

Integrating SAP-BW Process Chains with QlikView Publisher

How-to paper for creating a synchronous interface for data load processes between SAP-BW and QlikView.

November 2013

Written by:

Yaniv Feldman – Senior BI developer and Project Manager in Netafim Ltd.

Email to: yaniv.feldman@netafim.com

Contents

Introduction.....	3
Step 1: Create a Publisher Task triggered by EDX.....	4
Step 2: Install <i>QMSEDX</i> in the Publisher machine	5
Step 3: Install <i>SAP Netweaver RFC SDK</i> in the Publisher machine.....	6
Step 4: Create an RFC server in the Publisher machine.....	8
Step 5: Create an RFC Destination in SAP BW	10
Step 6: Create an ABAP program in SAP BW.....	11
Step 7: Create a Process Type in SAP-BW.....	12
Step 8: Create a Process Chain in SAP-BW	13
Step 9: Make sure everything works	14
Appendix 1	15

Introduction

Many big enterprises use QlikView as a reporting tool and SAP-BW as a data-warehouse. SAP-BW offers the *Process-Chains* as a robust visual tool for managing and monitoring data loads. QlikView offers the *Publisher* which supports the same. However, no reliable interface exists between the two thus making it very difficult for BI teams cope with numerous, complicated and parallel data loads which are required to get the data up to date and reports distributed on-time.

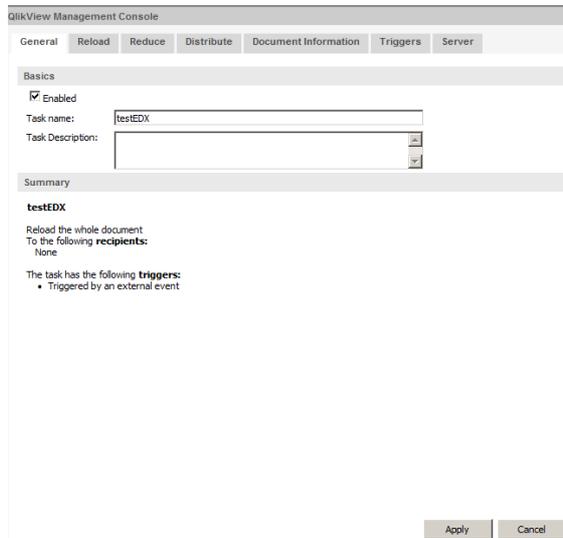
The purpose of this article is to explain how to create a reliable synchronous interface between SAP-BW and Publisher by fully integrating *Process Chains* with *Publisher Tasks* using RFC connectivity and EDX calls. By doing so, BI teams can harness the power and stability of the BW Process Chains and utilize it with the QlikView Publisher.

This paper addresses SAP-BW and QlikView developers/administrators and requires basic understanding of both.

The products involved in this paper are SAP-BW 7.01 SP8 and QlikView Server 11 SR2.

Step 1: Create a Publisher Task triggered by EDX

- Open the QMC and create a task triggered by EDX.
- Make sure the task name is *testEDX*



The screenshot shows the 'QlikView Management Console' with the 'Triggers' tab selected. The 'Basics' section is active, showing a task named 'testEDX'. The task is enabled, and its description is empty. The 'Summary' section shows that the task is triggered by an external event.

QlikView Management Console

General Reload Reduce Distribute Document Information Triggers Server

Basics

Enabled

Task name: testEDX

Task Description:

Summary

testEDX

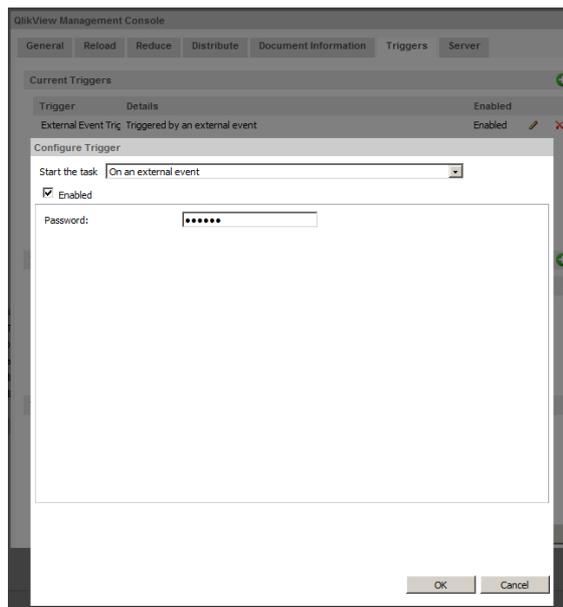
Reload the whole document
To the following recipients:
None

The task has the following triggers:

- Triggered by an external event

Apply Cancel

- Make sure the EDX password is *123456*



The screenshot shows the 'QlikView Management Console' with the 'Triggers' tab selected. The 'Current Triggers' section shows a list of triggers, including 'External Event Trig' which is triggered by an external event. The 'Configure Trigger' dialog is open, showing the configuration for the external event trigger. The trigger is enabled, and the password is set to '123456'.

QlikView Management Console

General Reload Reduce Distribute Document Information Triggers Server

Current Triggers

Trigger	Details	Enabled
External Event Trig	Triggered by an external event	Enabled

Configure Trigger

Start the task On an external event

Enabled

Password: 123456

OK Cancel

Step 2: Install QMSEDX in the Publisher machine

QMSEDX is a windows executable command-line that calls Publisher tasks using EDX calls.

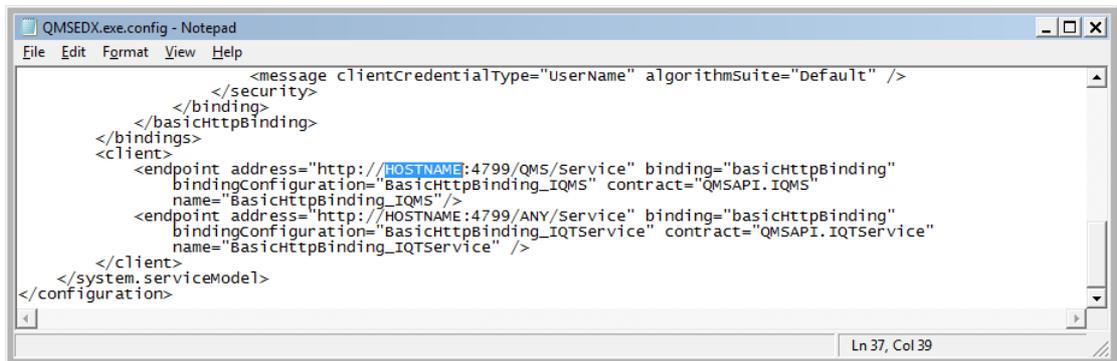
- Go to QlikCommunity to this address:
<http://community.qlikview.com/docs/DOC-2650>
This article is how-to guide for trigger Publisher Tasks using EDX via command-line. Read the article carefully and follow all the instructions.
- Download *QMSEDX_CommandLine_V1.exe* from QlikCommunity website:
http://community.qlikview.com/servlet/JiveServlet/download/2650-11-28526/QMSEDX_CommandLine_v1.exe
- Run *QMSEDX_CommandLine_V1.exe* in the Publisher machine. When prompted for installation route, type-in the following:

```
\\<Publisher Host Name>\c$\QlikView\EDX\
```

- After installation, go to folder:

```
\\<Publisher Host Name>\c$\QlikView\EDX\QMSEDX_CommandLine
```

- Open the file *QMSEDX.exe.config* using Notepad:
Go to line 37 and look for "HOSTNAME". Replace HOSTNAME with your Publisher machine host name. Do the same in line 41.



- Make sure everything works by opening a CMD session and executing the following command:

```
C:\QlikView\EDX\QMSEDX_CommandLine\qmsedx.exe -task=testEDX -  
qms=http://<YourPublisherHostName>:4799/QMS/Service -password=123456
```

- After executing the command open the QMC monitor and make sure that the task is running.

Step 3: Install SAP Netweaver RFC SDK in the Publisher machine

SAP Netweaver RFC SDK enables SAP customers and partners to develop 3rd party tools that communicate with SAP systems using RFC technology. The SDK includes a ready-made executable that we'll use.

- Ask your SAP Basis team to download SAP Netweaver RFC SDK. Make sure you download the correct SDK according to your BW installation and the Operating System that runs on your Publisher machine. Here's the URL for the download page:

<https://websmp210.sap-ag.de/~form/handler? APP=00200682500000001943& EVENT%20=DISPHIER&HEADER=N&FUNCTIONBAR=N&EVENT=TREE&TMPL%20=01200314690200010197&V=M AINT>

- Ask the Basis team to extract the content of the downloaded file to the following folder in your Publisher machine:

\\<Publisher Host Name>\d\$\Install\Rfc_win\

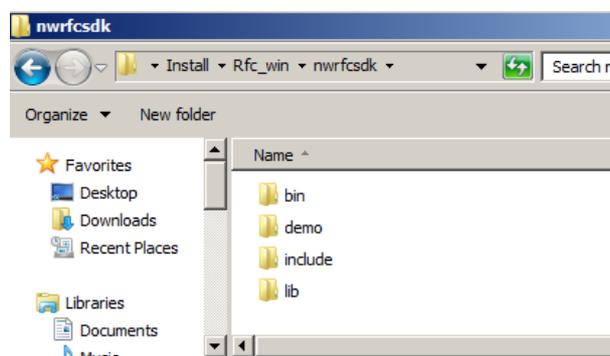
Here is a full 'How-to' link for the Basis team:

<http://wiki.sdn.sap.com/wiki/display/ABAPConn/Download+and+Installation+of+NW+RFC+SDK>

After the SDK was extracted you should see the following route in your Publisher machine:

\\<Publisher Host Name>\d\$\Install\Rfc_win\nwrfcsdk\

The route should include 4 sub-folders:



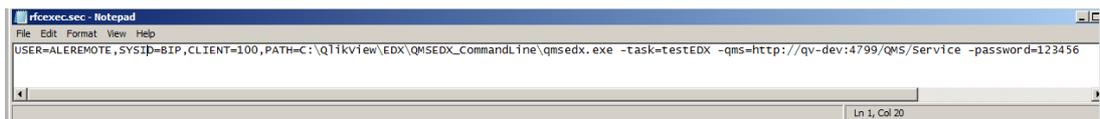
- Go to folder *bin* and Copy the file *rfcexec.exe* to folder *lib*. This executable is intended to 'listen' to an SAP system and launch executables according to commands coming from SAP.

- Go to folder *lib* and create a new text file called *rfcexec.sec*
This file provides security detailed by: SAP system + client, SAP username and executable file name. Commands that are not detailed in this file will not be executed. Edit the file in Notepad and create the following entry in a single line:

```
USER=ALEREMOTE, SYSID=<YourSAPBWSYSTEMID>, CLIENT=<YourSAPBWClientNumber>, PATH=C:\QlikView\EDX\QMSEDX_CommandLine\qmsedx.exe -task=testEDX -qms=http://<YourPublisherHostName>:4799/QMS/Service -password=123456
```

- * SYSID could be for example: *BIP* (3 chars)
- * CLIENT could be for example: *100* (3 digit number)
- * We put value *testEDX* in *-task* parameter. This means that the command will call a Publisher task with the same name.
- * We put value *123456* in *-password* parameter. This means that command will call a Publisher task triggered by EDX with password *123456*.
- * Please make sure that there are no blank lines in the file.

This is how it should look:



- Save and close *rfcexec.sec*

Step 4: Create an RFC server in the Publisher machine

- Make sure you have the file *srvany.exe* located in the publisher machine in the following route:

```
<..\Windows\System32>
```

srvany.exe is a Microsoft executable which runs simple .exe files as services.

- In the Publisher machine, start a CMD session with following command
`sc create QVrfcexec binpath= c:\windows\system32\srvany.exe`

This command will create a service in the Windows Server Manager.

- In the Publisher machine, start *regedit*.

Lookup the following registry route:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\QVrfcexec\Parameters
```

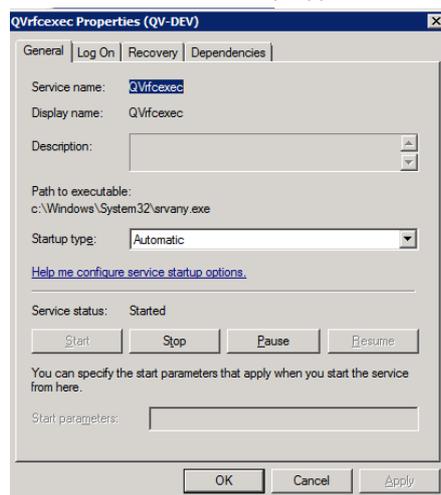
Maintain the *Application* parameter:

```
D:\Install\Rfc_win\nwrfcsdk\lib\rfcexec.exe -a  
PUBLISHER -g <YourSAPBWHostGatewayName> -x  
<YourSAPBWGatewayServiceName> -f  
D:\Install\Rfc_win\nwrfcsdk\lib\rfcexec.sec
```

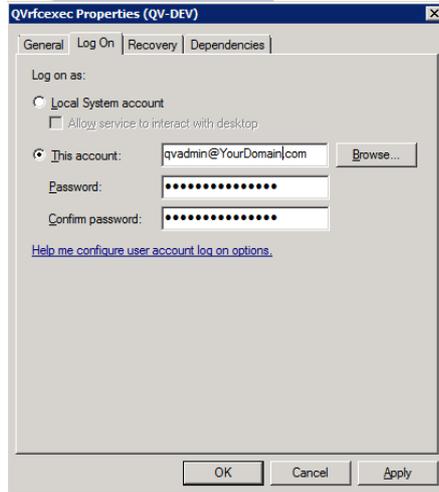
Exit *regedit*.

The service will run *rfcexec.exe*. We will use it to run *qmsedx.exe* which will trigger EDX tasks.

- In the Publisher machine, go to *Server Manager* → *Configuration* → *Services*. Look for the service *QVrfcexec* and edit it.
- Make sure that Startup type is Automatic:



- Assign a proper user for the service. Make sure that this user is a member of the following groups in your Publisher machine: *QlikView Administrators*, *QlikView Management API*, *QlikView EDX*:



- Start the service

Step 5: Create an RFC Destination in SAP BW

- Ask your SAP Basis team to create a new RFC Destination in SAP-BW. The RFC Destination type should be *TCP/IP* (type 'T') with the following parameters:
 - Activation Type: *Registered Sever Program*
 - Program ID: *PUBLISHER*
 - Gateway Host: *<Your SAP-BW Gateway Host Name>*
 - Gateway service: *<Your SAP-BW Gatway Service name>*

The screenshot shows the SAP configuration interface for an RFC Destination named 'QV-DEV PUBLISHER'. The 'Connection Type' is set to 'TCP/IP Connection'. The 'Description' field contains 'QV-DEV PUBLISHER'. Under the 'Activation Type' section, 'Registered Server Program' is selected. The 'Program ID' is 'PUBLISHER'. Under 'Start Type of External Program', 'Default Gateway Value' is selected. The 'CPI-C Timeout' is set to 'Default Gateway Value' with a value of 60 seconds. Under 'Gateway Options', the 'Gateway Host' is 'YourGWHostName.com' and the 'Gateway service' is 'YourGWServiceName'.

- Make sure the RFC connection is OK by checking connectivity:

RFC - Connection Test

Connection Test QV-DEV PUBLISHER
Connection Type TCP/IP Connection

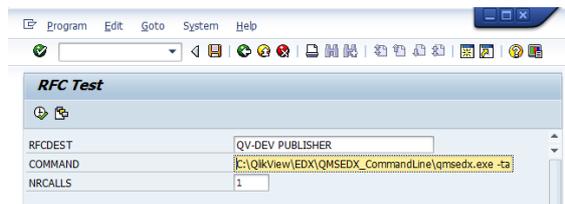
Action	Result
Logon	3 msec
Transfer of 0 KB	11 msec
Transfer of 10 KB	16 msec
Transfer of 20 KB	16 msec
Transfer of 30 KB	26 msec

Step 6: Create an ABAP program in SAP BW

- Log on to SAP-BW. Make sure that you logged on to the System ID and Client that you entered in the *rfcexec.sec* file.
- Ask your ABAP team to create a new ABAP executable program called *ZSRFCEXEC* based on the code in Appendix 1.
- Go to transaction SE38 and execute report *ZSRFCEXEC*.

Enter the following text in the *COMMAND* parameter:

```
C:\QlikView\EDX\QMSEDX_CommandLine\qmsedx.exe -task=testEDX -  
qms=http://<YourPublisherHostName>:4799/QMS/Service -  
password=123456
```



- Save these values as a new variant named *ZtestEDX*

Step 7: Create a Process Type in SAP-BW

- In SAP-BW, go to RSPC, choose *Maintain Process Types*:
- Create a new Process Type called ZABAP:

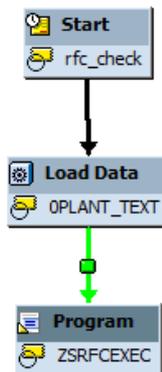
The screenshot shows the SAP BW 'Possible Process Types' details screen for the process type ZABAP. The interface includes a menu bar with 'Table View', 'Edit', 'Goto', 'Selection', 'Utilities(M)', 'System', and 'Help'. Below the menu is a toolbar with various icons. The main content area is titled 'Display View "Possible Process Types": Details' and contains the following fields:

Process Type	ZABAP
Possible Process Types	
Short description	Program
Long description	ZABAP Program.
ObjectName	CL_RSFC_ABAP
Object Type	ABAP OO Class
Possible Events	Process ends "successful" or "incorrect"
<input checked="" type="checkbox"/> Repeatable	
<input type="checkbox"/> Repairable	
ID	
<input type="checkbox"/> Internal Name	
<input type="checkbox"/> Own Mail	
Process Category	5
Two-digit no.	5
Documentation Type	Free Text
Docu. Object	RSFC_ABAP
<input type="checkbox"/> Only in BW Clients	
Component	

This Process type executes an ABAP report that can end in 2 possible ways: *successful* or *incorrect*. This is important because it will enable us to monitor the success of EDX calls in the Process Chain.

Step 8: Create a Process Chain in SAP-BW

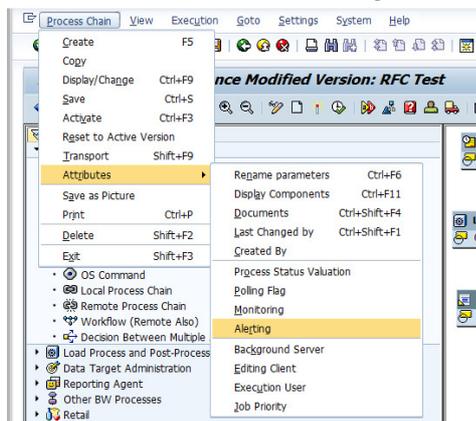
- Create a new Process Chain.
- Create a ZABAP process as the last step



In this example we schedule load of *OPLANT_TEXT*. The Publisher task will be triggered only if *OPLANT_TEXT* was completed successfully.

- Make sure that the ZABAP step is calling report ZSRFCEXEC with variant ZtestEDX

- Make sure that Alerts are being send whenever the Process Chain fails:



Mark Send Alerts If Errors Occur

- Activate the Process Chain

Step 9: Make sure everything works

- Execute the new process chain and check its status.
- If the Process Chain finished successfully – the Publisher Task should be running.
- If the Process Chain failed, please check the Process Chain log:

Failure in the report *ZSRFCEXEC* may be caused by two non-zero return-codes:

1. *Access Denied* (RC=1) – this means that the command entered in *ZSRFCEXEC* doesn't match the command in *srfcexec.sec* . In this case, the command is not authorized hence not executed.
2. *Error when opening an RFC connection* (RC=2) – this means that the service in the Publisher is not communicating with SAP-BW. This may indicate the service in the Publisher machine is not running or that something was configured wrong. In this case – go through all this document again and make sure that everything was done properly.

Appendix 1

Source Code for ABAP report ZSRFCEXEC.

```
REPORT ZSRFCEXEC.
```

```
"This code is based on SAP standard ABAP report SRFCEXEC which  
"utilizes the RFCCEXEC server. Report ZSRFCEXEC includes system  
"messages and return codes in order to reflect the success and  
"failure of the RFC call that will determine the status of the  
"Process Chain.
```

```
SET EXTENDED CHECK OFF.
```

```
PARAMETERS: RFCDEST LIKE rfcdes-rfcdest DEFAULT 'QV-DEV PUBLISHER',  
            COMMAND(512) LOWER CASE,  
            NRCALLS(5) DEFAULT '1'.
```

```
DATA: RFC_MESS(128),  
      NRCOUNT(5),  
      STARTIME TYPE I,  
      ACTUTIME TYPE I,  
      REQUTIME TYPE I.
```

```
DATA: BEGIN OF PIPEDATA OCCURS 1,  
      LINE(80),  
      END OF PIPEDATA.
```

```
IF COMMAND(1) = SPACE.  
  WRITE: / 'Missing command'.  
  EXIT.  
ENDIF.
```

```
NRCOUNT = 1.  
DO NRCALLS TIMES.  
  CLEAR PIPEDATA.  
  REFRESH PIPEDATA.  
  GET RUN TIME FIELD STARTIME.  
  CALL FUNCTION 'RFC_REMOTE_EXEC'  
    DESTINATION RFCDEST  
    EXPORTING  
      COMMAND = COMMAND  
    TABLES  
      PIPEDATA = PIPEDATA  
    EXCEPTIONS  
      SYSTEM_FAILURE = 1 MESSAGE RFC_MESS  
      COMMUNICATION_FAILURE = 2 MESSAGE RFC_MESS.
```

```
IF SY-SUBRC NE 0.  
  message e000(r1) with RFC_MESS ' ; SY-SUBRC=' SY-SUBRC. " 'RFC  
  Connection Failed'.  
  WRITE: / 'Call RFC_REMOTE_EXEC          SY-SUBRC = ', SY-SUBRC.
```

```

    DETAIL.
    WRITE: / RFC_MESS.
    EXIT.
ENDIF.

GET RUN TIME FIELD ACTUTIME.
IF ACTUTIME > STARTIME.
    REQUTIME = ACTUTIME - STARTIME.
    REQUTIME = REQUTIME / 1000.
ELSE.
    REQUTIME = 0.
ENDIF.
WRITE: / 'Call RFC_REMOTE_EXEC           Destination: ', RFCDEST.
SKIP 1.
SUMMARY.
WRITE: / 'Required Time for RFC: ', REQUTIME, 'ms'.
SKIP 1.
SUMMARY.
WRITE: / 'Command for remote executing:'.
DETAIL.
WRITE: / COMMAND.
SKIP 1.
SUMMARY.
WRITE: / 'End of RFC call no.', NRCOUNT.
NRCOUNT = NRCOUNT + 1.
WRITE: / '-----'
-'.
SKIP 1.
ENDDO.

```