status table that the job referred to by the ‘data transfer request number’ is a job that has been aborted or cancelled. The ‘data transfer request number’ (INITRNR) is returned to Windows when calling the function module /QTQVC/CREATE_REQUEST_IDOC which is triggering the Extraction job the first time. This number is used as input to the function module below when resending a complete job and has to be logged in Windows.

The function module can also resend a single IDoc or tRFC records that has failed in transfer to QlikView.

Start up the ‘listener’.

**Function module /QTQVC/RESEND_EXTRACTION_JOB**

Use parameters

**For resend of a complete job:**
RLOGSYS (CHAR 10)
INITRNR (CHAR 30)

Set parameters:
RLOGSYS = name of Logical system.
INITRNR = value from function module /QTQVC/CREATE_REQUEST_IDOC

**For resend of a single IDoc:**
IDOC (NUMC 16)

Set parameters:
IDOC = IDoc number.

**For resend of failed tRFC records:**
RLOGSYS (CHAR 10)
DATE_LOW (CHAR 8)
DATE_HIGH (CHAR 8)
USER (CHAR 12)

Set parameters:
RLOGSYS = name of Logical system
DATE_LOW = Resend from date, format type 20130218
DATE_HIGH = Resend to date, format type 20130219
USER = the user that is used to log on to SAP

IF the parameter IDOC is filled, the function module will try to resend a single IDoc regardless of if the other parameters are filled.
IF the parameter USER is filled (and not IDOC is filled), the function module will try to resend failed tRFC records regardless if the other parameters are filled.

Otherwise, the function module will try to resend a complete IDoc extraction job.

The function module gets all IDocs (for transfer method IDoc) included in the extraction job or a single IDoc. It calls a standard SAP function module that changes the status of the IDoc(s) in SAP to 30 (IDoc ready for dispatch). Then it calls another standard SAP program that sends the IDoc(s) again.

The function module also returns values in the following fields (for a single IDoc or failed tRFC records no values are returned):
- JOBNAME (CHAR 32)
- JOBDATE (CHAR 8)
- JOBTIME (CHAR 6)

When all IDocs are received, use the above fields as parameters to the function module /QTQVC/EXTRACTION_STATUS and update the jobstatus in the status table record to ‘F’ (finished).

If the operation is not successful, the function module returns an error message.

Get Hierarchy details

When the user has selected one Hierarchy:

Function module RSAP_REMOTE_HIERARCHY_CATALOG

If no data found, display message ‘No data found for given language’.

Use parameters
- I_CHABASNM (CHAR 30)
- I_LANGU (CHAR 1)

Set parameters:
- I_CHABASNM = name of Hierarchy
- I_LANGU = User input earlier when the available Hierarchies were fetched from SAP

The function module will return a table called E_T_HIERS containing Hierarchy information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHABASNM</td>
<td>CHAR 30</td>
<td>Reference characteristic</td>
</tr>
<tr>
<td>HIENM</td>
<td>CHAR 30</td>
<td>Hierarchy name</td>
</tr>
<tr>
<td>VERSION</td>
<td>CHAR 3</td>
<td>Hierarchy version</td>
</tr>
<tr>
<td>HCLASS</td>
<td>CHAR 4</td>
<td>Hierarchy class</td>
</tr>
</tbody>
</table>
Select one name. Use the selected name, version and language as parameters to the function module /QTQVC/CREATE_REQUEST_IDOC.

If the operation is not successful, the function module returns an error message.

**Timeouts**

There are some Timeout parameters that can be set in the QlikView script in the connection string. If not set in the connection string each Timeout has a default value. All Timeout values are given in seconds.

**TimeoutSAP**

This timeout is used in the SAP part of the Connector. When trying to start a new Extractor job in SAP there is a check to see that no other Extractor job is already started for the given Logical system (true for the IDoc transfer method) or that the same Extractor job is not already running (true for the tRFC transfer method). An Extractor job in SAP consists of two parts. The first part is an Extraction batch job which creates all IDoc:s or tRFC packages containing the extraction data. The second part is the RFC transfer of the IDoc:s or tRFC packages to QlikView. Sometimes there can be interruptions in the RFC transfer. A number of retries to send the data is then performed. The value in the TimeoutSAP parameter decides how long time SAP should wait to start a new job if there is an ongoing RFC transfer of a previous job. When the number of seconds in the timeout is exceeded, it is likely that there is a permanent error in the previous RFC transfer. The previous job is then regarded as failed (the corresponding record in the SAP status table is set to ‘aborted’) and the new job is allowed to start. The default value is 14400 seconds (240 minutes).

**TimeoutInit**

This timeout is used in the Windows part of the Connector. It decides the maximum time Windows should wait to receive data from SAP about the activated Extractors and Hierarchies possible to use. If the timeout is exceeded, the QlikView job is aborted. The default value is 900 seconds (15 minutes).

**TimeoutActivity**

This timeout is used in the Windows part of the Connector. It decides the maximum time Windows
should wait to receive data from SAP created by the Extraction job. If the timeout is exceeded, the QlikView job is aborted and the status value in the corresponding record in the SAP status table is set to ‘aborted’.
The default value is 7200 seconds (120 minutes).

**TimeoutData**
This timeout is used in the Windows part of the Connector. It decides the maximum time Windows should wait between the receiving of each data IDoc/tRFC data package. If the timeout is exceeded, the QlikView job is aborted and the status value in the corresponding record in the SAP status table is set to ‘aborted’.
The default value is 3600 seconds (60 minutes).

**Migrate activated Extractors**

In the first system in the SAP system landscape, put the activated Extractors that should be migrated into the database table /QTQVC/EXTRACT. Use the transaction /QTQVC/MIGRATE_PREP:
Then in the same system in the SAP system landscape, create a Transport request containing activated Extractors for the specific Logical system used above. Perform step 1 in the transaction /QTQVC/MIGRATE_EXTR:
Then perform step 2 and step 3 described in this transaction.

After doing that, execute the same transaction /QTQVC/MIGRATE_EXTR in the SAP system where step 2 and step 3 were performed. Perform step 4:
Then perform step 5 in each system in the SAP system landscape.

Some general rules:

The database table /QTQVC/EXTRACT will contain the activated Extractors from the latest execution of the transaction /QTQVC/MIGRATE_PREP, that is the transaction overwrites the previous content in the table for a Logical system. Therefore it is important to execute the transaction /QTQVC/MIGRATE_PREP before executing the transaction /QTQVC/MIGRATE_EXTR which creates the Transport request. When the Transport request is created it is not allowed to change the content of the database table /QTQVC/EXTRACT until the Transport request has been migrated. No new transport request should be created until the last one has been used in all systems in the landscape.

However the database table /QTQVC/EXTRACT is using Logical system as key, so it is possible to put activated Extractors for a different Logical system in the table and to create a separate Transport request in parallel for that Logical system.
When the Extractors are uploaded and activated in the next system in the landscape there is a check that an Extractor is not already activated. If so, there will be a warning in the log file, the activation of the Extractor is skipped and the program continues with the next Extractor from the Transport request. If the user wants to change the fields of an activated Extractor by using this functionality, the Extractor has to be manually deactivated in each system before uploading and activation takes place.

**Migrate Extractor Environment**

First create the Extractor Environment in the unlocked Source system with the existing transaction /N/QTQVC/EXTRACTOR_ADM.

Then check the database table RSBASIDOC in the Source system with transaction SE16. Enter the name of the Extractor Environment Logical System and click **Execute**:

![Data Browser: Table RSBASIDOC: Selection Screen](image)

![Table RSBASIDOC Display](image)
Then enter the name of the TSPREFIX (Transfer structure prefix) from the Source system in the same table in the TARGET system like below and click **Execute**:

![Data Browser: Table RSBASIDOC: Selection Screen](image)

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
<th>to</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLGDSYS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RLOGSYS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OBJSTAT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIDOCTYP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSIDOC3X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSPREFIX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRCTYPE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAPREL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSTPNM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SBWBCTRL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RBWBCTRL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Width of Output List**: 250  
**Maximum No. of Hits**: 200

If a record is found with the same TSPREFIX in the Target system, the transport function cannot be used (a workaround is to create a new Extractor Environment in the Source system to get another TSPREFIX that is not already used in the Target system).

If the prerequisite above is fulfilled, then execute the transaction /N/QTQVC/MIGRATE_ENV in the Source system to put the Extractor Environment into a transport.

Then migrate and import the transport into the correct client in the Target system.

After the import, execute the transaction /N/QTQVC/CONNECT_ENV in the Target system. Enter Logical system of receiver, User name and Transfer structure prefix (found above in the Source system) and click **Execute**:
Then go to transaction WE21 (under Transactional RFC) in the Target system and generate a new Port name and save it. RFC destination should contain the Extractor Environment Logical System name:

Then go to transaction WE20 and find the partner profile for the Partner Type LS and the Extractor Environment Logical System name.
Double click the Message Type RSINFO:
Assign the Port name which was generated above to **Receiver port**, click **Enter** and **Save**.

Do the same for the **Message Type** RSSEND.

Lastly perform the ‘Modify’ option in the transaction `/n/QTQVC/EXTRACTOR_ADM` to update **Gateway Host** and **Gateway Service** of the RFC connection used by the Logical system.

The Extractor Environment is ready to use.