BC - XBP
Background Processing, Job Scheduling System Test Catalogue for BC-XBP 6.10 (Version 2.0) and BC-XBP 7.0 (Version 3.0)

External Interface for Background Processing

Prerequisites regarding the certification of BC-XBP 2.0 and / or 3.0.

XBP 3.0 available with support package 16 of SAP NetWeaver release 7.00 is an enhancement of XBP 2.0.

XBP 2.0 is available with Releases / Basis Releases :
4.6C: SP 41
4.6D: SP 35
610: SP 28
620: SP 17
All newer Basis Releases: from the beginning

Third-party products cannot be certified for XBP 3.0 without having been certified for XBP 2.0. The two certifications can be done together or separately, depending on the certification scenarios in questions (see below).

This document contains the test catalogues for XBP 2.0 and XBP 3.0.
Please note that this document has two page numberings.
In the upper right corner you see the running page numbers of the whole document.
According to this numbering

the XBP 2.0 test catalogue starts on page 4
the XBP 3.0 test catalogue starts on page 27.

On the bottom you see the page numbering of the two individual test catalogues for XBP 2.0 and XBP 3.0. The page numbers given in the contents of the test catalogues are the individual page numbers of the test catalogue.

If a software vendor wants to get certified for XBP 2.0 only (BC-XBP 2.0 standalone scenario), he has to pass all mandatory test scenarios of the XBP 2.0 test catalogue. The XBP 2.0 test catalogue contained in this document has undergone one slight change: The former test scenario 7 (error handling) is not mandatory any more, but now optional. It is now up to the external scheduler, which error message is displayed in its user interface.

If a software vendor wants to get certified for XBP 3.0, there are two possible scenarios:

1. The software vendor has an existing and certified XBP 2.0 product (BC-XBP 3.0 standalone scenario).

In this case the vendor has to pass all scenarios in the XBP 3.0 test catalogue. The scenarios tested with XBP 2.0 do not need to be tested again nor is it necessary to go through a XBP 2.0 certification if the existing software vendor product is still certified (not expired etc.).

2. The software vendor does not have a product certified for XBP 2.0. Similarly, the existing certified software product certified for XBP 2.0 has expired (BC-XBP 2.0 and 3.0 combination scenario).

In this case, all mandatory test scenarios in the XBP 2.0 and XBP 3.0 test catalogue must be tested. The optional test scenarios in XBP 2.0 still remain optional.

FAQ:

1. Question: With XBP 2.0 the variant functions of the function group SVAR_RFC were shipped (for example RS_CREATE_VARIANT_RFC).
XBP 3.0 now contains new variant functions of the function group SXBP_VAR.
Why? Will the old variant functions (SVAR_RFC) be obsolete in the future?

Answer: The functions of SVAR_RFC were regarded as a bit complicated, and moreover some software vendors demanded more functionality (e.g. reading the selection screen of an ABAP program).
Therefore, a complete new group of functions (SXBP_VAR) are shipped with XBP 3.0. These functions do not only cover the functionality of SVAR_RFC, but exceed it.

However: The function group SVAR_RFC is and will remain part of XBP 2.0.
If an external scheduler logs on to the SAP system with XBP version number **2.0**, it has to call the functions of SVAR_RFC for operations on variants.
If an external scheduler logs on to the SAP system with XBP version number **3.0**, it may only use the functions of SXBP_VAR for operations on variants.

Therefore, the functions of SVAR_RFC are part of the XBP 2.0 certification, and the functions of SXBP_VAR are part of the XBP 3.0 certification.

2. Question: The test catalogue for XBP 2.0 contains the optional part ‘Error handling’. In this chapter it is tested, that the external scheduler receives and displays a certain error message in a certain error situation.
Why is such a test not part of the XBP 3.0 certification?

Answer: It became obvious that there are of course more sources of error than error messages in the system. Therefore, in many error situations, the error message text which was returned by the XBP function to the external scheduler, was not precise enough.
Starting with XBP 3.0, the possible set of error messages which an XBP function can return, is not limited any more. SAP reserves the right to substitute imprecise error messages with more precise ones.
Thus, if an external scheduler logs on to the SAP system with XBP version number 3.0, it cannot expect a certain error message in a certain error situation.
Of course, the format of the error structure (BAPIRET2) and the message type (E = error) remain unchanged.

BC - XBP 6.10
Background Processing,
Job Scheduling System
WAS 6.10 (Version 2.0)
Test Catalogue

External Interface for
Background Processing

Copyright

Contents

Symbols

Symbol	Meaning
⚠	Warning
💬	Example
💡	Tip
🧭	Recommendation
SYN	Syntax

1 Overview

This document describes the technical certification for the XBP 2.0 interface of an external job scheduler.
You should read the manuals *BC-XMI Interface Description* and *XBP Interface Version 2.0* before reading this document.
This documentation is valid only together with a consulting appointment and the verbal explanation provided there.

1.1 Certification Requirements

The following requirements should be met by external schedulers connected to the SAP system using XMI and XBP functions on the basis of RFC :

- The external job scheduling system must be able to fulfill the steps in the following test catalogue in a manner, that means the results can be reviewed.
- The external job scheduling system must also display the results of the tests in an appropriate way in its user interface.
- The external scheduler must not use other function calls to the SAP system than XMI and XBP functions and the functions for variant management (function group SVAR_RFC).
- The test contains a mandatory and an optional part. The reason, why there is an optional part, and the 'rules' for the optional part are explained in the optional part itself. Please read the optional part in any case, because some tests of the optional part may be mandatory for you.

2 Structure of the test plan

For the certification of an external job scheduling system it is required, that all the XBP functions are used in the correct way. This includes the logon using the XMI interface in order to create a (XMI-) session. External job schedulers should call the XBP functions only from within XMI sessions. Otherwise the XBP functions return an error.

The test catalogue is structured as follows:

Chapter 3	Contains the tests of the elementary functions such as creating a job, starting a job, changing a job, reading job data and so on. This test is almost the same as for XBP 1.0. (mandatory)
Chapter 4	Contains a test regarding print parameters for a job step. (mandatory)
Chapter 5	Contains the tests regarding the parent/child functionality. (mandatory)
Chapter 6	Contains the tests regarding the Intercept functionality. (mandatory)
Chapter 7	Contains a test for the error handling of the external scheduler. (mandatory)
Chapter 8	Contains a test for changing values of a report variant. (mandatory)
Chapter 9	Optional part
Chapter 10	Is the appendix and contains the source code of two ABAP programs needed for some tests.

Important remarks for testing:

a) The tests should be done in the same order as listed, because some tests depend on the results of a previous one. Due to the way some schedulers work, certain test scenarios can be done ‘in one step’, e.g. creating and starting a job.

b) Some tests may have a certain precondition, for example the creation of a job. This precondition has to be fulfilled by the external scheduler through the XBP interface, unless otherwise stated.

c) With XBP 2.0, the SAP Background Processing System introduces two new major functionalites, the *Parent Child Functionality* and *Job Interception*.
They are described in detail in the XBP 2.0 documentation.
The accessibility of these new functionalities through XBP 2.0 is a big part of this test catalogue.
As also mentioned in the XBP 2.0 documentation, the two new functionalities can be completely switched on or off. By default, they are switched off.
One possibility to switch them on or off is the ABAP program INITXBP2.
For some test scenarios it may be necessary to explicitly to switch them on or off.
Moreover, it may be necessary to create or delete *intercept criteria*. This can be done by adding an entry to the table TBCICPT1 using transaction SE16.

3 Basic Functions

3.1 Creating jobs with ABAP steps

Description

Create two jobs XBP_TEST_1 and XBP_TEST_2 with at least two ABAP steps each.
For specifying the ABAP reports as job steps, the scheduler has to offer a value help.

Prerequisites

A connection to an SAP SAP R/3 System is established.
An XBP session has to be established through the XMI interface.

Realization

The jobs should be created using the function modules
BAPI_XBP_JOB_OPEN
BAPI_XBP_JOB_ADD_ABAP_STEP (or BAPI_XBP_ADD_JOB_STEP)
BAPI_XBP_JOB_CLOSE

For the value help for the ABAP reports, the function BAPI_XBP_REPORT_SEARCH has to be used.

For example, use the programs RSPARAM and SHOWCOLO as ABAP steps.
XBP_TEST_1 should have job class A.
XBP_TEST_2 should have job class C.

Test

1. In the SAP job overview (transaction transaction SM37) the two jobs appear. The job class can be verified in the job details.

2. The Job details/Step list shows the correct values

3.2 Starting a job (asap)

Description

Start the job XBP_TEST_1 through BAPI_XBP_JOB_START_ASAP.

Prerequisites

Make sure that Job Interception is switched off.
A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
The job XBP_TEST_1 is scheduled and managed within the external job scheduling system.

Realization

Use BAPI_XBP_JOB_START_ASAP.and specify a target server.

Test

1. In the SAP job overview (transaction SM37), the job XBP_TEST_1 appears as *released* (with a delay), *ready*, *running*, or *finished*.

3.3 Starting a job (immediately)

Description

Start the job XBP_TEST_2 using BAPI_XBP_JOB_IMMEDIATELY.

Prerequisites

Make sure that Job Interception is switched off.
A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
The job XBP_TEST_2 is scheduled and managed within the external job scheduling system.
There are free batch resources.

Realization

a) Use BAPI_XBP_JOB_START_IMMEDIATELY and specify a target server..

Test

The job XBP_TEST_2 appears in the SAP job overview (transaction SM37) as *released* (with a delay), *ready*, *running*, or *finished*.

3.4 Monitoring a job (reading the status)

Description

The status of a job should be read.

Prerequisites

Make sure that job interception is switched off.
A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
Create and start a job XBP_TEST_3 with step BTCLOOP (it is an endless loop) .

Realization

Read the status of the job XBP_TEST_3 with BAPI_XBP_JOB_STATUS_GET.
After performing the next test scenario 3.5. read the job status again.

Test

The returned status should be *running* first case and *cancelled* in the second case.

3.5 Canceling a job

Description

The job XBP_TEST_3 should be cancelled.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
The job XBP_TEST_3 is managed within the external job scheduling system.

Realization

Use BAPI_XBP_JOB_ABORT.

Test

The job should be shown as cancelled in transaction SM37.

3.6 Deleting a job

Description

The job XBP_TEST_3 should be deleted.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
The job XBP_TEST_3 is managed within the external job scheduling system.

Realization

Use BAPI_XBP_JOB_DELETE

Test

The Job XBP_TEST_3 should not appear any more in transaction SM37.

3.7 Raising an event

Description

Raise the event SAP_TEST.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
Create within the SAP system a job XBP_TEST_4, which waits for event SAP_TEST.

Realization

Use the function BAPI_XBP_EVENT_RAISE.

Test

The job XBP_TEST_4 should run after raising the event. Check in transaction SM37.

3.8 Reading the job log

Description

Read the job log of the job XBP_TEST_4.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.

The job XBP_TEST_4 is managed by the external job scheduler.

Realization

Use BAPI_XBP_JOB_JOBLOG_READ.

Test

The job log in the SAP system is identical with the job log shown by the external job scheduler.

3.9 Copying a job

Description

Create a job by copying the job XBP_TEST_4 to XBP_TEST_5.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
The job XBP_TEST_5 is managed by the external job scheduler.

Realization

Use BAPI_XBP_JOB_COPY.

Test

Check in transaction SM37, that the jobs XBP_TEST_4 and XBP_TEST_5 have the same job data (except start condition). The new job XBP_TEST_5 should be in status *scheduled* without start condition.

3.10 Selecting jobs

Description

Select all jobs that have been created during this certification procedure by the SAP user, under which the external scheduler logs on to the SAP system.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.

Realization

Use BAPI_XBP_JOB_SELECT.

Test

Compare with sm37.

3.11 Reading the variants of a report

Description

Read all variants of a given ABAP report in a certain client. Note, that variants (except system variants starting with SAP&) are client-dependent.
The report can be chosen during the test; it should have at least 5 variants.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.

Realization

Schedule a job XBP_TEST_6 with the above report as a job step. While adding this step to the job, the user of the external scheduler should be able to display a list with all variants of this report. Use BAPI_XBP_VARIANT_INFO_GET.

Test

Display the variants of the report, for example e.g. in transaction se38. They should be the same as those displayed by the external scheduler.

3.12 Setting the Audit Level XMI

Description

Set the audit level for the XMI logging higher/lower. This function is mandatory, because it is the only way to set the audit level within the XMI interface.

Explanation:
The XMI framework, which allows external Management Tools to log on to the SAP system and create XMI-sessions, also offers the possibility of writing a trace of the functions called during an XMI-session. The trace level can be set with the function BAPI_XMI_SET_AUDITLEVEL and can be viewed in the SAP system using transaction rz15. In the terminology of XMI, the trace level is called audit level.
If the function BAPI_XMI_SET_AUDITLEVEL is not called explicitly in an XMI session, the audit level is 0 by default. Most XBP functions that make changes of some description in the system (create jobs, change jobs etc.), are already traced at audit level 0. Most XBP functions that only read information (such as BAPI_XBP_JOB_STATUS_GET) are only traced with audit level 2 or higher.

Prerequisites

A connection to an SAP R/3 System is established. A (XBP) session has to be established through the XMI interface.

Realization

The audit level of the XMI interface should be raised to 2 using the external job scheduling system. Use BAPI_XMI_SET_AUDITLEVEL . Then a status check of an existing job should be performed by the external scheduler (see 3.8)

If everything worked fine, set the audit level back to the default value 0.

Test

Check, if there are additional messages in the XMI log for the call of BAPI_XBP_JOB_STATUS_GET.

3.13 BAP_XMI_LOGON and BAPI_XMI_LOGOFF

These two functions are not tested separately. If all the previous tests ran without problem with the XMI logon / logoff, these functions are correctly used by the external job scheduler.

4 Print Parameters

4.1 Specifying print parameters for a job step

Description

Create a job XBP_PRIPAR that has report RSWATCH0 as its only step.
In the print specifications of the job step specify the following:

Output device = <existing printer>
Title of spool request = XBP 2.0 Certification
Department = SAP Basis

For specifying the output device, a value help has to be offered by the scheduler.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.

Realization

Create the job as described in 3.1.
For the print parameters, use the appropriate parameter structure of BAPI_XBP_JOB_ADD_ABAP_STEP or BAPI_XBP_ADD_JOB_STEP.
For the value help for the output device, the function BAPI_XBP_OUTPUT_DEVICE_SEARCH has to be used.

Test

After creating the job, check the print parameters using transaction SM37. The print parameters of a job step can be found by displaying the step details and the clicking ‘Print specifications’.

5 Parent/child functionality

For the following tests a report ZMAKECHILD is needed, which creates n jobs CHILD_1, …., CHILD_n.
For testing, n = 4, and one of the four jobs should execute BTCLOOP as a job step.

SAP will provide such a report ZMAKECHILD of this type (see also appendix).
For the following tests, the parent/child function must be switched on, the intercept functionality must be switched off.

5.1 Starting a parent job

Description

Create a job PARENT that has report ZMAKECHILD its only step.
Then start the job PARENT immediately.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.

Realization

Create and start the job as described in 3.1 and 3.2 / 3.3.

Test

The job PARENT and the four child jobs should be displayed by transaction SM37.

5.2 Retrieving the child jobs of a job

Description

Retrieve the child jobs of the job PARENT (started in 5.1).

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
The job PARENT is managed within the external job scheduling system.

Realization

Use BAPI_XBP_JOB_CHILDREN_GET.

Test

Compare the returned job names and job counts with those from transaction SM37.

5.3 Reading the status of the child jobs of a job

Description

Read the status of the child jobs of the job PARENT (started in 5.1).

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
The job PARENT is managed within the external job scheduling system.

Realization

Use BAPI_XBP_JOBLIST_STATUS_GET.

Test

Compare the delivered status data with the data shown in transaction SM37.

6 Job Interception

For the following tests, job interception has to be switched on.
This is done by executing the program INITXBP2
If there are some intercepted jobs in the SAP system from earlier testing, they have to be deleted.

6.1 Making an intercept entry

Description

Make an intercept entry of the form < *client,* BCXBP610I, ICP_TEST* >, where *client* is the client, in which the certification test is performed.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.

Realization

Use BAPI_XBP_MODIFY_CRITERIA_TABLE.

Test

Logon to the SAP system under the user, for which the intercept entry has been made.
Create in transaction SM36 the two jobs ICP_TEST_1 and ICP_TEST_2 with immediate start. Check their status in transaction SM37. It should be *scheduled* (in newer releases it is actually displayed as *intercepted*).
Check also the table TBCICPT1. The intercept criteria specified above must be there.

6.2 Selecting all intercepted jobs

Description

Show all intercepted jobs.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.

Realization

Use BAPI_XBP_GET_INTERCEPTED_JOBS.

Test

The two jobs ICP_TEST_1 and ICP_TEST_2 should be displayed. Check their status also in transaction SM37. It should be *scheduled*.

6.3 Starting an intercepted job ‘asap’

Description

Start the job ICP_TEST_1 (or both) as soon as possible.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
The job ICP_TEST_1 is managed within the external job scheduling system.

Realization

Use BAPI_XBP_JOB_START_ASAP.

Test

The job ICP_TEST_1 (or both) should appear in transaction SM37 as *released* with a delay, *ready*, *running*, or *finished* (or even *aborted*).

7 Variants

Note: The functions dealing with variants belong to the function group SVAR_RFC.

7.1 Changing an existing variant of a report

Description

Change at least one parameter field and at least one select option (low and high value) of an existing variant of a report. Use, for example, the report ZVARI in the appendix.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.

Realization

Use the function RS_CHANGE_CREATED_VARIANT_RFC.

Test

Check the variant using transaction SE38.

8 Optional Part

The mandatory part tested basic functionality, which external job schedulers are required to fullfill in order to get XBP 2.0 certification.
The optional part tests functionality, which an external may not offer, or which an external scheduler substituted by more or less equivalent functionalities, which could be implemented without the use of the XBP interface.

For the optional test scenarios the following rules apply:

If the external job scheduler implemented a functionality as described in the test scenario (i.e. using the BAPI mentioned in the test scenario), then it has to undergo the test corresponding to that functionality. A failure has the same consequences as the failure of a mandatory test.

If the external job scheduler did not implement a functionality as described in the test scenario, the test will, of course, not be done. This has no consequences for the result of the whole certification test.

For each of the following test scenarios the candidate has to say before the test, if it will be part of the test or not.

8.1 Create an external job step

Description

Create a job with a step, which is the call of an external command (choose a suitable command from SM49)

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
A suitable external command has to exist or to be created via SM69 (SAP responsibility)

Realization

Use BAPI_XBP_JOB_ADD_EXT_STEP or BAPI_XBP_ADD_JOB_STEP..

Test

Check the job in SM37.

8.2 Read job data from the SAP system

Description

Read the data of a job, which has not been created by the external scheduler. Show the start condition of that job.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
A job has to be created before the test via SM36 (SAP responsibility)

Realization

Use BAPI_XBP_JOB_DEFINITION_GET or BAPI_XBP_JOB_READ..

Test

Check the values displayed by the external scheduler.

8.3 Create a job with status 'released'

Description

Create a job, which has the status 'released' and a start condition within its SAP job data.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.

Realization

Use BAPIs to create a job in status 'planned' first, then modify the job using BAPI_XBP_JOB_HEADER_MODIFY..

Test

Check the job in SM37.

8.4 Count Jobs with the same name

Description

For a given jobname, retrieve the number of jobs, which have that name.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.

Realization

Use BAPI_XBP_JOB_COUNT

Test

Compare the result displayed by the external scheduler with the number of TBTCO entries, which have that jobname.

8.5 Read the Spool list of a job

Description

Read and display the spool list of a job.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
Provide a job, which has a spool list. (Chose e.g job step RSWATCH0).

Realization

Use BAPI_XBP_JOB_SPOOLLIST_READ_20

Test

Compare the spool list displayed by the external scheduler with the spool list displayed by the SAP system..

8.6 Get information about the batch ressources

Description

Retrieve the available batch ressources for the current moment and for a certain date and time in the future.

Prerequisites

A connection to an SAP R/3 System is established.

An XBP session has to be established through the XMI interface.

Realization

Use BAPI_XBP_GET_CURR_BP_RESOURCES or
BAPI_XBP_GET_BP_RESRC_ON_DATE, resp.

Test

Compare the values displayed with the values in the SAP system, given by the operation modes.

8.7 Value help for external commands

Description

For the external commands show an arbitrary example of using the value help from within the scheduler. (With ‘external commands’ the commands, which are displayed in SM49, are meant.)

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.

Realization

Use BAPI_XBP_EXT_COMM_SEARCH.

Test

Compare the values displayed with the values in SM49.

8.8 Value help for print formats

Description

For a certain printer show the allowed print formats.

Prerequisites

A connection to an SAP R/3 System is established.
An XBP session has to be established through the XMI interface.
A printer has to be correctly configured in the SAP system (SAP responsibility)

Realization

Use BAPI_XBP_PRINT_FORMAT_SEARCH.

Test

Compare the values displayed with the values in the print parameters window (in the SM36 step editor)..

8.9 Error Handling

In error situations the BAPI functions return error information in an export structure to the caller. In the XBP 2.0 interface documentation, for each function all its possible errors are listed.
The external scheduler must be able to react to all errors that the BAPIS used in this test possibly return.
In order to test this, SAP will implement a mechanism in the SAP certification system, which prompts the tested BAPIs into returning all their possible error codes.

The BAPIs used in this test are:

(*mandatory part:*)
BAPI_XBP_JOB_OPEN
BAPI_XBP_JOB_ADD_ABAP_STEP (or BAPI_XBP_ADD_JOB_STEP)
BAPI_XBP_REPORT_SEARCH
BAPI_XBP_JOB_CLOSE
BAPI_XBP_JOB_START_ASAP
BAPI_XBP_JOB_START_IMMEDIATELY
BAPI_XBP_JOB_STATUS_GET
BAPI_XBP_JOB_ABORT
BAPI_XBP_JOB_DELETE
BAPI_XBP_EVENT_RAISE
BAPI_XBP_JOB_JOBLOG_READ
BAPI_XBP_JOB_COPY
BAPI_XBP_JOB_SELECT
BAPI_XBP_VARIANT_INFO_GET
BAPI_XBP_OUTPUT_DEVICE_SEARCH
BAPI_XBP_JOB_CHILDREN_GET
BAPI_XBP_JOBLIST_STATUS_GET
BAPI_XBP_GET_INTERCEPTED_JOBS
BAPI_XMI_LOGON
BAPI_XMI_LOGOFF
BAPI_XMI_SET_AUDITLEVEL

(*optional part:*)
BAPI_XBP_JOB_ADD_EXT_STEP
BAPI_XBP_JOB_DEFINITION_GET
BAPI_XBP_JOB_READ
BAPI_XBP_JOB_HEADER_MODIFY
BAPI_XBP_JOB_COUNT
BAPI_XBP_JOB_SPOOLLIST_READ_20
BAPI_XBP_GET_CURR_BP_RESOURCES
BAPI_XBP_GET_BP_RESRC_ON_DATE
BAPI_XBP_EXT_COMM_SEARCH
BAPI_XBP_PRINT_FORMAT_SEARCH

9 Appendix

This appendix contains the source code of two ABAP programs needed for some tests.

9.1 ZMAKECHILD

```
*&---------------------------------------------------------------------*
*& Report  ZMAKECHILD                                                  *
*&                                                                     *
*&---------------------------------------------------------------------*
*&                                                                     *
*&                                                                     *
*&---------------------------------------------------------------------*

REPORT  ZMAKECHILD  .

data: jobname  like tbtcjob-jobname.
data: jobcount like tbtcjob-jobcount.

data: report   like sy-repid.

data: start_day  like sy-datum.
data: start_time like sy-uzeit.

data: rel like BTCH0000-CHAR1.
data: cnt(2) type n.


start_day = sy-datum.
start_time = sy-uzeit + 60.

DO 4 times.

write sy-index to cnt.
concatenate 'CHILD_' cnt into jobname.

* job_open creates a job-header and returns the
* jobcount, which is needed in the following calls
* of job_submit and job_close.

CALL FUNCTION 'JOB_OPEN'
  EXPORTING
    JOBNAME                = jobname
    jobclass               = 'C'
 IMPORTING
   JOBCOUNT               =  jobcount
 EXCEPTIONS
   CANT_CREATE_JOB        = 1
   INVALID_JOB_DATA       = 2
   JOBNAME_MISSING        = 3
   OTHERS                 = 4
          .
IF SY-SUBRC <> 0.
   write: / 'Error in job_open, sy-subrc =', sy-subrc.
   exit.
ENDIF.


* job_submit must be called for each job step.
if sy-index = 1.
   report = 'BTCLOOP'.
else.
   report = 'RSWATCH0'.
```

```
endif.

CALL FUNCTION 'JOB_SUBMIT'
  EXPORTING
    AUTHCKNAM                  = sy-uname
    JOBCOUNT                   = jobcount
    JOBNAME                    = jobname
    REPORT                     = report
 EXCEPTIONS
   BAD_PRIPARAMS               = 1
   BAD_XPGFLAGS                = 2
   INVALID_JOBDATA             = 3
   JOBNAME_MISSING             = 4
   JOB_NOTEX                   = 5
   JOB_SUBMIT_FAILED           = 6
   LOCK_FAILED                 = 7
   PROGRAM_MISSING             = 8
   PROG_ABAP_AND_EXTPG_SET     = 9
   OTHERS                      = 10
          .

IF SY-SUBRC <> 0.
  write: / 'Error in Job_Submit, sy-subrc =', sy-subrc.
  exit.
ENDIF.


* job_close specifies the global job data, e.g. the start condition

CALL FUNCTION 'JOB_CLOSE'
  EXPORTING
    JOBCOUNT                 = jobcount
    JOBNAME                  = jobname
    SDLSTRTDT                = start_day
    SDLSTRTTM                = start_time
  IMPORTING
    JOB_WAS_RELEASED                  = rel
  EXCEPTIONS
    CANT_START_IMMEDIATE              = 1
    INVALID_STARTDATE                 = 2
    JOBNAME_MISSING                   = 3
    JOB_CLOSE_FAILED                  = 4
    JOB_NOSTEPS                       = 5
    JOB_NOTEX                         = 6
    LOCK_FAILED                       = 7
    OTHERS                            = 8
          .
IF SY-SUBRC <> 0.
   write: / 'Error in Job_Close, sy-subrc =', sy-subrc.
   exit.
ENDIF.

ENDDO.
```

9.2 ZVARI

```
*&---------------------------------------------------------------------*
*& Report  ZVARI
*&
*&---------------------------------------------------------------------*
*&
*&
*&---------------------------------------------------------------------*

REPORT  ZVARI.

tables tbtco.
```

```
parameters filename(50).
select-options date for tbtco-strtdate.

write: / 'filename =', filename.

skip.

loop at date.
   write: / 'date-low =', date-low.
   write: / 'date-high =', date-high.
endloop.
```

BC - XBP 7.00
Background Processing,
Job Scheduling System
SAP NW 7.00 (Version 3.0)
Test Catalogue

External Interface for
Background Processing

Copyright

Contents

Symbols

Symbol	Meaning
⚠	Warning
💬	Example
💡	Tip
🧭	Recommendation
SYN	Syntax

1 Overview

This document describes the technical certification for the XBP 3.0 interface of an external job scheduler.

⚠ Please, note:

XBP 3.0 is based on the XBP 2.0 interface. Therefore, the basis for XBP 3.0 certification is that your scheduler fulfills also the mandatory requirements of the XBP 2.0 test catalogue.

In addition, you should read the manual *XBP Interface Version 3.0*.

1.1 Certification Requirements

The following requirements should be met by external schedulers connected to the SAP system using XMI and XBP functions on the basis of RFC:

- The external job scheduling system must be able to fulfill the steps in the following test catalogue that the results can be reviewed.
- The external job scheduling system must also display the results of the tests in an appropriate way in its user interface.
- The external scheduler must not use other function calls to the SAP system than XMI and XBP functions and the functions for variant management (function group SVAR_RFC).
- The test contains only mandatory functions!

2 Structure of the test plan

The test catalogue is structured as follows:

Chapter 3	Contains the tests regarding the Criteria Manager.
Chapter 4	Contains the tests regarding the Event History.
Chapter 5	Contains the tests regarding application information functions.
Chapter 6	Contains the tests regarding spool and archiving functions.
Chapter 7	Contains the tests regarding variant functions.

3 Criteria Manager Functions

3.1 Defining Criteria for the Event History

Description

Create and activate a criteria profile for an event history criteria type using the following logical condition: Log all SAP events excluding SAP_END_OF_JOB.

Prerequisites

- A connection to an SAP system is established.
- An XBP session has to be established through the XMI interface.

Realization

The profile should be created using the function modules

- BAPI_CM_CRITERIA_SET
- BAPI_CM_PROFILES_GET
- BAPI_CM_PROFILE_ACTIVATE
- BAPI_CM_PROFILE_CREATE
- BAPI_CM_PROFILE_DEACTIVATE (optional)
- BAPI_CM_PROFILE_DELETE (optional)

Verification

1. In the SAP Criteria Manager (transaction SM64 or report CRITERIA_MANAGER), the Criteria Profile for the Event History appears and is active.
2. The profile details show the correct values.

3.2 Defining Criteria for Job Interception

Description

Create and activate a criteria profile for an interception criteria type using the following logical condition: Intercept all jobs having a job name starting with TEST excluding TEST_123 with job class B.

Prerequisites

- A connection to an SAP system is established.
- An XBP session has to be established through the XMI interface.

Realization

The profile should be created using the function modules

- BAPI_CM_CRITERIA_SET
- BAPI_CM_PROFILES_GET
- BAPI_CM_PROFILE_ACTIVATE
- BAPI_CM_PROFILE_CREATE

- BAPI_CM_PROFILE_DEACTIVATE (optional)
- BAPI_CM_PROFILE_DELETE (optional)

Verification

1. In the SAP Criteria Manager (report CRITERIA_MANAGER), the Criteria Profile for the job interception appears and is active.
2. The profile details show the correct values.

4 Event History Functions

4.1 SAP Event-Driven Scheduling with Control in an External Scheduler

Description

In this test scenario it shall be proven, that the external scheduler provides a mechanism to get aware of the fact, that an SAP batch event has been raised internally within the SAP system (e.g. via SM64).

Prerequisites

- A connection to an SAP system is established.
- An XBP session has to be established through the XMI interface.

Realization

For this test the event SAP_TEST with parameter 123 shall be used.

The external scheduler should show each occurrence of this event with parameter 123 (after it has been raised in the SAP system) in its user interface, or the external scheduler should perform a certain action after each occurrence of the event. This action can be e.g. starting a certain job in the SAP system, and this action would prove, that the external scheduler has become aware of the event.

For the value help of the event names, the function BAPI_XBP_EVENT_DEFINITIONS_GET has to be used.

The event history profile should be created using the function modules:

- BAPI_CM_CRITERIA_SET
- BAPI_CM_PROFILES_GET
- BAPI_CM_PROFILE_ACTIVATE
- BAPI_CM_PROFILE_CREATE
- BAPI_CM_PROFILE_DEACTIVATE
- BAPI_CM_PROFILE_DELETE (optional)

The synchronization should be done using the function modules:

- BAPI_XBP_BTC_EVTHISTORY_GET
- BAPI_XBP_BTC_EVTHIST_CONFIRM

Verification

1. Raise event SAP_TEST with parameter 123 (using transaction SM64 or program sapevt).
2. Check in the UI of the external scheduler or check, that the specified action (see above) has been performed.
3. Repeat points 1 and 2.

5 Application Information Functions

5.1 Reading Application Log Content and Application Return Code

Description

Schedule a job that executes a program which is creates an application return code and writes entries into the application log. The application return code and the content of the log can be read. Use BTCTESTN as a test program.

Prerequisites

- A connection to an SAP system is established.
- An XBP session has to be established through the XMI interface.

Realization

The job should be created using the function modules:

- BAPI_XBP_JOB_OPEN
- BAPI_XBP_JOB_ADD_ABAP_STEP (or BAPI_XBP_ADD_JOB_STEP); as ABAP step use program BTCTESTN
- BAPI_XBP_JOB_CLOSE

The application information can be read with the function modules:

- BAPI_XBP_APPL_INFO_GET
- BAPI_XBP_APPL_LOG_CONTENT_GET

Verification

1. Create a job with an ABAP step that creates an application return code and writes entries in the application log, for example with program BTCTESTN.
2. Read the application return information with BAPI_XBP_APPL_INFO_GET. This function module returns the application return code and, if available, a table of handles of the application log.
3. These handles can be used to read the content of the application log with function module BAPI_XBP_APPL_LOG_CONTENT_GET.

6 Spool and Archiving Functions

6.1 Reading the Content of a Spool Request Specified by Number

Description

In this scenario, the content of a spool request that has been specified by its number is to be retrieved via the XBP interface.

Prerequisites

- A connection to an SAP system is established.
- The corrections of note 1171295 have to be in the test system.
- The audit level is set to the value 2, so that the use of the required functions can be verified in the XMI log (RZ15).

Realization

1. Use BAPI_XBP_READ_SELSCREEN to read the parameters of report RSPO0015.
2. Use BAPI_XBP_JOB_ADD_ABAP_STEP to create a job with a temporary variant of RSPO0015 with RUNS = 3 and PRINTER = LP01.
3. Use BAPI_XBP_JOB_DEFINITION_GET to retrieve the numbers of the spool requests created by the job.
4. Use BAPI_XBP_JOB_READ_SINGLE_SPOOL to read the content of the second spool request.

Verification

- Check XMI log via RZ15.
- Use UI of scheduling tool to display the contents of the spool list.

6.2 Passing an E-Mail Address as Recipient for the Spool List of a Job

Description

In this scenario, an e-mail address is to be passed as the recipient for the spool list created by a background job.

Prerequisites

- A connection to an SAP system is established.
- The corrections of note 1171295 have to be in the test system.
- The audit level is set to the value 2, so that the use of the required functions can be verified in the XMI log (RZ15).

Realization

Create a job and add an e-mail address as recipient to the job via function BAPI_XBP_JOB_CLOSE.

Verification

- Check XMI log via RZ15.
- Check with transaction SOST that a mail request has been created for the specified mail address.

7 Variant Functions

In the following scenarios, functions of the function groups SXBP_VAR and SXBP are to be used.

For the scenarios below, use the selection screen and variants of the following program:

```
REPORT  ZVARI_1.

TABLES: tbtco, tbtcp.

PARAMETERS user LIKE tbtco-sdluname default 'scheduler'.

PARAMETERS date LIKE sy-datum default '19720101'.

PARAMETERS chbox AS CHECKBOX.

PARAMETERS radio11 RADIOBUTTON GROUP tst1.
PARAMETERS radio21 RADIOBUTTON GROUP tst1  DEFAULT 'X'.
PARAMETERS radio31 RADIOBUTTON GROUP tst1.

PARAMETERS radio12 RADIOBUTTON GROUP tst2.
PARAMETERS radio22 RADIOBUTTON GROUP tst2  DEFAULT 'X'.
PARAMETERS radio32 RADIOBUTTON GROUP tst2.

SELECT-OPTIONS sel_op1 FOR tbtco-jobname no-extension.
SELECT-OPTIONS sel_op2 FOR tbtco-jobcount no-display.

select-options sel_prog for tbtcp-xpgparams.
parameters par_prog like tbtcp-xpgparams.
```

7.1 Reading the Selection Screen of a Program and Creating a Variant

Description

A variant for the program ZVARI_1 (see above) is to be created. In order to do that, the scheduler needs to read the selection screen of the program first.

Prerequisites

- A connection to an SAP system is established.
- The corrections of note 1171295 have to be in the test system.
- The audit level is set to the value 2, so that the use of the required functions can be verified in the XMI log (RZ15).

Realization

Use BAPI_XBP_READ_SELSCREEN to read the selection screen.

Use BAPI_XBP_VARIANT_CREATE to create the variant.

Verification

- Check XMI log via RZ15.
- Check in SE38, if a new variant has in fact been created.

7.2 Changing the Variant of a Program

Description

Change the variant created above. Change at least two fields.

Prerequisites

- A connection to an SAP system is established.
- The corrections of note 1171295 have to be in the test system.

Realization

Use BAPI_XBP_VARIANT_CHANGE to change the variant.

Verification

- Check XMI log via RZ15.
- Check in SE38, if the values of the variant have been changed.

7.3 Reading the Selection Screen of a Program and Creating a Job with a Temporary Variant

Description

A background job with job step ZVARI_1 is to be created. Selection screen data for ZVARI_1 has to be specified, so that a temporary variant is created. In order to do that, the scheduler needs to read the selection screen of the program first.

Prerequisites

- A connection to an SAP system is established.
- The corrections of note 1171295 have to be in the test system.
- The audit level is set to the value 2, so that the use of the required functions can be verified in the XMI log (RZ15).

Realization

Use BAPI_XBP_READ_SELSCREEN to read the selection screen.
Use BAPI_XBP_JOB_ADD_ABAP_STEP to create the job step and the temporary variant.

Verification

- Check XMI log via RZ15.
- Check in the step data of SM37, if a new temporary variant has in fact been created.