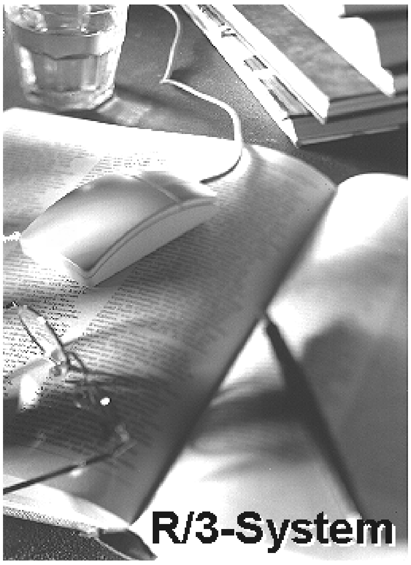
BC – XBP 6.10
Background Processing,
Job Scheduling System
WAS 6.10 (Version 2.0)
Documentation



External Interface for Background Processing

Copyright

Contents

Release Information for XBP Version 2.0 - 2002

XBP 2.0 is an enhancement of XBP 1.0. This means that XBP 1.0 is a subset of XBP 2.0. Therefore, customers who use XBP 1.0 can install XBP 2.0 without any change of functionality in their current applications. XBP 2.0 will be available for SAP Basis releases 4.6B and higher. For the releases that are already on the market, XBP 2.0 will be delivered by Support Packages.

The following lists give you an overview of the differences between XBP 1.0 and XBP 2.0. All these functions are described in detail later in this document.

1. Parent/child functionality (new):

Function module	Feature
BAPI_XBP_JOB_CHILDREN_GET	Getting all children created by a job.
BAPI_XBP_JOB_PARENT_CHILD_INFO	Getting child/parent relation of a job.
BAPI_XBP_SPECIAL_CONFIRM_JOB	Setting special confirmation types for jobs, such as for child jobs.

2. Intercept functionality (new)

Function module	Feature
BAPI_XBP_GET_INTERCEPTED_JOBS	Retrieving jobs with the new status 'Intercepted'.
BAPI_XBP_SPECIAL_CONFIRM_JOB	Setting special confirmation types for jobs, such as intercepted jobs.
BAPI_XBP_MODIFY_CRITERIA_TABLE	Modifying the criteria table.

3. New basic functionalities (BAPIs):

Function module	Feature
BAPI_XBP_CONFIRM_JOB	Confirming any kind of job.
BAPI_XBP_EVENT_RAISE	Triggering an event from outside.
BAPI_XBP_JOBLIST_STATUS_GET	Determining the status of a list of jobs by reading R/3 information on all jobs. Contains also information about parent/child relations.
BAPI_XBP_JOB_COPY	Copying jobs.
BAPI XBP_JOB_HEADER_MODIFY	Changing global data of a job, such as the start condition.
BAPI_XBP_ADD_JOB_STEP	Adding and inserting a step to a job via XMI.
BAPI_XBP_MODIFY_JOB_STEP	Changing and deleting a job step via XMI.
BAPI_XBP_JOB_READ	Obtaining key job parameters from job headers and job steps.
BAPI_XBP_JOB_SPOOLLIST_READ_20	Replacing the existing function modules • BAPI_XBP_SPOOLLIST_READ

	• BAPI_XBP_SPOOLLST_READ_RW
BAPI_XBP_REPORT_SEARCH	Returning a list of ABAP reports available in the current system, whose names match a certain wildcard.
BAPI_XBP_EXT_COMM_SEARCH	Returning a list of external commands available in the current system, whose names match a certain wildcard.
BAPI_XBP_OUTPUT_DEVICE_SEARCH	Returning a list of output devices available in the current system, whose names match a certain wildcard.
BAPI_XBP_PRINT_FORMAT_SEARCH	Returning a list of print formats available for a certain printer. The names of print formats match certain wildcards.
BAPI_XBP_NEW_FUNC_CHECK	Reading and changing the status of interception and parent/child functionality.

4. Enhanced basic functionality (BAPIs):

Function module	Enhancement/New Feature
BAPI_XBP_JOB_OPEN	Parameter for choosing the job class.
BAPI_XBP_JOB_CLOSE	Parameter for describing the spool list recipient in internal format.
BAPI_XBP_JOB_STATUS_GET	Parameter for information about child/parent relations.
BAPI_XBP_JOB_ADD_ABAP_STEP	Structures for the specification of all print and archive parameters.
BAPI_XBP_JOB_ABAP_STEP_MODIFY	Structures for the specification of all print and archive parameters.
BAPI_XBP_JOB_START_IMMEDIATELY BAPI_XBP_JOB_START_ASAP	Can now also start jobs with status ‘intercepted’ and ‘released’.
BAPI_XBP_JOB_SELECT	Can now select several types of jobs: • all jobs • jobs without general confirmation • jobs without any type of confirmation

However, the new features interception and parent/child relation may not be useful for all users. Therefore, we want to give the user the option of switching off these functionalities to avoid unnecessary data storage in the system and unnecessary internal subroutine calls. Functions that can be switched on are switched off initially. With XBP 2.0, you can switch the new functions on and off with the ABAP program **INITXBP2.**

Remember to run INITXBP2 before using the job interception and parent/child functions.

Symbols

Symbol	Meaning
⚠	Warning
💬	Example
💡	Tip
🧭	Recommendation
<syn>	Syntax

1. Introduction

This document deals with the connection of an external job management system, often called an *external scheduler*, to the SAP R/3 system. By external job management, we mean software which allows jobs to be scheduled, run, and monitored from outside the SAP R/3 system.

For this purpose, SAP has defined an open generic interface. This is called XBP, which stands for eXternal interface for job Background Processing. XBP is one of a range of open interfaces which SAP intends to make available in the future for system management tasks. SAP R/3's internal CCMS (Computing Center Management System), with this range of interfaces, offers support to software manufacturers by allowing integration with existing system administration tools.

2. This Document - An Overview

We assume that the reader already has a certain degree of knowledge of the SAP R/3 system. Also, he/she should be familiar with terms like application server, dispatcher, scheduler, and work process in relation to SAP R/3 Basis. Additionally, he/she should be familiar with the Remote Function Call (RFC).

This overview helps you to find the topics relevant for you:

Chapter	Contents
3	General description of the external interface.
4	Short introduction to SAP R/3 background processing.
5	Description of the external interface concept.
6	Actual description of the XBP interface; the motivation behind it, its technical background and a functional overview.
7	Technical description of the function modules. This is provided to help with the implementation of external agents.

3. The Function of External Interfaces

The motivation for the development of external interfaces arose from the desire to let SAP installations - especially large ones - be overseen by other software vendors' tools.

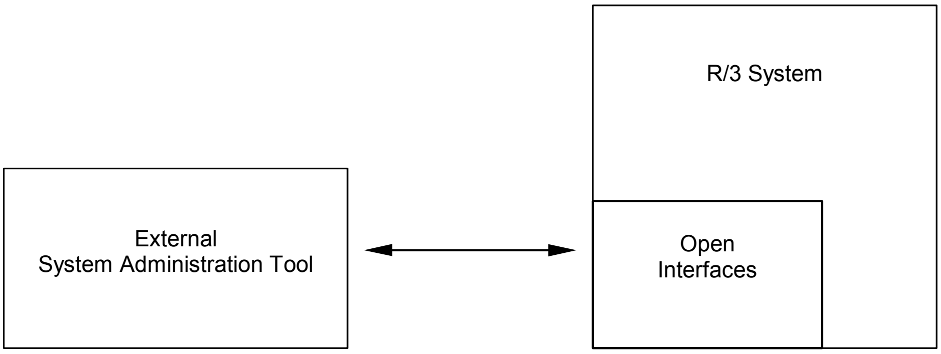


Fig. 3.1: Location of external interfaces

External interfaces allow you to integrate SAP R/3 simply and seamlessly into both local administrative tools and business-wide system management infrastructures. This integration should not and cannot completely replace the use of CCMS. Complex and security-critical system control tasks will always need to be carried out by CCMS. Furthermore, CCMS basic functionality is expected and required by SAP R/3 applications. To explain by example: even if you use an external scheduling tool for background jobs, the internal job system in SAP R/3 is still required to carry out the background jobs generated by applications such as SAP R/3 archiving or the Workbench.

The aim of the integration is to allow the customer a homogenous information infrastructure. The aim of the interface is to facilitate the flow of information between the SAP R/3 system and external tools.

In summing up, we could say that external tools provide additional flexibility to complement the existing basic functionality of CCMS.

4. A Short Introduction to SAP R/3 Background Processing

First, we need to explain why background processing has a place in a dialog-oriented standard application. What is a 'job' within the SAP R/3 system, and how does the system carry it out?

We intend to demonstrate the concept using an example. Afterwards, we will introduce the architecture of the background processing system.

4.1 Motivation

SAP R/3 is, above all, an interactive system. In other words, the vast majority of tasks are carried out in dialog with the user. However, there are also good reasons for the inclusion of a background processing system in SAP R/3.

Besides the tasks carried out in dialog, there are numerous tasks processing large amounts of data and requiring lots of performance that do not need user interaction. With the help of the background processing system, such tasks are normally scheduled for times when no users are working in the system (nights, weekends), in order to avoid resource conflicts with the dialog users.
At the scheduled time, these tasks are started by the background processing system and executed without user interaction, even without a connection to any frontend

server.
This mechanism is especially useful for tasks that have to be carried out periodically, for example each week or each month. In the background processing system, these tasks – including the period - have to be specified once only. No further action is required from the user with respect to regular execution.

4.2 Concept

A task executed by the background processing system is called a 'background job', 'batch job', or simply 'job'. Technically speaking, a job in the SAP background processing system executes one or more ABAP programs or calls on the OS level, which are referred to as the job steps. The job steps are executed sequentially in the order of their definition.

So, roughly speaking, one can say that defining a job consists of defining the job steps and the job header (start conditions, target server, and other data).

Jobs are identified by their name (Ex.: `PAYROLL_RUN`). However, since these names are not unique if the same application job is repeated, jobs also have job numbers, which ensure that they have a unique identification

In its life cycle, a job always has exactly one of the following statuses :

Status	Description
Scheduled	The job steps but no start conditions have been defined.
Released	The job definition is complete and the job is waiting for the start conditions to be fulfilled.
Ready	The job is in its start phase, which normally takes only a fraction of a second. This status is of no interest for the end user. However, it is useful for error analysis by SAP support.
Active	The job is being executed.
Finished	The job is finished without errors.
Cancelled	The job execution was terminated due to an error.

Additionally, from XBP 2.0 on, it is possible to *intercept* jobs. 'Interception' means that the jobs are not started at the moment when their start conditions are fullfilled, but deactivated and restarted later. However, note that *intercepted* is not really a new status in the SAP background processing system (see chapter 4.2.5 'Intercepting jobs').

Figure 4.1 shows the chain of these statuses:

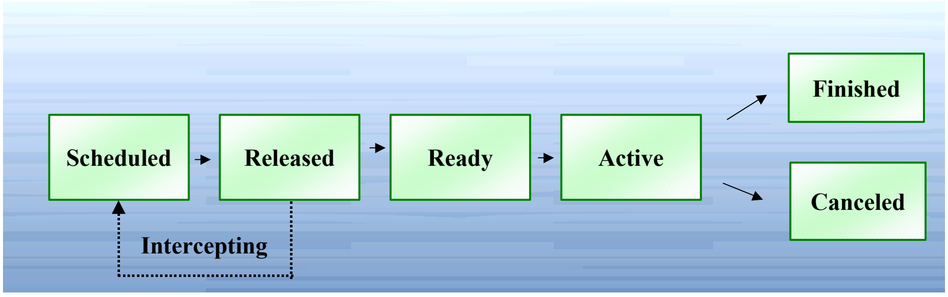


Fig. 4.1: Status chain for jobs.

In the next sections, we follow the life cycle of a job systematically. Figure 4.2 shows an example of the activities which can cause a job to change status. A job ‘scheduled’ by a program is ‘released’ by a dialog. It then proceeds through the ‘ready’ and ‘active’ statuses with help from SAP R/3 system programs. If everything has gone according to plan, the job status moves on to ‘finished’. A program error will lead to a final status of ‘canceled’. XBP 2.0 now offers you the possibility to intercept jobs and to reschedule and restart them later. This feature allows you to prioritize jobs dynamically as described later in detail.

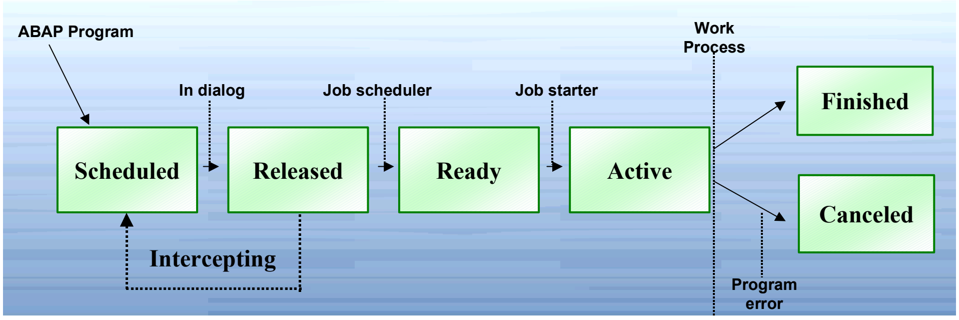


Fig. 4.2: Status chain for jobs with possible causes of status change.

You can see the current status of a particular job in the SAP R/3 system using the job overview (transaction SM37).

4.2.1 Creating Jobs

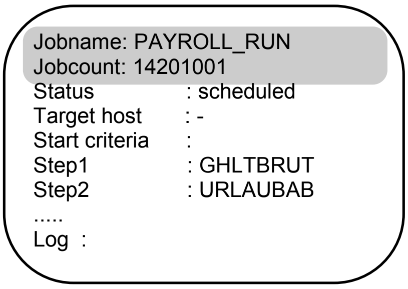
A job definition consists of a list of steps and administrative information known as job header. The job header contains the start conditions, job class, and target server.

If a job is created without start conditions, the job receives the status ‘scheduled’. If a job is defined completely, the job receives the status ‘released’.

Within an SAP system, there are two ways of creating a job:

1. In a dialog (Transaction SM36, or *Tools* ➔ *Administration* ➔ *Jobs* ➔ *Define Job*) you enter the job name, job class and, if necessary, a target machine. Next, you enter a list of steps and a **start time**, if required.
2. New jobs are created from ABAP programs using the JOB_OPEN, JOB_SUBMIT and JOB_CLOSE function modules, which are part of the Batch API. You have the same parameters and degree of freedom here as in the dialog.

In each of these cases, the job number is created by the system itself to ensure that the job has a unique identification. Figure 4.3 shows the specification of a newly created job, with its name, number, status and steps.



```
Jobname: PAYROLL_RUN
Jobcount: 14201001
Status          : scheduled
Target host     : -
Start criteria  :
Step1           : GHLTBRUT
Step2           : URLAUBAB
.....
Log  :
```

Fig. 4.3: A created job from the system's point of view (greatly simplified)

If **no** target machine is specified for the job, the system decides itself on which application server the job is to be carried out. This is the normal procedure recommended by SAP, because the system can then carry out its own load balancing. You should only specify the target machine if that server has particular resources necessary for the job processing.

4.2.2 Releasing Jobs

A job is released (set to status 'released') as soon as it has been completely created. This means that the job has to have at least one step and a start condition. The SAP background system offers several types of start conditions, which can be specified by the user.

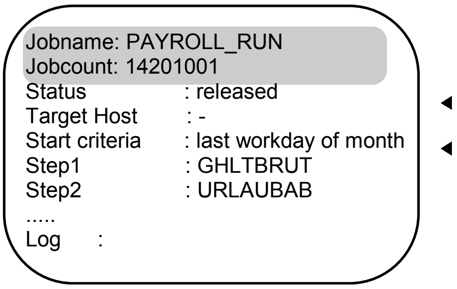
Note that different start conditions cannot be combined by 'and' or 'or'. Exactly one of the following start conditions has to be specified for each job:

Start condition	Description
Immediate	A job is executed as quickly as possible.
Date/ time	A job is started when its start date and time are reached.
Event	A job is started when a particular event is triggered in the system or by a program at operating system level. E.g. when data for processing is imported to a server by file transfer.
Preceding job	A job is started once a particular preceding job has run.
Change of operation mode	A job starts once the SAP R/3 system has switched into a particular operation mode.
Start on working day	A job is started when a particular working day of the month is reached, for example the last working day of a month.

Once you have released a job by setting a start condition, the **Job scheduler** in the SAP R/3 system becomes responsible for the future progress of the job.

For some of the above start criteria you can also specify that the job should recur. Thus you can ensure that a new job with the same name, but a different job number for each repetition, is released, for example, each time a particular event occurs, or every day from today. A job with such a start condition is called a periodic job.

In selecting start criteria, you can use various calendars, either pre-defined or user-defined. This allows you a still greater degree of freedom, by allowing you to specify deviations in the frequency of periodic jobs (e.g. carry out periodically, but not on Public Holidays).



Jobname: PAYROLL_RUN
Jobcount: 14201001

Status	: released
Target Host	: -
Start criteria	: last workday of month
Step1	: GHLTBRUT
Step2	: URLAUBAB
.....	
Log :	

Fig. 4.4: A released job from the point of view of the system (greatly simplified).

Figure 4.4 shows that as well as the status changing, the start criteria have been laid down in the job description.

4.2.3 Starting Jobs (Ready and Active)

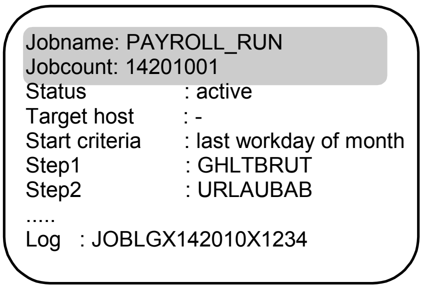
Within the system, the **Job scheduler** surveys the start conditions of the released jobs. For each job, the scheduler monitors whether its start condition is fullfilled.

Note that the term ‘job scheduler’ is somewhat simplified. In fact, there are different kinds of job scheduler (time-based and event-based). And, in a system with several servers the number of job schedulers varies. However, the task is the same for all. They monitor the start conditions of the released jobs and start the released jobs once their start conditions are fulfilled. Therefore, in this documentation ‘job scheduler’ is used without differentiation.

In the SAP system, there is a special type of work process, the background work process. This work process is reserved for executing jobs. Of course, the job scheduler can only start a job if there is a free background work process.

Technically, a job start functions as follows:

When the start condition of the job is fulfilled, the job scheduler checks if there is a free background work process available. If this is the case, the job scheduler sets the job to the status ‘ready’ and sends a job start message, which is assigned to a free background work process by the system. The background work process finally sets the job to the status ‘active’ and starts executing it. The component of the background work process responsible for this is called the ‘job starter’.



Jobname: PAYROLL_RUN
Jobcount: 14201001

Status	: active
Target host	: -
Start criteria	: last workday of month
Step1	: GHLTBRUT
Step2	: URLAUBAB
.....	
Log	: JOBLGX142010X1234

Fig. 4.5: An active job from the point of view of the system (greatly simplified).

If there is no free background work process available at the time when the start condition is fulfilled, the job scheduler monitors the job until there is a free work process available. Then the job scheduler sends the job start message as described above.

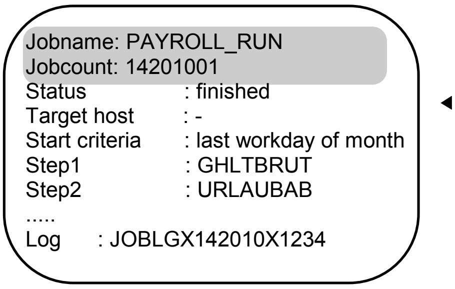
The status ‘ready’ is a technical status and of no interest for the user. Normally, a job has this status only for a fraction of a second and the user does not even see this status.

As figure 4.5 shows, the job definition has now gained a link to a **job log**.

4.2.4 Ending a Job (Canceled or Finished)

A job logs its steps in a job log, which is visible to the user. If everything runs according to plan and no errors occur, the job acquires the status ‘finished’ . This can be seen both in the job log and the dialog in Transaction SM37.

If an ABAP error or any other interruption occurs at runtime, an error message is recorded in the job log, and the job is terminated.



```
Jobname: PAYROLL_RUN
Jobcount: 14201001
Status          : finished
Target host     : -
Start criteria  : last workday of month
Step1           : GHLTBRUT
Step2           : URLAUBAB
.....
Log     : JOBLGX142010X1234
```

Fig. 4.6: A finished job from the point of view of the system (greatly simplified).

Figure 4.6 shows the final view of the job. The job status is ‘finished’. Further information can be seen in the log.

4.2.5 Intercepting Jobs

Interception is a new feature of the R/3 background processing system, which is fully usable by XBP 2.0.

Intercepting jobs means that at the moment when the start conditions of the jobs are fulfilled, the jobs are set back to the status ‘scheduled’ and receive a special attribute. By calling a new XBP function, the external scheduler can receive a list of all intercepted jobs. Although ‘intercepted’ is not really a new status (as described above, an intercepted job has the status ‘scheduled’ and a specific attribute), it is presented to XBP 2.0 as a new status. If logged on with version 2.0, the function BAPI_XBP_STATUS_GET will return an I for an intercepted job, whereas it will return P, if logged on with version 1.0.

Of course, it is not intended to subject jobs to interception in general. The user can define criteria in the new table TBCICPT1 (client, job name, job creator including wildcards), and only the jobs that match these intercept criteria are intercepted. For instance, a table entry   (100, babu* , *) means that all jobs created in client 100 by users beginning with babu are intercepted. These table entries can be added, changed and deleted with a new XBP function.

Note that the decision as to whether or not a job has to be intercepted, is made by the SAP system at the moment, when the start condition is fulfilled. That means that at that moment the job data are matched against the table contents. A job is intercepted if it matches the intercept criteria at that moment.

Example for the use of interception:

The administrator might want to intercept all jobs of certain users or with certain job names on weekends when long-running and time critical batch jobs are executed. In this case interception provides dynamic job prioritization.

However, the new intercept feature may not be used by all customers. Therefore, this functionality can be globally switched off completely. This has the advantage that internal calls of subroutines in the runtime system are avoided.

Functionalities that can be switched on are switched off initially. With XBP 2.0, you can switch the new functions globally on and off with the ABAP program **INITXBP2.**

Remember to run INITXBP2 before using the job interception function.

More information about the technical implementation of the status ‘intercepted’

All XBP 2.0 functions that return status information of a job will return an ‘I’(= Intercepted) for intercepted jobs. But that does not mean that there will be the value ‘I’ in the R/3 field TBTCO-status. The field STATUS in the TBTCO table does not contain an ‘I’, but a ‘P’. Only the additional attribute ‘intercepted’ in the table TBTCCNTXT makes it clear that we are dealing with an intercepted job.

Attribute ‘intercepted-confirmed’

In order to know if there are any intercepted jobs, the external scheduler calls a function at short intervals (BAPI_XBP_GET_INTERCEPTED_JOBS). To prevent this function from returning the same intercepted jobs again and again, the scheduler can confirm a list of intercepted jobs. Confirmation means that the scheduler informs the R/3 system that it already knows these intercepted jobs, and that a subsequent call of BAPI_XBP_GET_INTERCEPTED_JOBS does not have to return these jobs again.

The confirmation of a list of intercepted jobs is done by calling of BAPI_XBP_SPECIAL_CONFIRM_JOB, which sets these jobs to status ‘intercepted-confirmed’. However, in some situations (such as after a breakdown) it might be useful to get a list of all intercepted jobs (including the confirmed ones). For this purpose, the function BAPI_XBP_GET_INTERCEPTED_JOBS has a special indicator. You can find detailed information in the technical description of these function modules later in this document.

Treatment of periodic intercepted jobs

If a job that matches the interception criteria is periodic, the R/3 rescheduling mechanism applies.

This means that if a periodic job is set to status ‘intercepted’, the R/3 system creates the successor immediately after intercepting the predecessor. The successor, of course, has the status ‘released’. If the start condition of the successor is fulfilled, it will be set to ‘intercepted’, and its successor is created, and so on.

New or enhanced function modules for the status ‘intercepted’:

- BAPI_XBP_MODIFY_CRITERIA_TABLE: New function module for adding and modifying the table with the intercept criteria
- BAPI_XBP_GET_INTERCEPTED_JOBS: New function module for returning jobs with status ‘intercepted’
- BAPI_XBP_SPECIAL_CONFIRM_JOB: New function module for setting special types of confirmation for a list of jobs, such as for intercepted jobs

- BAPI_XBP_NEW_FUNC_CHECK: New function module for reading and changing the status of the intercept function and the parent/child functionality
- BAPI_XBP_JOB_START_IMMEDIATELY/BAPI_XBP_JOB_START_ASAP: Enhanced XBP 1.0 features; additional functions to start intercepted jobs and released jobs
- BAPI_XBP_JOBLIST_STATUS_GET: Enhanced XBP 1.0 feature; returns now also the status эinterceptэ

4.2.6 Parent/Child Functionality

In general a business process that is carried out by a job or rather a collection of jobs, does not only consist of static jobs, which are known in advance and shown right away in SM37, but also of jobs that are created at runtime by the static jobs, such as to dynamically distribute workload. A job that is released by another job is called a child job, and the releasing job is called a parent job.

For a job scheduling system, it is important to know about the existence and current status of the child jobs of a certain parent job, because in the internal logic of many applications a parent job is considered as ‘finished’ only if the parent job itself **and** its child jobs are finished.

Up to now there has been no proper way for an external scheduler to find out whether a job has child jobs or not. Now the SAP background processing system stores the parent/child data of jobs automatically and offers functions to access these data.

The new functionality can be fully used by XBP 2.0. This means XBP 2.0 also offers functions to access the parent/child data of jobs.

New or enhanced function modules:

- BAPI_XBP_JOB_CHILDREN_GET: New function module returning a list of all child jobs of a certain job.
- BAPI_XBP_JOB_PARENT_CHILD_INFO: New function module returning information as to whether the job is a child or a parent and some other useful information.
- BAPI_XBP_JOB_STATUS_GET: Enhanced function returning now also the parent/child information for a single job
- BAPI_XBP_SPECIAL_CONFIRM_JOB: As is the case with intercepted jobs, it is also possible to confirm child jobs. This has the effect that confirmed jobs are not returned anymore by subsequent calls of BAPI_XBP_JOB_CHILDREN_GET.
- BAPI_XBP_JOBLIST_STATUS_GET: New function module receiving a list of jobs, for example the list of all child jobs of a certain job, and completing the list by the addition of status information for each job and a flag that indicates, if a job has child jobs.

However, the new parent/child feature may not be used by all customers. If a customer does not use this feature, there is not need to write the parent/child information into the database. So this feature can be globally switched off with the program **INITXBP2.**

Remember to run INITXBP2 before using the parent/child functionalitiy.

Attribute ‘child job-confirmed’:

In order to find out, if a certain job has child jobs, the external scheduler calls a function at short intervals (BAPI_XBP_JOB_CHILDREN_GET). To prevent this function from returning the same child jobs again and again, the scheduler can confirm a list of child jobs. Confirmation means that the scheduler informs the R/3 system that it already knows these child jobs, and that a subsequent call of BAPI_XBP_JOB_CHILDREN_GET does not have to return these jobs again.

The confirmation of a list of child jobs is done by calling BAPI_XBP_SPECIAL_CONFIRM_JOB, which sets these jobs to status ‘child-confirmed’ (for more details see the chapter below). However, in some situations (such as after a breakdown) it might be useful to get a list of all child jobs (including the confirmed ones). For this purpose, the function BAPI_XBP_JOB_CHILDREN_GET has a special indicator. You can find detailed information in the technical description of these function modules later in this document.

4.2.7 Confirming Jobs

The concept of confirming jobs is even wider than already described in the context of interception and parent/child jobs. There are three XBP 2.0 functions for job selection:

- BAPI_XBP_JOB_SELECT for general job selection
- BAPI_XBP_JOB_CHILDREN_GET for child job selection
- BAPI_XBP_GET_INTERCEPTED_JOBS for the selection of intercepted jobs

These functions are normally called at intervals by the external job scheduler and return a list of jobs. If you do not want the system to return the same jobs over and over again, you can confirm them. Confirmation means that the scheduler informs the R/3 system that it already knows these jobs, and that a subsequent call of the selection function module does not have to return these jobs again.

There are two types of confirmation:

- **General:** With general confirmation, the job scheduler confirms that it knows a job in general. Jobs are confirmed generally with BAPI_XBP_CONFIRM_JOB. When you use any of the three selection functions, the generally confirmed jobs are not returned if the corresponding indicator (parameter name: SELECTION) is set appropriately.
- **Special:** With special confirmation, the job scheduler confirms that it knows that a job has certain characteristics, for example if a job is an intercepted job or a child job.

  Child jobs and intercepted jobs are confirmed with BAPI_XBP_SPECIAL_CONFIRM_JOB. When you use BAPI_XBP_JOB_CHILDREN_GET or BAPI_XBP_GET_INTERCEPTED_JOBS the specially confirmed jobs are not returned again, if the corresponding indicator is set appropriately.

  > However, in some situations (such as after a breakdown) it might be useful to get a list of all intercepted or child jobs (including the confirmed ones). This function has a special indicator for this purpose.

Chapter 7 contains details of these function modules.

4.3 Architecture of the SAP R/3 Job Scheduling System

The essential components of the SAP R/3 background processing are the database tables which contain the job administration data and steps, the job scheduler, the job starter, the job log, and the spool subsystem.

Within this document we can only present a rough sketch of how these elements work together. The aim is, however, to provide a simple model of how the system works.

4.3.1 Job Administration in the Database

All essential job administration information and the job steps themselves are stored in the database. This ensures the consistency and security of the relevant information.

The most important of the database tables is the job data table. This contains entries necessary for job administration: job name, job number, target host, desired start time, job log name and much more. The step list for a job is not contained in this table.

The step list table contains a number of ABAP and operating system level programs (‘external programs’) for each job. Figures 4.3 - 4.6 show data from both tables in one view.

An event table lists all events defined in SAP R/3, along with the jobs that they trigger when the particular event occurs in the system.

4.3.2 The Job scheduler

In actual fact, the job scheduler consists of two schedulers - one for event based and one for time based jobs. According to the start criteria for a particular job, one or the other of these schedulers assumes responsibility for passing it on. For the sake of this overview, we will make no further distinction between the two schedulers.

The job scheduler for time controlled jobs is started regularly on all the application servers in the SAP R/3 system which carry out background processing and have background work processes for that purpose. You can set the interval at which it is started in the profile parameter rdisp/btctime for each application server. The event driven job scheduler is started on the application server on which an event is triggered.

When the scheduler begins its task, it selects jobs from the database (job administration data) which have reached their start date or whose triggering event has taken place. The scheduler also takes into account whether any background work processes are free. If it comes across free processes of the right type, it tries to send as many jobs as possible to these processes, always bearing in mind the priority of the jobs and maintaining a reserve of background work processes for important class A jobs. Jobs which cannot be processed are left untouched. Jobs, on the other hand, which are sent to the background work processes are marked as such in the database (‘ready’ status) and, after a very short delay, are taken up by the chosen work process and processed completely (without interruption). If the job scheduler starts a periodic job, it immediately reschedules the successor.

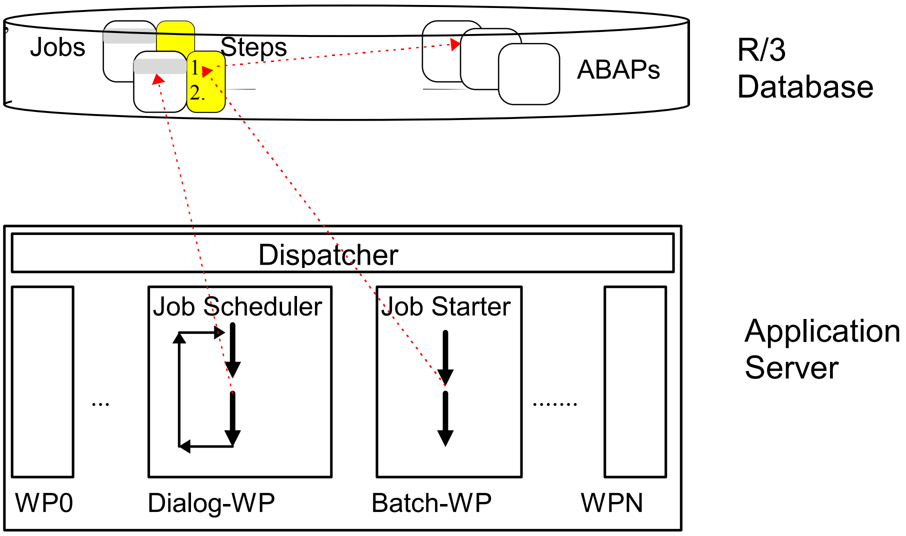


Fig. 4.7: System architecture of SAP R/3 background processing (simplified).

4.3.3 The Job Starter and the Job Runtime Environment

The job starter is a compact program which is processed in the background work process and finds a step list in the database. The step list is then processed step by step. If an external program is involved, a UNIX or Windows NT command is sent to the operating system. If the step consists of an ABAP program, this is carried out in the SAP system.

4.3.4 The Job Log

Generally speaking, the job log is not stored in the database. It is a TemSe object, and usually a file at operating system level. The TemSe objects are **tem**porary and **se**quential SAP R/3 objects, which can be stored system-wide in the database, but are normally kept directly in the file system. In any case, the name of the TemSe object - normally the filename - is stored in the job administration information.

If a job log is requested for a job, the TemSe object is displayed. The user cannot tell from the dialog whether this is a database object or whether it has been saved in a different way.

4.3.5 Job Output

Most ABAP or external programs generate output while they are running. Possible output includes error messages, messages about a program's progress, or lists resulting from a report.

These outputs and messages are not immediately visible, since the programs are not running in dialog, but in the background. To avoid this output being lost, it is saved so that it can be looked at later. The output lists of a job are looked after by the spool output management (TemSe). All messages are recorded in the job log. The job log contains information on all job steps.

The step list stores information on the location of the output of each job. The reason for this is that each step can create its own output.

5. The External Interface Concept

The XBP interface described in this document is part of an interface package for external system management tools. On the SAP R/3 side, the interfaces are constructed through a pool of function modules. External management systems are able to call these using RFC (Remote Function Call). You can find example calls in the documentation on the interface itself. A short introduction to RFC forms part of the technical basics at the end of this chapter.

In the following section, we are working on the assumption that the external system management tool is represented to the SAP R/3 system by an agent. This agent is the communication partner of the function modules.

5.1 Range of Interfaces

The range of interfaces, under the name XM (eXternal SAP R/3 Management), consists at present of the following individual components:

- **XBP** eXternal Interface for Background Processing
- **XBR** eXternal Interface for Backup & Recovery (so far called: BRI or Backint)
- **XMB** eXternal Interface for Monitoring Basics
- **XMI** eXternal Monitor Interface
- **XOM** eXternal Interface for Output Management

In the following table you can see the abbreviated name of the interface (and its current version), its implementation technique, the SAP R/3 Release with which it will be delivered and a short description. The certification column shows whether and when a concrete interface can be certified. Only the abbreviations from the first column will be used in technical descriptions.

Abbr./ Version	**Technique**	**R/3 Rel.**	**Short Description**	**Certification**
XBP **2.0**	RFC	4.6 B	Job scheduling API Function modules for planning, releasing and monitoring SAP R/3 jobs.	poss. BC-XBP
XBR **2.0**	CLI script	3.0G	Backup Integration API (also known as BRI) Backup and recovery functions for databases	poss. BC-BRI
XMB **0.1**	RFC	3.1G	Basis API Basis functions for monitoring basis function modules for scheduling, and releasing monitoring SAP R/3 jobs.	No
XMI	RFC	3.1G	External access to CCMS can be logged and monitored.	poss. BC-XBP
XOM **0.1**	RFC CLI	4.0A	Output Management Controlling and return messages for output management	poss. BC-XOM
XAL	RFC	4.0A	Monitoring API Interface for system monitoring and alert	

	management.

One thing which all of these RFC interfaces have in common is that their function modules call exactly the same function pools as the internal SAP R/3 operations. Figure 5.1 shows this using XBP. This does not, of course, apply to the XMI interface, since there is no internal equivalent.

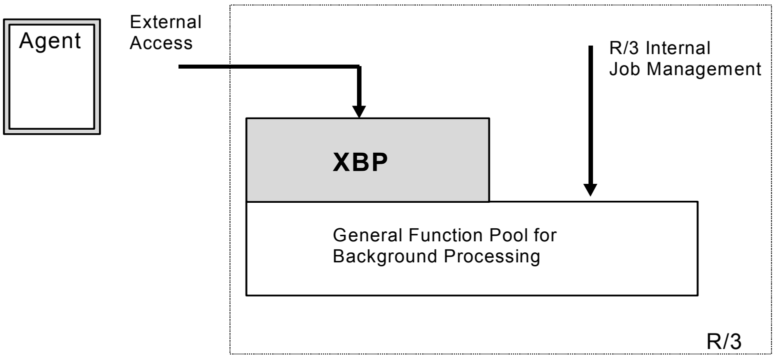

Agent
External Access
R/3 Internal Job Management
XBP
General Function Pool for Background Processing
R/3

Fig. 5.1: The external interface principle, exemplified by XBP

5.2 Naming Conventions

It is aimed to have a standardized naming convention for function modules, and this is widely kept to. Thus function modules for the XM-interface family have the prefix **BAPI**, an identification for the actual interface and arising from the target object and the intended action.

Syntactic structure for a function module name: `BAPI_<SS>_<Object>_<Action>`

Example of the naming convention: `BAPI_XBP_JOB_OPEN`

In the example, the actual interface is signified by XBP (eXternal interface for Background Processing).

5.3 Technical Basics

The most important technical basics for the external interfaces are XMI and RFC:

- XMI is an interface which logs the activities of users and agent programs each time a function module of an external interface is called. In particular, XMI logs the agent’s first access to the function module pool. At this point, the name of the external program is recorded and its version number checked.
- RFC (Remote Function Call) forms the communications platform for direct calls to the function modules which implement the interface on the SAP R/3 side.

5.3.1 XMI Monitor: External Access

Within SAP R/3 all external CCMS interfaces use the same function modules. These function modules can also be collected into an interface themselves. The name XMI (eXternal Monitoring Interface) was established since the interface was intended to log external access.

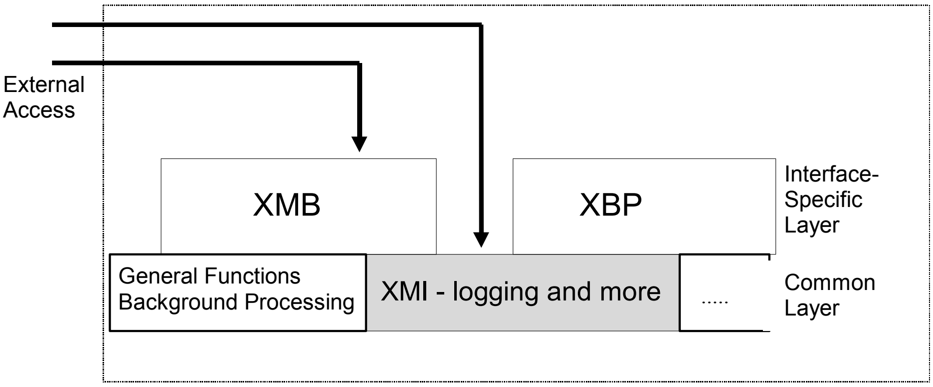


Fig. 5.2: XMI as a common layer for external interfaces.

For developers who want to integrate external tools into CCMS, the XMI interface remains almost invisible, appearing only at the beginning and end of a CCMS session in the form of two functions:

- **BAPI_XMI_LOGON:** Agent logs on to an external interface
- **BAPI_XMI_LOGOFF:** External program logs off from an external interface

You have the option of using further functions from the external tool

- **BAPI_XMI_ENTER_LOGMSG:** Writes a message to the XMI log
- **BAPI_XMI_GET_VERSIONS:** Queries the current version of the interface
- **BAPI_XMI_SELECT_LOG:** Reads the XMI message log

The information which is logged and collected in the access log can be viewed using Transaction RZ15.

The following is displayed: Name of the agent which tried to make contact with the SAP R/3 system, and that of its supplier, the user name (if a user logged on) and any changes or attempted changes which were carried out (for example, reset alert).

5.3.2 RFC Remote Function Call

A function within the SAP system can be called from outside via Remote Function Call (RFC), if the function is RFC-enabled. The RFC-enabling is just an indicator in the function attributes.

How it works

RFC is a general concept, allowing communication between SAP R/3 systems or between an SAP R/3 system and external programs.

In the case of an external system management tool we are dealing with an example of the latter, in which the program adopts the role of a client in relation to the SAP R/3 system. The service offered by the application server involves the delivery of internal system information or carrying out individual system management activities. In order to perform calls via RFC, the client needs the RFC library offered by SAP. This is a collection of necessary functions, which can be called from within C programs.

RFC works in sessions. In other words, the user opens an RFC, carries out RFC tasks and then concludes the session. In opening the session, the SAP R/3 logon procedure, with user ID and password, must be carried out. The user must be identifiable to and authorized by the SAP R/3 system. In other words, users must exist within the SAP R/3 system, which are then used by the external agent. Obviously, you need to set up authorizations for these users which correspond with the activities that they are to carry out.

The Most Important RFC Library Functions

The most important functions of the RFC library at a glance:

Name	*Short Description*
RfcOpen( )	A connection is made with an application server. The connection information is either specified directly or read from a file.
RfcCallReceive( )	Synchronous call from an RFC client Activates a function module in an SAP R/3 system. After processing, the client calling regains control.
RfcCall( )	Calls a function module without waiting for it to end.
RfcLastError( )	A function allowing you to analyze RFC errors. Detailed information is given in stdout format.
RfcClose( )	RFC connection closed, session finished.

ItCreate( )
ItDelete( )
	Storage space needs to be created and released when tables are exchanged between a client program and an SAP R/3 system.

Further RFC Documentation

- Other RFC functions and call forms are dealt with in the RFC tutorial

  SAP R/3 ABAP Development Workbench, which you can order from SAP.

- If you have an SAP R/3 installation running, equivalent information is available under 'SAP Library -> Basis Components -> Basis Services / Communication Interfaces -> Remote Communications'.

  This documentation answers questions relating to parameter transfer and configuration of an RFC connection (rfc destinations using a file), and also gives the syntax of RFC functions with examples.

  Before implementation, you are strongly recommended to consult this documentation.

Framework of a client RFC program in C

In order not to be too theoretical about RFC we have included a practical example. The framework of a program which is linked to the RFC library as an RFC client using the XMB interface would look something like this:

```
#include "saprfc.h"
main()
{
   rfc_handle = RfcOpen(&rfc_opt);
   function = "BAPI_XMI_LOGON";
   .....

   rfc_rc = RfcCallReceive(  rfc_handle,  function,  exporting,
   importing,  tables,  &exception  );

   function = "BAPI_XMI_SELECT_LOG";
   ......

   rfc_rc = RfcCallReceive(  rfc_handle,  function,  exporting,
   importing,  tables,  &exception  );

    /* a lot more action */
   function = "BAPI_XMI_LOGOFF";
   .........

   rfc_rc = RfcCallReceive(  rfc_handle,  function,  exporting,
   importing,  tables,  &exception  );

   RfcClose(rfc_handle);

}
```

From this we can see the model for a session between an SAP R/3 system and an external management tool. An RFC session contains one or more XMI sessions. Each XMI session contains a series of function calls to the individual interface function module.

Open RFC session	*Authorization from SAP R/3*
Open XMI session for XBP	*Identification of XM*
Call XBP functions	*Identification of agent user*
Open XMI session for XMB	*Identification of XM*
Call XMB functions	*Identification of agent user*
Call XBP functions	*Identification of agent user*
Close XMI session	
Close RFC session	

RFC is being developed further

Another small warning to finish with: the RFC interface is currently being developed further and improved. You should therefore always make sure that you have the current version of the RFC documentation.

6. XBP - External Job Scheduling Interface (external JOB-API)

As explained in the section above, an external interface is a collection of RFC-enabled functions. The XBP interface is an external interface to the SAP background processing system.

Note: The XBP interface must not be confused with the 'normal' batch API, which is a collection of non RFC-enabled functions. The 'normal' batch API is an internal ABAP-API to the SAP background processing system.

Why does SAP offer the XBP interface?

Many customers do not process their data with just one SAP system. They usually have a landscape consisting of one or more SAP systems as well as non-SAP systems. The non-SAP systems usually also have some kind of a background processing system.

There are interdependencies between the systems of such a landscape.

Example:

The non-SAP system A creates data using a background job. The SAP system B then processes this data in a job. This means that there is a job Y in SAP system B, which can only start after job X in non-SAP system A has finished.

Such a scenario demonstrates the need for a central job management system. The SAP background processing system, of course, cannot monitor jobs of non-SAP systems. In addition, the interdependencies between jobs even in a single system are sometimes so complex that they cannot be described with the functions of the internal batch API.

A central job management system (often referred to as 'external scheduler') connects to the SAP system via the XBP interface. The functionality of the XBP interface is not more complex than the one offered by the internal batch API, but based on this functionality the external scheduler implements its 'added value', for example graphical editors for job nets and complex start conditions.

In order to manage jobs centrally in a system landscape containing non-SAP systems, the non-SAP systems also have to provide an interface to which the external scheduler can connect.

6.1 What Is Required of the Interface

In order to be able to work in the SAP R/3 system, an external job scheduling system must be able to carry out the following activities within the SAP R/3 system:

- Create jobs
- Modify jobs
- Delete jobs
- Start jobs (start immediately)
- Terminate active jobs
- Access information about jobs (status, log, ...)

- Access information about resources in the SAP R/3 job scheduling system (number and status of background work processes)

There are XBP functions for carrying out all of these activities.

Any critical changes to SAP R/3 jobs (creation, editing, deletion etc) are recorded in the *XMI log* in the SAP R/3 system. You can display entries in this log using Transaction RZ15.

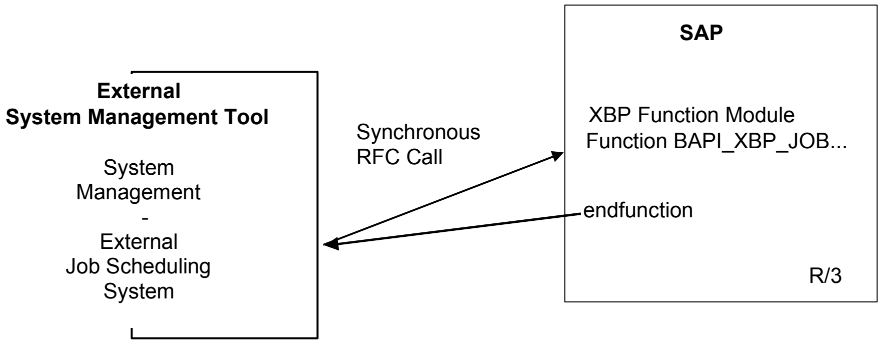


Fig 6.1: How a synchronous RFC function call to XBP works.

6.2 XBP Interface - Description

There follows a short overview of the function modules which make up XBP. Reference information about the modules is given in chapter 7.

The interface can be divided roughly into the following tasks:

Logging on/logging off

Function Modules	Short Description
BAPI_XMI_LOGON	Connect to the external management interface
BAPI_XMI_LOGOFF	End the SAP R/3 session of the external job management system

Defining jobs

Function Modules	Short Description
BAPI_XBP_JOB_OPEN	Create a job
BAPI_XBP_JOB_ADD_ABAP_STEP	Add an ABAP step to an existing job
BAPI_XBP_JOB_ADD_EXT_STEP	Add an external program to an existing job
BAPI_XBP_JOB_CLOSE	Finish job creation
BAPI_XBP_JOB_DEFINITION_GET	Read job definitions from the SAP R/3 system

Working with jobs

Function Modules	Short Description
BAPI_XBP_JOB_ABORT	Abort job
BAPI_XBP_JOB_DELETE	Delete job

Function Modules	Short Description
BAPI_XBP_JOB_COPY	Copy a job with all attributes
BAPI_XBP_JOB_HEADER_MODIFY	Modify key job parameters

Modifying jobs

Function Modules	Short Description
BAPI_XBP_JOB_ABAP_STEP_MODIFY	Modify an ABAP step
BAPI_XBP_JOB_EXT_STEP_MODIFY	Modify an external step

Starting jobs and triggering events

Function Modules	Short Description
BAPI_XBP_JOB_START_ASAP	Start as soon as possible
BAPI_XBP_JOB_START_IMMEDIATELY	Start immediately
BAPI_XBP_EVENT_RAISE	Trigger an event from outside

Adding, changing, and deleting job steps via XMI

Function Modules	Short Description
BAPI_XBP_ADD_JOB_STEP	Add and insert a step to a job via XMI
BAPI_XBP_MODIFY_JOB_STEP	Change and delete a step of a job via XMI

Intercepting and confirming jobs

Function Modules	Short Description
BAPI_XBP_GET_INTERCEPTED_JOBS	Retrieve jobs with the new status 'Intercepted'
BAPI_XBP_CONFIRM_JOB	Confirm jobs in general
BAPI_XBP_SPECIAL_CONFIRM_JOB	Set special confirmation types for jobs (intercept, parent/child)
BAPI_XBP_MODIFY_CRITERIA_TABLE	Modify the criteria table

Monitoring / Controlling the SAP R/3 job scheduling system

Function Modules	Short Description
BAPI_XBP_JOB_STATUS_CHECK	Is the job status still correct? Is the WP working?
BAPI_XBP_JOB_COUNT	How many jobs exist with a particular name?
BAPI_XBP_JOB_SELECT	Select sets of jobs by various criteria
BAPI_XBP_JOB_STATUS_GET	Read job status
BAPI_XBP_JOBLIST_STATUS_GET	Determine status of a list of jobs
BAPI_XBP_JOB_JOBLOG_READ	Read job log

Function Modules	Short Description
BAPI_XBP_JOB_READ	Obtain key job parameters from job header and job steps
BAPI_XBP_JOB_SPOOLLIST_READ_20	Replaces the function modules BAPI_XBP_JOB_SPOOLIST_READ BAPI_XBP_JOB_SPOOLLIST_READ _RW
BAPI_XBP_JOB_SPOOLLIST_READ	Read spool data
BAPI_XBP_JOB_SPOOLLIST_READ_RW	Read spool list in raw format
BAPI_XBP_NEW_FUNC_CHECK	Read and change status of interception and parent/child functionality
BAPI_XBP_JOB_CHILDREN_GET	Get all children of a job
BAPI-XBP_JOB_PARENT_CHILD_INFO	Determine parent/child relation

Information about the Background System

Function Modules	Short Description
BAPI_XBP_ GET_CURR_BP_RESOURCES	Read current background server and work processes
BAPI_XBP_ GET_BP_RESRC_ON_DATE	Background work processes at a particular time
BAPI_XBP_GET_BP_SRVRES_ON_DATE	Is there a background work process on a server at a particular time?
BAPI_XBP_VARIANT_INFO_GET	Does the ABAP program have variants?

Value help functions

Function Modules	Short Description
BAPI_XBP_REPORT_SEARCH	Return a list of ABAP reports available in the current system
BAPI_XBP_EXT_COMM_SEARCH	Return a list of external commands available in the current system
BAPI_XBP_OUTPUT_DEVICE_SEARCH	Return a list of output devices available in the current system
BAPI_XBP_PRINT_FORMAT_SEARCH	Return a list of print formats available for a certain printer
BAPI_XBP_VARIANT_INFO_GET	Get the variants of an ABAP program

7. XBP Reference Manual

This is the reference manual for the interface function modules. Please bear in mind that the very latest interface parameter types can only be obtained from your system (Transaction SE37, Function Module - interface).

Concerning the following function descriptions the notions IMPORT and EXPORT are described from the view of the respective function module. This means, the module is called with the import parameters and returns export parameters.

Whenever there is a new or enhanced feature in XBP 2.0, there is an entry ('New in XBP 2.0') in the table of the function description.

7.1 Requirements for Using the XBP Interface

The requirement for using the XBP interface is an existing XMI session. The technical requirements (RFC, SAP R/3 Version) are contained in earlier chapters.

7.1.1 Logging on to the SAP R/3 System with an External Job Management System

Before you call a function module in the XBP interface for the first time, it is important that:

- the external job management system logs onto the SAP R/3 system first, using an SAP R/3 user name and password. (C function `RfcOpen`).
- the external job management system is authenticated by the CCMS external interface administration using the function module BAPI_XMI_LOGON:

Function name	BAPI_XMI_LOGON
Short description	Connecting to the external Management Interfaces
BAPI object name	**SystemMngmtSession**
BAPI method name	**Logon**

RFC interface	`function BAPI_XMI_LOGON`
	`importing`
	`  EXTCOMPANY like BAPIXMLOGR-EXTCOMPANY`
	`    type RFC_CHAR length 16`
	`  EXTPRODUCT like BAPIXMLOGR-EXTPRODUCT`
	`    type RFC_CHAR length 16`
	`  INTERFACE like BAPIXMLOGR-INTERFACE`
	`    optional`
	`    type RFC_CHAR length 3`
	`  VERSION like BAPIXMLOGR-VERSION`
	`    optional`
	`    type RFC_CHAR length 10`
	`exporting`
	`  RETURN structure BAPIRET2 length 548`
	`     number of fields 14`
	`  SESSIONID like BAPIXMLOGR-SESSIONID`
	`     type RFC_CHAR length 24`

Parameter (Input)	• EXTCOMPANY is the name of the supplier of the external management tool.

	• EXTPRODUCT is the product name of external management tool. • INTERFACE (optional) is the interface for logging on. • VERSION (optional) is the version of the interface in SAP R/3 required by the external tool.
Parameter (Output)	• SESSIONID is the ID of the RFC connection. • BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_ALREADY_LOGGED_ON: This product is already logged onto the interface. • MSG_ALREADY_LOGGED_ON_GEN: The company has already logged on. • MSG_CANT_LOG: Activity was terminated, because the SAP R/3 XMI logging mechanism returned an error. • MSG_INVALID_PARAMETERS: EXTCOMPANY and EXTPRODUCT are different within a session. • MSG_LOGON_DENIED: Logon was denied because the SAP R/3 user used by the external job management session to log onto the SAP R/3 system is not authorized to work with the external job management system. • MSG_LOGON_DENIED_GEN: You have no authorization for general logon. • MSG_PROBLEM_DETECTED: XMI problem which cannot be further specified. • MSG_UNKNOWN_INTERFACE: The interface required by the external tool is not supported by the system. • MSG_UNKNOWN_VERSION: The version required by the external tool is not supported by the system.

The logon will only be successful if you assign the following authorization values to the user for the SAP R/3 authorization object **S_XMI_PROD**:

- Name of the company which supplies the external job management system
- Program name of the external job management system
- Name of the interface which the user wants to work with

7.1.2 External Job Management System - Logging Off

Function name	BAPI_XMI_LOGOFF
Short description	If you want to end the external job management system’s SAP R/3 session, you first need to call the BAPI_XMI_LOGOFF function module.
BAPI object name	**SystemMngmtSession**
BAPI method name	**Logoff**

RFC interface	`function BAPI_XMI_LOGOFF`
	`importing`
	`  INTERFACE like BAPIXMLOGR-INTERFACE`
	`      optional`
	`      type RFC_CHAR length 3`
	`exporting`
	`  RETURN like BAPIRET2 structure`
	`     length 548 number of fields 14`

Parameter (Input)	• INTERFACE (optional) from which you should log off.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: There is no connection with the SAP R/3 system. • MSG_PROBLEM_DETECTED: XMI problem which cannot be further specified.
Remarks	Afterwards you must close the RFC connection using the C RFC function `rfcClose`.

7.2 Defining Jobs

You must observe the following procedure in defining SAP R/3 jobs:

- Open job (BAPI_XBP_JOB_OPEN)
- Assign (BAPI_XBP_JOB_ADD_ABAP_STEP / BAPI_XBP_JOB_ADD_EXT_STEP) one or more job steps (ABAP Programs or external Programs) to the job.
- Close job and assign start time if required (BAPI_XBP_JOB_CLOSE).

After having defined a job, you can read the definition with the function module BAPI_XBP_JOB_DEFINITION_GET.

7.2.1 Opening Jobs

Function name	**BAPI_XBP_JOB_OPEN**
Short description	If you want to create a new job, you must first 'open' it. When you open a job, its name is recorded and a job number assigned to it. The job name and number are used as a unique key for all subsequent function calls.
New in XBP 2.0	**The optional import parameter JOBCLASS has been added (see description below).**
BAPI object name	**BackgroundJob**
BAPI method name	**Open**

RFC interface	`function BAPI_XBP_JOB_OPEN`
	`importing`
	`  JOBNAME like BAPIXMJOB-JOBNAME`
	`       type RFC_CHAR length 32`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`  JOBCLASS like BAPIXMJOB-JOBCLASS`
	`       optional`
	`       type RFC_CHAR length 1`
	`exporting`
	`  JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`       type RFC_CHAR length 8`
	`  RETURN structure BAPIRET2 length 548`
	`       number of fields 14`

Parameter (Input)	• JOBCLASS is an optional import parameter. The caller can choose a job class (A, B, or C).
Parameter (Output)	• JOBCOUNT is the system-generated job number. • BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_JOBNAME_MISSING : You have not entered a job name. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has found an error. • MSG_EXT_USER_MISSING: The name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not yet logged on to the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.2.2 Assigning an ABAP Program to a Job Step

<table>
<tr><th>Function name</th><th>BAPI_XBP_JOB_ADD_ABAP_STEP</th></tr>
<tr><td>Short description</td><td>An ABAP program which you intend to run within a job can be assigned to a job step. Enter a valid variant name for the program if the ABAP program works with variants.</td></tr>
<tr><td><b>New in XBP 2.0</b></td><td><b>Two new optional import parameters ALLPRIPAR and ALLARCPAR have been added (see description below).</b></td></tr>
<tr><td>BAPI object name</td><td><b>BackgroundJob</b></td></tr>
<tr><td>BAPI method name</td><td><b>AddABAPStep</b></td></tr>
<tr><td>RFC interface</td><td><pre>
function BAPI_XBP_JOB_ADD_ABAP_STEP

importing
  JOBNAME like BAPIXMJOB-JOBNAME
     type RFC_CHAR length 32
  JOBCOUNT like BAPIXMJOB-JOBCOUNT
     type RFC_CHAR length 8
  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER
     type RFC_CHAR length 16
  ABAP_PROGRAM_NAME like BAPIXMREP-REPORTID
     type RFC_CHAR length 40
  ABAP_VARIANT_NAME like BAPIXMREP-VARIANTNAM
     optional default SPACE
     type RFC_CHAR length 14
  SAP_USER_NAME like BAPIXMSTEP-AUTHCKNAM
     optional default SY-UNAME
     type RFC_CHAR length 12
  LANGUAGE like BAPIXMSTEP-LANGUAGE
     optional default SY-LANGU
     type RFC_CHAR length 2
  PRINT_PARAMETERS structure BAPIXMPRNT 12
     optional default SPACE
     length 48 number of fields 12
  ARCHIVE_PARAMETERS structure BAPIXMARCH
     optional default SPACE
     length 23 number of fields 3
  ALLPRIPAR structure BAPIPRIPAR
     optional default SPACE
     length 176 number of fields 22
  ALLARCPAR structure BAPIARCPAR
     optional default SPACE
     length 328 number of fields 18

exporting
  STEP_NUMBER like BAPIXMJOB-STEPCOUNT
     type RFC_INT4 length 4
  RETURN structure BAPIRET2 length 548
     number of fields 14
</pre></td></tr>
<tr><td>Parameter (Input)</td><td><ul>
<li>JOBNAME is the name of a background job.</li>
<li>JOBCOUNT is the ID number of a job. Together with the job name, the job number identifies the job uniquely.</li>
<li>EXTERNAL_USER_NAME is the name of the user in the external scheduler who called the function.</li>
<li>ABAP_PROGRAM_NAME is name of the ABAP program that is to be executed in this job step. The program must be type 1 (interactively executable).</li>
<li>ABAP_VARIANT_NAME is an optional parameter to determine variants for the specified ABAP report.</li>
</ul></td></tr>
</table>

	• SAP_USER_NAME is an optional parameter to specify an R/3 user with whose authorizations the job is processed. • LANGUAGE is an optional parameter to enter the two-digit SAP language key for this job step. You can find the language keys that are allowed in table T002. • PRINT_PARAMETERS (optional): You can use this structure to transfer parameters for the printer products of the job step to the spool system. You can specify on which output device the print requests are to be printed, whether print requests should be retained in the spool system or output immediately and so on. If no structure is transferred, the system uses the print parameters in the default values of the user you used to log onto the R/3 system (rfcOpen). • ARCHIVE_PARAMETERS (optional): You can use this structure to transfer parameters for archiving the printer results of the job step to the optical archiving system. You can use this parameter to specify whether or not the spool requests of the job step are to be archived. If no structure is transferred, the system uses the archiving details in the fixed values of the user you used to log onto the R/3 system (rfcOpen). • ALLPRIPAR and ALLARCPAR are optional structures for the specification of all print and archive parameters and complement PRINT_PARAMETERS and ARCHIVE_PARAMETERS. If ALLPRIPAR and ALLARCPAR are initial, PRINT_PARAMETERS and ARCHIVE_PARAMETERS are evaluated. If ALLPRIPAR and ALLARCPAR contain a value, this value is used. ALLARCPAR is a set of archive parameters used for steps consisting of an ABAP program. The ALLARCPAR parameter contains the whole set of archive parameters that can be used by an ABAP program. ALLPRIPAR is a set of print parameters used for steps consisting of an ABAP program. The ALLPRIPAR parameter contains the whole set of print parameters that can be used by an ABAP program.
Parameter (Output)	• STEP_NUMBER is the number of job steps. • BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_JOBID_MISSING: You did not enter a job number. • MSG_JOBNAME_MISSING: You did not enter a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system. • MSG_PROGNAME_MISSING: You did not enter an ABAP program name. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has found an error. • MSG_EXT_USER_MISSING: The external user name is missing. This is the name of a user in the external job scheduling system. • MSG_INVALID_PRINT_PARAMS: Printer entry invalid. • MSG_INVALID_ARCHIVE_PARAMS: Archiving parameters invalid. • MSG_NO_ARCHIVE_INFO: Archiving information not given.

	• MSG_CANT_LOG: Activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool is not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.2.3 Assigning an External Program to a Job Step

Function name	BAPI_XBP_JOB_ADD_EXT_STEP
Short description	You can assign a program to a job step which runs outside the SAP R/3 system, for example a C program. Enter the name of the program and that of the host on which the external program is to run
BAPI object name	**BackgroundJob**
BAPI method name	**AddExternalStep**

RFC interface	`function BAPI_XBP_JOB_ADD_EXT_STEP`
	`importing`
	`  JOBNAME like BAPIXMJOB-JOBNAME`
	`     type RFC_CHAR length 32`
	`  JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`     type RFC_CHAR length 8`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`  EXT_PROGRAM_NAME like BAPIXMSTEP-PROGRAM`
	`     type RFC_CHAR length 128`
	`  EXT_PROGRAM_PARAMETERS like BAPIXMSTEP-`
	`  PARAMETER`
	`     optional default SPACE`
	`     type RFC_CHAR length 255`
	`  WAIT_FOR_TERMINATION like BAPIXMAUX-CHAR1`
	`     optional default 'X'`
	`     type RFC_CHAR length 1`
	`  SAP_USER_NAME like BAPIXMSTEP-AUTHCKNAM`
	`     optional default SY-UNAME`
	`     type RFC_CHAR length 12`
	`  TARGET_HOST like BAPIXMSTEP-XPGTGTSYS`
	`     type RFC_CHAR length 32`
	
	`exporting`
	`  RETURN structure BAPIRET2 length 548`
	`      number of fields 14`
	`  STEP_NUMBER like BAPIXMJOB_STEPCOUNT`
	`      type RFC_INT4 length 4`

Parameter (Input)	• JOBNAME is the name of a background job. • JOBCOUNT is the ID number of a job. Together with the job name, the job number identifies the job uniquely. • EXTERNAL_USER_NAME is the name of the user in the external scheduler who caused the function call. • EXT_PROGRAM_NAME is the name of the program that is to be executed by the background processing system. • EXT_PROGRAM_PARAMETERS is an optional parameter that may be required by the external program at runtime. The parameters are transferred to the external program at the start time as character strings. • WAIT_FOR_TERMINATION is an optional parameter, which has the effect that the background job waits for the external program to finish before processing the next job step (synchronous job step

	processing). This option is activated by default. • TARGET_HOST is the name of the host computer on which the external program is to be executed. • SAP_USER_NAME is an optional parameter to specify R/3 users with whose authorizations the job is processed.
Parameter (Output)	• STEP_NUMBER is the number of job steps. • BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_JOBID_MISSING: You did not enter a job number. • MSG_JOBNAME_MISSING: You did not enter a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in SAP R/3 system. • MSG_PROGNAME_MISSING: You did not specify the name of the external program. • MSG_TARGETHOST_MISSING: You did not specify the target host. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered a problem. • MSG_EXT_USER_MISSING: External user name is missing. This is the name of a user in the external job scheduling system. • MSG_CANT_LOG: Activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged on to the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.2.4 Closing Job Definitions

Function name	BAPI_XBP_JOB_CLOSE
Short description	You close a job definition using the BAPI_XBP_JOB_CLOSE function module.
New in XBP 2.0	**The optional import parameter RECIPIENT_OBJ has been added (see parameter description below).**
BAPI object name	**BackgroundJob**
BAPI method name	**Close**

RFC interface	`function: BAPI_XBP_JOB_CLOSE`
	`importing`
	`  JOBNAME like BAPIXMJOB_JOBNAME`
	`     type RFC_CHAR length 32`
	`  JOBCOUNT like BAPIXMJOB_JOBCOUNT`
	`     type RFC_CHAR length 8`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR_EXTUSER`
	`    type RFC_CHAR length 16`
	`  TARGET_SERVER like BAPIXMJOB-EXECSERVER`
	`     optional type RFC_CHAR length 20`
	`  RECIPIENT_OBJ structure SWOTOBJID`
	`      optional length 100 number of fields 4`
	`exporting`
	`  RETURN structure BAPIRET2 length 548`
	`     number of fields 14`

Parameter (Input)	• JOBNAME is the name of a background job. • JOBCOUNT is the ID number of a job. Together with the job name, the job number identifies the job uniquely.

	• EXTERNAL_USER_NAME is the name of the user in the external scheduler who called the function. • TARGET_SERVER is an optional parameter with which you can have the job executed on the R/3 instance that you specify. • SWOTOBJID is a structure with the following fields: OGSYS char 10 OBJTYPE char 10 OBJKEY char 70 DESCRIBE char 10 The structure describes the spool list recipient. However, corresponding to the function module job_close, the format is internal and not simply the name of the spool list recipient.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_JOBID_MISSING: You did not enter a job number. • MSG_JOBNAME_MISSING: You did not enter a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in SAP R/3 system. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_EXT_USER_MISSING: Name of external user is missing. This is the name of a user in the external job scheduling system. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging system returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged on to the CCMS XMI interface. Therefore, the activity cannot be carried out. • MSG_CHILD_REGISTER_ERROR: An error occurred during child registration.

7.2.5 Reading Job Definitions from the SAP R/3 System

Function name	BAPI_XBP_JOB_DEFINITION_GET
Short description	You use the BAPI_XBP_JOB_DEFINITION_GET function module to read all the data associated with a job (name, job class, steps, start conditions etc.).
BAPI object name	**BackgroundJob**
BAPI method name	**GetDefinition**

RFC interface	`function BAPI_XBP_JOB_DEFINITION_GET`
	`importing`
	`  JOBNAME like BAPIXMJOB_JOBNAME`
	`     type RFC_CHAR length 32`
	`  JOBCOUNT like BAPIXMJOB_JOBCOUNT`
	`     type RFC_CHAR length 8`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR_EXTUSER`
	`    type RFC_CHAR length 16`
	
	`exporting`
	`  JOB_HEAD structure BAPIXMJOB`
	`    length 388 number of fields 35`
	`  RETURN structure BAPIRET2 length 548`
	`    number of fields 14`
	
	`tables`
	`  STEP_TBL structure BAPIXMSTEP`
	`     length 980 number of fields 56`

Parameter (Input)	• JOBNAME is the name of a background job. • JOBCOUNT is the ID number of a job. Together with the job name, the job number identifies the job uniquely. • EXTERNAL_USER_NAME is the name of the user in the external scheduler who caused the function call.
Parameter (Output)	• JOB_HEAD is job header data (name, job class...). • BAPIRET2 is the return structure used by BAPIs.
Tables	• STEP_TBL is a table with job step data.
MessageIDs	• MSG_JOBID_MISSING: You have not specified the job ID number. • MSG_JOBNAME_MISSING: You did not enter a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in SAP R/3 system. • MSG_NO_JOBSTEPS: Job does not yet have any steps. This can occur if a job already exists in the database but no steps have been assigned to it. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.3 Starting a Job

You can start ‘scheduled’ or ‘intercepted’ jobs using the XBP interface in the SAP R/3 system with the start time types ‘start immediately’ or ‘as soon as possible‘. To do this, use the following function modules:

- Start job immediately (BAPI_XBP_JOB_START_IMMEDIATELY)
- Start job as soon as possible (BAPI_XBP_JOB_START_ASAP)

Besides these function modules, you can use the function BAPI_XBP_EVENT_RAISE to trigger a background processing event. All jobs with the status ‘released’ waiting for this event will then be started by the R/3 job scheduler.

Also note the new function BAPI_XBP_JOB_HEADER_MODIFY described in chapter 7.5.1 ‘Modifying Job Global Data’. With this function a start condition can be assigned to a job with the status ‘scheduled’. The R/3 job scheduler then takes care of starting the job.

7.3.1 Starting Jobs Immediately

Function name	**BAPI_XBP_JOB_START_IMMEDIATELY**
Short description	This function attempts to start a job immediately. If the job cannot be started immediately because, for example, all background work processes are busy, the function reports this to its caller. The job is then started after a delay.
New in XBP 2.0	**This function module can now also start jobs with status ‘intercepted’ and ‘released’.**
BAPI object name	**BackgroundJob**
BAPI method name	**StartImmediately**

RFC interface	`function BAPI_XBP_JOB_START_IMMEDIATELY`
	`importing`
	`  JOBNAME like BAPIXMJOB_JOBNAME`
	`     type RFC_CHAR length 32`
	`  JOBCOUNT like BAPIXMJOB_JOBCOUNT`
	`     type RFC_CHAR length 8`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR_EXTUSER`
	`     type RFC_CHAR length 16`
	`  TARGET_SERVER like BAPIXMJOB-EXECSERVER`
	`     type RFC_CHAR length 20`
	
	`exporting`
	`  RETURN structure BAPIRET2 length 548`
	`      number of fields 14`

Parameter (Input)	• JOBNAME is the name of a background job. • JOBCOUNT is the ID number of a job. Together with the job name, the job number identifies the job uniquely. • EXTERNAL_USER_NAME is the name of the user in the external scheduler who called the function. • TARGET_SERVER is a parameter with which you can have the job executed on the R/3 instance that you specify.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_JOBID_MISSING: You did not enter a job number. • MSG_JOBNAME_MISSING: You did not enter a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system.

	• MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system discovered an error. • MSG_NO_IMMEDIATE_START_POSS: Job cannot be started immediately, since no background work processes are free. Note that the Job is not released in this case and is therefore not waiting to run inside the system. • MSG_PRIVILEGE_MISSING: The SAP R/3 user with which the external job management system logged on to the SAP R/3 system, is not authorized to release the job. • MSG_EXT_USER_MISSING: External user name is missing. This is the name of a user in the external job scheduling system. • MSG_CANT_LOG: Activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged on to the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.3.2 Starting Jobs as Soon as Possible

Unlike the BAPI_XBP_JOB_START_IMMEDIATELY function, no error is returned if the job cannot be started immediately.

Function name	BAPI_XBP_JOB_START_ASAP
Short description	You use this function to start a job as soon as possible. You can specify the application server this job should be targeted for.
New in XBP 2.0	**This function module can now also start jobs with status ‘intercepted’ and ‘released’.**
BAPI object name	**BackgroundJob**
BAPI method name	**StartAsSoonAsPossible**

RFC interface	`function BAPI_XBP_JOB_START_ASAP`
	`importing`
	`  JOBNAME like BAPIXMJOB_JOBNAME`
	`     type RFC_CHAR length 32`
	`  JOBCOUNT like BAPIXMJOB_JOBCOUNT`
	`     type RFC_CHAR length 8`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR_EXTUSER`
	`     type RFC_CHAR length 16`
	`  TARGET_SERVER like BAPIXMJOB-EXECSERVER`
	`     type RFC_CHAR length 20`
	
	`exporting`
	`  RETURN structure BAPIRET2`
	`     length 548 number of fields 14`

Parameter (Input)	• JOBNAME is the name of a background job. • JOBCOUNT is the ID number of a job. Together with the job name, the job number identifies the job uniquely. • EXTERNAL_USER_NAME is the name of the user in the external scheduler who called the function. • TARGET_SERVER is a parameter with which you can have the job executed on the R/3 instance that you specify.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.

MessageIDs	• MSG_JOBID_MISSING: You did not enter a job ID number • MSG_JOBNAME_MISSING: You did not enter a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system found an error. • MSG_PRIVILEGE_MISSING: The SAP R/3 user with which the external job management system logged onto the SAP R/3 system is not authorized to release the job. • MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_CANT_LOG: Activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged on to the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.3.3 Triggering an Event from Outside

Function name	**BAPI_XBP_EVENT_RAISE**
Short description	This function module is the XBP-equivalent of BP_EVENT_RAISE. With this function, you can trigger an event from outside.
New in XBP 2.0	**This function module can now also start jobs with status ‘intercept’ and ‘released’.**
BAPI object name	**BackgroundJob**
BAPI method name	**EventRaise**

RFC interface	`function BAPI_XBP_EVENT_RAISE`
	`importing`
	`  EVENTID like BAPIXMLOGR-EVENTID`
	`    type RFC_CHAR length 32`
	`  EVENTPARM like BAPIXMJOB-EVENTPARM optional`
	`    type RFC_CHAR length 64`
	`  EXTERNAL_USER_NAME like BAPIXMJOB-EXTUSER`
	`     type RFC_CHAR length 16`
	`exporting`
	`   RETURN structure BAPIRET2`
	`     length 548 number of fields 14`

Parameter (Input)	• EVENTID is the name of the event. • EVENTPARM are optional parameters of the event. • EXTERNAL_USER_NAME is the name of the XBP user.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_EVENT_DOES_NOT_EXIST: Event does not exist in the SAP R/3 system. • MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_CANT_LOG: Activity was terminated because the SAP R/3 XMI logging mechanism returned an error.

	• MSG_NOT_LOGGED_ON: The external management tool has not logged on to the CCMS XMI interface. The activity cannot therefore be carried out. • MSG_PARAM_MISSING: EventID is missing. • MSG_EVENT_RAISE_FAILED: Event could not be triggered.

7.4 Copying Jobs

Function name	BAPI_XBP_JOB_COPY
Short description	With this function module you can copy a job or, to be more precise, a job definition. The job is copied including all definition data, except for the start conditions. The copy has the status ‘scheduled’. A name can be specified for the target with the parameter **target_jobname**. If no name is specified, the target job has the same name as the source job. With the optional STEP_NUMBER parameter you can specify that not all steps of the original job should be copied. In this case this parameter specifies the number of the very first step to start copying from.
New in XBP 2.0	**The entire function module is new in XBP 2.0.**
BAPI object name	**BackgroundJob**
BAPI method name	**Copy**

RFC interface	`function BAPI_XBP_JOB_COPY`
	
	`importing`
	`  SOURCE_JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`     type RFC_CHAR length 8`
	`  SOURCE_JOBNAME like BAPIXMJOB-JOBNAME`
	`     type RFC_CHAR length 32`
	`  TARGET_JOBNAME like BAPIXMJOB-JOBNAME`
	`     optional default SPACE`
	`     type RFC_CHAR length 32`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`  STEP_NUMBER like BAPIXMJOB-STEPCOUNT`
	`     optional default 0`
	`     type RFC_INT4 length 4`
	`exporting`
	`  TARGET_JOBCOUNT  structure BAPIXMJOB-JOBCOUNT`
	`     type RFC_CHAR length 8`
	`  RETURN like BAPIRET2`
	`     length 548 number of fields 14`

Parameter (Input)	• SOURCE_JOBCOUNT is the ID of the job to be copied. • SOURCE_JOBNAME is the name of the job to be copied. • TARGET_JOBNAME (optional) is the name of the target job. If this parameter is not specified, the name of the target job is the name of the source job. • EXTERNAL_USER_NAME is the name of the XBP user. • STEP_NUMBER (optional) specifies the number of the first step to start copying job data from. Valid values of this parameters are 0, 1, ..., <highest step number>. If 0 and 1 all steps are copied.

Parameter (Output)	- TARGET_JOBCOUNT: Job ID number of the newly created job. - BAPIRET2: Return structure for function modules used for BAPIs.
MessageIDs	- MSG_JOBNAME_MISSING: Name of the job to be copied is missing. - MSG_JOBID_MISSING: Job ID number to be copied is missing. - MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. - MSG_CANT_LOG: Activity was terminated because the SAP R/3 XMI logging mechanism returned an error. - MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. - MSG_WRONG_STEP_NUMBER: The value of the parameter STEP_NUMBER is higher than the number of steps of the jobs. - MSG_CANT_SELECT: Specified source job cannot be selected. - MSG_PRIVILEGE_MISSING : Authorization for copying jobs is missing. - MSG_PROBLEM_DETECTED : Problems other than those stated above. - MSG_JOB_DOES_NOT_EXIST: Specified job does not exist.

7.5 Controlling Jobs

Control functions currently include the modification of job global data, termination of active jobs, and deletion of obsolete – not running - jobs.

7.5.1 Modifying Job Global Data

Global job data can be changed with the following function module. For example, a start condition can be assigned to a job. By defining a start condition, the external scheduler can give control back to the R/3 scheduling mechanism. This is useful for downtimes of the external scheduler.

Function name	**BAPI_XBP_JOB_HEADER_MODIFY**
Short description	This function module is intended for modifying key job parameters, which are stored in the job header. New values can be set using field masks.
New in XBP 2.0	**The entire function module is new in XBP 2.0**
BAPI object name	**BackgroundJob**
BAPI method name	

RFC interface	function BAPI_XBP_JOB_HEADER_MODIFY
	importing
	`JOBNAME like BAPIXMJOB-JOBNAME`
	`   type RFC_CHAR length 32`
	`JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`   type RFC_CHAR length 8`
	`JOB_HEADER structure BP20HEAD`
	`  length 385 number of fields 29`
	`EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`   type RFC_CHAR length 16`

	DONT_RELEASE like BAPIXMINFO-DONOTRELE optional type RFC_CHAR length 1 MASK structure BP20HMSK OPTIONAL length 5 number of fields 5 exporting RETURN like BAPIRET2 length 548 number of fields 14
Parameter (Input)	• JOBNAME is the name of the job, whose header needs to be modified. • JOBCOUNT is the ID number of the job whose header needs to be modified. • JOB_HEADER is the new job header, some fields of which should replace old ones (see the MASK parameter). • EXTERNAL_USER_NAME: The name of the XBP user. • DONT_RELEASE (optional) specifies whether the job should be released after the header change. • MASK (optional) is a mask with indicators for each field in header. If a field is selected, it is expected that JOB_HEADER has a new value for this field. If a field is left blank, the corresponding field in the header stays intact.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_JOBNAME_MISSING : Job name missing. • MSG_JOBID_MISSING : Job ID number is missing. • MSG_EXT_USER_MISSING : External user name is missing. • MSG_CANT_LOG : Activity was terminated because the R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON : The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. • MSG_JOB_DOES_NOT_EXIST : Specified job does not exist. • MSG_CANT_READ_JOBDATA : Unable to read job data. • MSG_INVALID_NEW_JOBDATA : Invalid new job data. • MSG_NO_MODIFY_PRIVILEGE_GIVEN : Current user does not have modify authorization. • MSG_NO_RELEASE_PRIVILEGE_GIVEN : Current user does not have release authorization. • MSG_CANT_ENQ_JOB : An error occurred while locking job in a database table. • MSG_CANT_RELEASE_JOB : Cannot release the job. • MSG_JOB_NOSTEPS : There are no steps in the job. • MSG_JOBCOUNT_MISSING : Job ID number is missing. • MSG_INVALID_TARGET : Invalid target server. • MSG_CANT_START_JOB_IMMEDIATELY : Immediate job start failed. • MSG_INVALID_STARTDATE : Invalid job start date.

	• MSG_JOB_NOT_MODIFIABLE_ANYMORE : The job is not modifiable anymore.

7.5.2 Aborting a Job

You can terminate an active job using the function module BAPI_XBP_JOB_ABORT.

Function name	BAPI_XBP_JOB_ABORT
Short description	Abort a running job.
BAPI object name	**BackgroundJob**
BAPI method name	**Abort**

RFC interface	`function BAPI_XBP_JOB_ABORT`
	`importing`
	`  JOBNAME like BAPIXMJOB-JOBNAME`
	`     type RFC_CHAR length 32`
	`  JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`    type RFC_CHAR length 8`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	
	`exporting`
	`  RETURN structure BAPIRET2 length 548`
	`    number of fields 14`

Parameter (Input)	• JOBCOUNT and JOBNAME of the job to be aborted. • EXTERNAL_USER_NAME is the name of the XBP user.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_JOBID_MISSING: You did not enter a job ID. • MSG_JOBNAME_MISSING: You did not enter a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system. • MSG_EXT_USER_MISSING: The name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_JOB_NOT_ACTIVE: The job cannot be terminated since it is not active. • MSG_NO_ABORT_PRIVILEGE: The SAP R/3 user used by the external job management system to log onto the SAP R/3 system is not authorized to terminate the job. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.5.3 Deleting a Job

Function name	**BAPI_XBP_JOB_DELETE**
Short description	Delete a - not running - job.
BAPI object name	**BackgroundJob**
BAPI method name	**Delete**

RFC interface	`function BAPI_XBP_JOB_DELETE`
	`importing`
	`  JOBNAME like BAPIXMJOB-JOBNAME`
	`     type RFC_CHAR length 32`
	`  JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`    type RFC_CHAR length 8`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	
	`exporting`
	`  RETURN structure BAPIRET2 length 548`
	`    number of fields 14`

Parameter (Input)	• JOBCOUNT and JOBNAME of the job to be deleted. • EXTERNAL_USER_NAME is the name of the XBP user.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_JOBID_MISSING: You did not enter a job number. • MSG_JOBNAME_MISSING: You did not enter a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system. • MSG_NO_JOB_FOUND: The job existed when the function module started, but cannot be found anymore for the actual deletion. • MSG_EXT_USER_MISSING: The name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_JOB_RUNNING: The job cannot be terminated since it is currently active. • MSG_NO_DELETE_PRIVILEGE: The SAP R/3 user used by the external job management system to log onto the SAP R/3 system is not authorized to delete the job. • MSG_ CANT_DEL_IN_JOBTABLE: While trying to delete the job one of the tables that contains job data entries can not be deleted. • MSG_CANT_DEL_JOBLOG: Failed to find or delete the job log of the specified job. Deleting the job continues. • MSG_PROBLEM_PRED_SUCC: A problem with the handling of the predecessor or successor of the deleted job occurred. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot

	be carried out. - MSG_COMMIT_FAILED: Failed to commit changes in database tables.

7.6 Modifying Steps in a Job

There are two functions which you can use to modify job steps containing ABAP or external programs, namely:

- BAPI_XBP_ABAP_STEP_MODIFY
  to modify a job step containing an ABAP program

  and

- BAPI_XBP_JOB_EXT_STEP_MODIFY
  to modify a job step containing an external program.

7.6.1 Modifying a Job Step Containing an ABAP Program

Function name	BAPI_XBP_JOB_ABAP_STEP_MODIFY
Short description	Modify a job step containing an ABAP program
New in XBP 2.0	**New optional structures for the specification of all print and archive parameters ALLPRIPAR and ALLARCPAR. They complement the old parameters PRINT_PARAMETERS and ARCHIVE_PARAMETERS.** **In addition, two new masks have been introduced: PRINT_MASK and ARCH_MASK (see description below).**
BAPI object name	**BackgroundJob**
BAPI method name	**ModifyABAPStep**
RFC interface	function BAPI_XBP_JOB_ABAP_STEP_MODIFY importing JOBNAME like BAPIXMJOB-JOBNAME type RFC_CHAR length 32 JOBCOUNT like BAPIXMJOB-JOBCOUNT type RFC_CHAR length 8 EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER type RFC_CHAR length 16 ABAP_PROGRAM_NAME like BAPIXMREP-REPORTID type RFC_CHAR length 40 ABAP_VARIANT_NAME like BAPIXMREP-VARIANTNAM optional default SPACE type RFC_CHAR length 14 SAP_USER_NAME like BAPIXMSTEP-AUTHCKNAM optional default SY-UNAME type RFC_CHAR length 12 LANGUAGE like BAPIXMSTEP-LANGUAGE optional default SY-LANGU type RFC_CHAR length 2 PRINT_PARAMETERS structure BAPIXMPRNT 12 optional length 48 number of fields 12 ARCHIVE_PARAMETERS structure BAPIXMARCH optional default SPACE length 23 number of fields 3 STEP_NUMBER like BAPIXMJOB-STEPCOUNT type RFC_INT4 length 10 ALLPRIPAR structure BAPIPRIPAR optional default SPACE length 176 number of fields 22 ALLARCPAR structure BAPIARCPAR optional default SPACE length 328 number of fields 18 PRINT_MASK structure PRIMASK optional

	`length 22 number of fields 22`
	`ARCH_MASK structure ARCMASK optional`
	`length 18 number of fields 18`
	`exporting`
	`RETURN structure BAPIRET2 length 548`
	`number of fields 14`

Parameter (Input)	• JOBNAME is the name of a background job. • JOBCOUNT is the ID number of a job. Together with the job name, the job number identifies the job uniquely. • EXTERNAL_USER_NAME is the name of the user in the external scheduler called the function. • ABAP_PROGRAM_NAME is the name of the ABAP program that is to be executed in this job step. The program must be type 1 (interactively executable). • ABAP_VARIANT_NAME is an optional parameter to determine variants for the specified ABAP report. • SAP_USER_NAME is an optional parameter to specify an R/3 user with whose authorizations the job is processed • LANGUAGE is an optional parameter to enter the two-digit SAP language key for this job step. You can find the language keys that are allowed in table T002. • PRINT_PARAMETERS (optional): You can use this structure to transfer parameters for the printer products of the job step to the spool system. You can specify on which output device the print requests are to be printed, whether print requests should be retained in the spool system or output immediately and so on. If no structure is transferred, the system uses the print parameters in the default values of the user you used to log onto the R/3 System (rfcOpen). • ARCHIVE_PARAMETERS (optional): You use this structure to transfer parameters for archiving the printer results of the job step to the optical archiving system. You can use this parameter to specify whether or not the spool requests of the job step are to be archived. If no structure is transferred, the system uses the archiving details in the fixed values of the user you used to log onto the R/3 system (rfcOpen). • STEP_NUMBER specifies the number signifying the position of a job step in the sequence of steps in a background job that you want to modify. The second job step in a background has number 2. • ALLPRIPAR and ALLARCPAR are optional structures for the specification of all print and archive parameters and complement the old parameters PRINT_PARAMETERS and ARCHIVE_PARAMETERS. If ALLPRIPAR and ALLARCPAR are initial, PRINT_PARAMETERS and ARCHIVE_PARAMETERS are evaluated. If ALLPRIPAR and ALLARCPAR contain a value, this value is used. ALLARCPAR is a set of archive parameters used for steps consisting of an ABAP program. The ALLARCPAR parameter contains the whole set of archive parameters that can be used by an ABAP program. ALLPRIPAR is a set of print parameters used for steps consisting of an ABAP program. The ALLPRIPAR

	parameter contains the whole set of print parameters that can be used by an ABAP program. - PRINT_MASK (optional) refers to ALLPRIPAR and ARCH_MASK (optional) refers to ALLARCPAR. PRINT_MASK and ARCH_MASK were introduced for the following reasons: If, for example, a field of the parameter ALLPRIPAR is initial, it is impossible to determine, whether the caller wants to overwrite the existing value of the corresponding print parameter with the initial value or whether he wants to leave the existing value untouched. Therefore, the parameter PRINT_MASK was introduced. It contains 22 fields of the same name as the fields of ALLPRIPAR, but all are simply indicator fields of length 1. For example, PRINT_MASK-PRTXT = 'X' means that the existing value of this print parameter should be overwritten with the value in ALLPRIPAR-PRTXT. If an indicator field is blank, the corresponding print parameter remains untouched. If the parameter ALLPRIPAR is initial, the function evaluates the old parameters PRINT_PARAMETERS. If a non-initial parameter ALLPRIPAR is passed by the caller, the parameter PRINT_MASK must not be initial either (see MSG_MASK_ERROR). The above example applies also to the pair ALLARCPAR and ARCH_MASK (optional). If ALLARCPAR is initial, the function evaluates the old parameter ARCHIVE_PARAMETERS.
Parameter (Output)	BAPIRET2 is the return structure used by BAPIs.
MessageIDs	- MSG_JOBID_MISSING: You did not enter a job number. - MSG_JOBNAME_MISSING: You did not enter a job name. - MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system. - MSG_STEP_COUNT_MISSING: You did not enter the number of the step to be modified. - MSG_INVALID_STEP_COUNT: Step number invalid, that is, there is no step with the given number. - MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. - MSG_MASK_ERROR: This message refers to ALLPRIPAR and ALLARCPAR: If ALLPRIPAR or ALLARCPAR is not initial, but the corresponding mask is initial, this message is returned. - MSG_PROGRAM_MISSING: The ABAP program name is missing. - MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. - MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. - MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity

	cannot be carried out. - MSG_NO_ARCHIVE_INFO: Required fields in archive parameters are missing. - MSG_INVALID_PRINT_PARAMS: One or more print parameters are incorrect. - MSG_INVALID_ARCHIVE_PARAMS: One or more archive parameters are incorrect.

7.6.2 Modifying a Job Step Containing an External Program

<table>
<tr><th>Function name</th><th>BAPI_XBP_JOB_EXT_STEP_MODIFY</th></tr>
<tr><td>Short description</td><td>To modify a job step containing an external program.</td></tr>
<tr><td>BAPI object name</td><td><b>BackgroundJob</b></td></tr>
<tr><td>BAPI method name</td><td><b>ModifyExternalStep</b></td></tr>
<tr><td>RFC interface</td><td><pre>
function BAPI_XBP_JOB_EXT_STEP_MODIFY
importing
  JOBNAME like BAPIXMJOB-JOBNAME
     type RFC_CHAR length 32
  JOBCOUNT like BAPIXMJOB-JOBCOUNT
     type RFC_CHAR length 8
  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER
     type RFC_CHAR length 16
  EXT_PROGRAM_NAME like BAPIXMSTEP-PROGRAM
     type RFC_CHAR length 128
  EXT_PROGRAM_PARAMETERS like BAPIXMSTEP-
  PARAMETER
     optional default SPACE
     type RFC_CHAR length 255
  WAIT_FOR_TERMINATION like BAPIXMAUX-CHAR1
     optional default 'X'
     type RFC_CHAR length 1
  TARGET_HOST like BAPIXMSTEP-XPGTGTSYS
     type RFC_CHAR length 32
  SAP_USER_NAME like BAPIXMSTEP-AUTHCKNAM
     optional default SY-UNAME
     type RFC_CHAR length 12
  STEP_NUMBER like BAPIXMJOB-STEPCOUNT
     type RFC_INT4 length 4

exporting
  RETURN structure BAPIRET2 length 548
     number of fields 14
</pre></td></tr>
<tr><td>Parameter (Input)</td><td><ul>
<li>JOBNAME is the name of a job.</li>
<li>JOBCOUNT is the ID number of a job. Together with the job name, the job number identifies the job uniquely.</li>
<li>EXTERNAL_USER_NAME is the name of the user in the external scheduler who caused the function call.</li>
<li>EXT_PROGRAM_NAME is the name of the program that is to be executed by the background processing system.</li>
<li>EXT_PROGRAM_PARAMETERS is an optional parameter that may be required by the external program at runtime. The parameters are transferred to the external program at the start time as character strings.</li>
<li>WAIT_FOR_TERMINATION is an optional parameter, which has the effect that the background job waits for the external program to finish before processing the next job step (synchronous job step processing). This option is activated by default.</li>
<li>TARGET_HOST is the name of the host computer on which the external program is to be executed.</li>
<li>SAP_USER_NAME is an optional parameter to specify R/3 users with whose authorizations the job is processed.</li>
<li>STEP_NUMBER specifies the number that displays the position of a job step in the sequence of steps in a background job that you want to modify. The second job step in a background has number</li>
</ul></td></tr>
</table>

	2.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_JOBID_MISSING: You did not enter a job number. • MSG_JOBNAME_MISSING: You did not enter a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system. • MSG_STEP_COUNT_MISSING: You did not specify the number of the step to be modified • MSG_INVALID_STEP_COUNT: The step number is invalid, that is, there is no step with the given number. • MSG_PROGNAME_MISSING: The name of the external program is missing. • MSG_TARGETHOST_MISSING: The name of the target host is missing. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_CANT_LOG: Activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.7 Adding, Changing, and Deleting Job Steps via XMI

7.7.1 Adding a Step to a Job via XMI

<table>
<tr><th>Function name</th><th>BAPI_XBP_ADD_JOB_STEP</th></tr>
<tr><td>Short description</td><td>Adds and inserts a step to a job via XMI.</td></tr>
<tr><td><b>New in XBP 2.0</b></td><td><b>The entire function module is new in XBP 2.0.</b></td></tr>
<tr><td>BAPI object name</td><td><b>BackgroundJob</b></td></tr>
<tr><td>BAPI method name</td><td></td></tr>
<tr><td>RFC interface</td><td><pre>
function BAPI_XBP_ADD_JOB_STEP
importing
  JOBNAME like BAPIXMJOB-JOBNAME
     type RFC_CHAR length 32
  JOBCOUNT like BAPIXMJOB-JOBCOUNT
     type RFC_CHAR length 8
  STEP structure BPJOBSTEP
     optional default SPACE
     length 498 number of fields 17
  STEP_NUM like BAPIXMJOB-STEPCOUNT
     type RFC_INT4 length 4
  EXTERNAL_USER_NAME LIKE BAPIXMLOGR-EXTUSER
     type RFC_CHAR length 16
  ALLPRIPAR structure PRI_PARAMS
     optional default SPACE
     length 196 number of fields 24
  ALLARCPAR structure ARC_PARAMS
     optional default SPACE
     length 332 number of fields 19

Exporting
  STEP_NUMBER like BAPIXMJOB-STEPCOUNT
     type RFC_INT4 length 4
  RETURN structure BAPIRET2
     length 548 number of field 14
</pre></td></tr>
<tr><td>Parameter (Input)</td><td>• JOBNAME is the name of the job to which a job step is added.
• JOBCOUNT is the ID number of the job to which a job step is added.
• STEP is a definition of the step to be added or inserted. This parameter has a field TYP for distinguishing between types of steps. Valid values of this field are ‘A’ for ABAP program, ‘C’ for external commands, and ‘X’ for external programs.
• STEP_NUM is the position where a new step should be inserted. Use 0 (zero), if you want to add a step at the end of existing steps.
• EXTERNAL_USER_NAME is the name of the external user that is used for XMI logging.
• ALLPRIPAR (optional) is a set of print parameters, which is only used if the new step consists of an ABAP program. The ALLPRIPAR parameter contains the whole set of print parameters that may be used by an ABAP program.
• ALLARCPAR (optional) is a set of archive parameters, which is only used if the new step consists of an ABAP program. The ALLARCPAR parameter contains the whole set of archive parameters that may be used by an ABAP program.</td></tr>
</table>

Parameter (Output)	• STEP_NUMBER is the number that displays the position of the inserted or added job step. • RETURN is the return code of the function module. If no error occurs, the return code is zero.

Detailed function
Description
	This function module is for adding and inserting a step to a job. In the case of adding a step, the step will be added at the end of all job steps. In the case of inserting a step, it will be placed at the position STEP_NUM and the rest of steps will be moved down. If the job does not have any steps yet, the new step will be the first step of this job.
	This function module operates on all types of job steps: ABAP program, external command, and external program. For ABAP programs one can specify print and archive parameters. If these optional parameters are skipped, the function module uses default print and archive parameters, obtained from the function module GET_PRINT_PARAMETERS.
	Example:
	Adding or inserting an ABAP program:
	`step-typ = 'A'.`
	`step-program = '<ABAP program name>'.`
	`step-parameter = '<Variant name>'.`
	`step-language = '<Language>'.`
	`step-authcknam = '<User name>'.`
	Adding or inserting an external command:
	`step-typ = 'C'.`
	`step-program = '<Command name>'.`
	`step-opsystem = '<Operating system>'.`
	`step-parameter = '<Command parameters>'.`
	`step-xpgtgtsys = '<XPG target system>'.`
	`step-language = '<Language>'.`
	`step-authcknam = '<User name>'.`
	Adding or inserting an external program:
	`step-typ = 'X'.`
	`step-program = '<External program name>'.`
	`step-xpgtgtsys = '<XPG target system>'.`
	`step-language = '<language>'.`
	step-authcknam = '<User name>'.

MessageIDs	• MSG_JOBID_MISSING: You did not specify a job number. • MSG_JOBNAME_MISSING: You did not specify a job name. • MSG_PROBLEM_DETECTED: The R/3 job scheduling system has discovered an error. • MSG_CANT_LOG: Activity was terminated because the R/3 XMI logging mechanism returned an error. • MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling

	system. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. • MSG_WRONG_STEP_TYPE: The type of step is wrong. • MSG_NO_ARCHIVE_INFO: Required fields in archive parameters are missing. • MSG_INVALID_PRINT_PARAMS: One or more print parameters are incorrect. • MSG_INVALID_ARCHIVE_PARAMS: One or more archive parameters are incorrect. • MSG_CANNOT_GET_PRIARC_PARAMS: Retrieving new print and archive parameters failed. • MSG_CANNOT_READ_JOB: Reading information about the specified jobs failed. • MSG_CANNOT_MODIFY_JOB: Error while writing new data about the specified job into R/3 databases. • MSG_ERROR_MODIFYING_WORKTABLE: Error occurred while modifying worktable. • MSG_WRONG_STEP_NUMBER: There are no steps with the specified step number.

7.7.2 Changing and Deleting a Job Step via XMI

<table>
<tr><th>Function name</th><th>BAPI_XBP_MODIFY_JOB_STEP</th></tr>
<tr><td>Short description</td><td>Changes and deletes a step of a job via XMI.</td></tr>
<tr><td>New in XBP 2.0</td><td><b>The entire function module is new in XBP 2.0.</b></td></tr>
<tr><td>BAPI object name</td><td><b>BackgroundJob</b></td></tr>
<tr><td>BAPI method name</td><td></td></tr>
<tr><td>RFC interface</td><td><pre>
function BAPI_XBP_MODIFY_JOB_STEP
importing
  JOBNAME like BAPIXMJOB-JOBNAME
     type RFC_CHAR length 32
  JOBCOUNT like BAPIXMJOB-JOBCOUNT
     type RFC_CHAR length 8
  STEP structure BPJOBSTEP
     optional default SPACE
     length 498, number of fields 17
  STEP_NUM like BAPIXMJOB-STEPCOUNT
     type RFC_INT4 length 4
  DELETE like BAPIXMINFO-DELETESTEP
     type RFC_CHAR length 1
  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER
     type RFC_CHAR length 16
  ALLPRIPAR structure PRI_PARAMS
     optional default SPACE
     length 196 number of fields 24
  ALLARCPAR structure ARC_PARAMS
     optional default SPACE
     length 332 number of fields 19
  PRINT_MASK structure PPR_MASK
     optional default SPACE
     length 22 number of fields 22
  ARCH_MASK structure APR_MASK
     optional default SPACE
     length 18 number of fields 18

  exporting
  STEP_NUMBER LIKE BAPIXMJOB-STEPCOUNT
      type RFC_INT4 length 4
  RETURN LIKE BAPIRET2 STRUCTURE
     length 548 number of fields 14
</pre></td></tr>
<tr><td>Parameter (Input)</td><td>• JOBNAME is the name of the job whose step is to be changed or deleted.
• JOBCOUNT is the ID number of the job whose step is to be changed or deleted.
• STEP defines the new step. This parameter has a field TYP for distinguishing between types of steps. Valid values for this field are ‘A’ for ABAP program, ‘C’ for external commands, and ‘X’ for external programs. <b>The STEP parameter is not evaluated, if this function module is used for deleting a step.</b>
• STEP_NUM is the number of the step to be changed or deleted.
• DELETE is an indicator that specifies if the step should be changed or deleted. If the indicator value is initial, the step is changed.
• EXTERNAL_USER_NAME is the name of the external user that is used for XMI logging.
• ALLPRIPAR (optional) is a set of print parameters used only if the new step consists of an ABAP program. The ALLPRIPAR</td></tr>
</table>

	parameter contains the whole set of print parameters that can be used by an ABAP program. See also PRINT_MASK. • ALLARCPAR (optional) is a set of archive parameters used only, if the new step consists of an ABAP program. The ALLARCPAR parameter contains the whole set of archive parameters that can be used by an ABAP program. See also ARCH_MASK. • PRINT_MASK (optional) is a mask for print parameters. By setting a value in a mask field, you specify that there is a new field value in the ALLPRIPAR parameter. By leaving a field blank you can specify that the corresponding field should be left intact in the ALLPRIPAR parameter. PRINT_MASK refers to ALLPRIPAR and ARCH_MASK refers to ALLARCPAR. PRINT_MASK and ARCH_MASK were introduced for the following reasons: If, for example, a field of the parameter ALLPRIPAR is initial, it is impossible to determine whether the caller wants to overwrite the existing value of the corresponding print parameter with the initial value or whether he wants to leave the existing value untouched. Therefore, the parameter PRINT_MASK was introduced. It contains 22 fields of the same name as the fields of ALLPRIPAR, but all are simply indicator fields of length 1. For example, PRINT_MASK-PRTXT = ‘X’ means that the existing value of this print parameter should be overwritten with the value in ALLPRIPAR-PRTXT. If an indicator field is blank, the corresponding print parameter remains untouched. If a non-initial parameter ALLPRIPAR is passed by the caller, the parameter PRINT_MASK must not be initial either. The same applies to the pair ALLARCPAR and ARCH_MASK. If the parameter ALLPRIPAR is initial, the function evaluates the old parameters PRINT_PARAMETERS. If ALLARCPAR is initial, the function evaluates the old parameters ARCHIVE_PARAMETERS. • ARCH_MASK (optional) is a mask for archive parameters. By setting a value in a mask field, you specify that there is a new field value in the ALLARCPAR parameter. By leaving a field blank you can specify that the corresponding field should be left intact in the ALLARCPAR parameter.
Parameter (Output)	• STEP_NUMBER is the number of the changed or deleted step. • RETURN is the return code of the function module. If no error occurs, the return code is zero.
Detailed function description	This function module is for changing and deleting a step of a job. In the case of changing a step, the step does not change its number, but its content is replaced by a new one. This function module operates on all types of job steps: ABAP program, external command, and external program. For ABAP programs one can specify print and archive parameters. In the case these optional parameters are skipped, the function module uses default print and archive parameters obtained from the GET_PRINT_PARAMETERS function module. In the case of deleting a step, the step is removed and the rest of the job steps are moved up. If the job has only one step, then deleting

	fails.
	Example
	Changing step N to an ABAP program:
	step-typ = 'A'.
	step-program = '<ABAP program name>'.
	step-parameter = '<Variant name>'.
	step-language = '<Language>'.
	step-authcknam = '<User name>'.
	step_num = <N>.
	Changing step N to an external command:
	step-typ = 'C'.
	step-program = '<Command name>'.
	step-opsystem = '<Operating system>'.
	step-parameter = '<Command parameters>'.
	step-xpgtgtsys = '<XPG target system>'.
	step-language = '<Language>'.
	step-authcknam = '<User name>'.
	step_num = <N>.
	Changing step N to an external program:
	step-typ = 'X'.
	step-program = '<External program name>'.
	step-xpgtgtsys = '<XPG target system>'.
	step-language = '<language>'.
	step-authcknam = '<User name>'.
	step_num = <N>.

MessageIDs	• MSG_JOBID_MISSING: You did not specify a job number. • MSG_JOBNAME_MISSING: You did not specify a job name. • MSG_PROBLEM_DETECTED: The R/3 job scheduling system has discovered an error. • MSG_CANT_LOG: Activity was terminated because the R/3 XMI logging mechanism returned an error. • MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity

	cannot be carried out. • MSG_WRONG_STEP_TYPE: The type of step is wrong. • MSG_NO_ARCHIVE_INFO: Required fields in archive parameters are missing. • MSG_INVALID_PRINT_PARAMS: One or more print parameters are incorrect. • MSG_INVALID_ARCHIVE_PARAMS: One or more archive parameters are incorrect. • MSG_CANNOT_GET_PRIARC_PARAMS: Retrieving new print and archive parameters failed. • MSG_CANNOT_READ_JOB: Reading information about the specified jobs failed. • MSG_CANNOT_MODIFY_JOB: Error while writing new data about the specified job into R/3 databases. • MSG_ERROR_MODIFYING_WORKTABLE: Error occurred while modifying worktable. • MSG_ERROR_READING_WORKTABLE: Error occurred while reading worktable. • MSG_WRONG_STEP_NUMBER: There are no steps with the specified step number. • MSG_NO_STEP_INFO: Step modification is impossible, because no step information is provided.

7.8 Intercepting and Confirming Jobs

Interception is a new feature in XBP 2.0. Job interception means that at the moment, when the start condition of the job is fulfilled the job is set back to the status ‘scheduled’ and receives a special attribute. By calling a new XBP function the external scheduler can receive a list of all intercepted jobs and take control over such jobs.

It is not intended to subject jobs to interception in general. The user can define criteria in the new table TBCICPT1 (client, job name, job-creator including wild cards), and only the jobs that match these intercept criteria, are intercepted. For instance, a table entry (100, babu* , *) means that all jobs created in client 100 by users beginning with babu are intercepted.

Example for the use of interception:

The administrator might want to intercept all jobs of certain users or with certain job names on weekends when long-running and time critical batch jobs are executed. In this case interception provides dynamic job prioritization.

You can find detailed information on interception in *Intercepting Jobs* on page 16.

7.8.1 Getting Intercepted Jobs

In order to find out if there are intercepted jobs, the external scheduler calls a function at short intervals (BAPI_XBP_GET_INTERCEPTED_JOBS). To prevent this function from returning the same intercepted jobs again and again, the scheduler can confirm a list of intercepted jobs. Confirmation means that the scheduler informs the R/3 system that it already knows these intercepted jobs, and that a subsequent call of BAPI_XBP_GET_INTERCEPTED_JOBS does not have to return these jobs again. The confirmation of a list of intercepted jobs is done by calling BAPI_XBP_SPECIAL_CONFIRM_JOB, which is explained in the following section.

Function name	BAPI_XBP_GET_INTERCEPTED_JOBS
Short description	This function module retrieves jobs which have status INTERCEPTED. The list of returned jobs may contain all intercepted jobs, or only those ones, which have not been confirmed with one or more special confirmations.
New in XBP 2.0	**The entire function module is new in XBP 2.0.**
BAPI object name	**BackgroundJob**
BAPI method name	

RFC interface	Function BAPI_XBP_GET_INTERCEPTED_JOBS
	
	Importing
	  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER
	     type RFC_CHAR length 16
	  SELECTION type CHAR2
	     optional DEFAULT 'AL'
	     type RFC_CHAR length 2
	  CLIENT like TBTCO_AUTHCKMAN optional
	  MORE_INFO like BTCH0000-CHAR1 optional
	
	Exporting
	  RETURN structure BAPIRET2 length 548
	     number of fields 14
	
	Tables

	`JOBINFO like BPICPINFO`
	`   length 54 number of fields 4`
	`JOBINFO2 like BPICPINF1`
	`   length 120 number of fields 10`

Detailed function description	This function module retrieves jobs that have the status INTERCEPTED. It is not necessary to retrieve the same jobs again and again. To reduce the list of returned jobs to the jobs that are not known to the caller, confirmation is used. The list of returned jobs may contain all intercepted jobs, or only those jobs that have not been confirmed with one or more confirmations. To specify that confirmation should be taken into account, the optional case-insensitive SELECTION parameter is used.
Parameter (Input)	• EXTERNAL_USER_NAME: The name of the XBP user. • SELECTION (optional): This parameter specifies what kind of confirmation should be taken into account when selecting intercepted jobs. 1. ‘AL’ (default) – return all intercepted jobs regardless what confirmation they have. 2. ‘NG’ – return only those intercepted jobs that do NOT have general confirmation. 3. ‘NS’ – return only those intercepted jobs that were NOT confirmed as intercepted. 4. ‘NC’ – return only those intercepted jobs that do NOT have any confirmation. To make a special confirmation for an intercepted job, use the BAPI_XBP_SPECIAL_CONFIRM_JOB function module with CONFIRMATION = ‘i’. To make a general confirmation for an intercepted job, use the BAPI_XBP_CONFIRM_JOB function module. • CLIENT (optional): The client, on which jobs should be selected. If no client is specified, jobs from all clients are retrieved. • MORE_INFO (optional): This is an optional flag specifying the function to retrieve more detailed information on the selected jobs. When this flag is left blank, then the JOBINFO table is used; otherwise JOBINFO2
Parameter (Output)	• BAPIRET2 is the standard return structure containing return values of the function.
Tables	• JOBINFO is the table that contains intercepted jobs. • JOBINFO2 is the table that contains more detailed information on intercepted jobs.
MessageIDs	• MSG_CANT_LOG: Activity was terminated because the R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. • MSG_WRONG_CLIENT: Wrong client number • MSG_WRONG_SELECTION_PAR: Inconsistent selection parameters.

7.8.2 Confirming Jobs

There are three function modules for job selection:

- BAPI_XBP_JOB_SELECT for general job selection
- BAPI_XBP_JOB_CHILDREN_GET for child job selection
- BAPI_XBP_GET_INTERCEPTED_JOBS for the selection of intercepted jobs

These functions are normally called at intervals by the external job scheduler and return general jobs, child jobs, or intercepted jobs, respectively. If you do not want the system to return the same jobs over and over again, you can confirm them. Confirmation means that the scheduler informs the R/3 system that it already knows these jobs, and that a subsequent call of the selection function module does not have to return these jobs again.

There are two types of confirmation:

- **General:** With the general confirmation the job scheduler confirms, that it knows a job in general. Jobs are generally confirmed with BAPI_XBP_CONFIRM_JOB. When you use BAPI_XBP_JOB_SELECT the generally confirmed jobs are not returned if the corresponding indicator is set.
- **Special:** With the special confirmation the job scheduler confirms, that it knows that a job has certain characteristics. Child jobs and intercepted jobs are confirmed with BAPI_XBP_SPECIAL_CONFIRM_JOB. When you use BAPI_XBP_JOB_CHILDREN_GET or BAPI_XBP_GET_INTERCEPTED_JOBS the specially confirmed jobs are not returned again if the corresponding indicator is set.

  However, in some situations (such as after a breakdown) it might be useful to get a list of all intercepted or child jobs (including the confirmed ones). This function has a special indicator for this purpose.

7.8.2.1 Confirming Jobs Generally

Function name	BAPI_XBP_CONFIRM_JOB
Short description	This function module allows the callers to set general confirmation for a list of jobs.
New in XBP 2.0	**The entire function module is new in XBP 2.0.**
BAPI object name	BackgroundJob
BAPI method name	

RFC interface	`Function BAPI_XBP_CONFIRM_JOB`
	`Importing`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`Exporting`
	`  RETURN structure BAPIRET2 length 548`
	`     number of fields 14`
	`Tables`
	`  JOBS structure BAPIXMJOBS (jobname, jobcount)`
	`     length 40 number of fields 2`

Detailed function description	This function module allows the caller (external scheduler) to confirm jobs that are already known to it. All the jobs that were confirmed this way will not be returned again when a new call to BAPI_XBP_JOB_SELECT is performed.
Parameter (Input)	• EXTERNAL_USER_NAME: The name of the XBP user.

Parameter (Output)	• BAPIRET2 is the standard return structure containing return values of the function.
Tables	• The JOBS table contains the jobs for which general confirmation should be set. After BAPI_XBP_CONFIRM_JOB is called, the JOBS table contains only those jobs, for which this operation failed, perhaps because the job was not intercepted, the job was deleted, or an internal error occurred.
MessageIDs	• MSG_EXT_USER_MISSING: External user name is missing. • MSG_CANT_LOG: Activity was terminated because the R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. • MSG_JOB_CONFIRMATION_FAILED: Not all jobs were confirmed. The JOBS table contains the jobs that have not been confirmed.

7.8.2.2 Performing a Special Confirmation on a Job

Child jobs and intercepted jobs are confirmed with BAPI_XBP_SPECIAL_CONFIRM_JOB. When you subsequently use BAPI_XBP_JOB_CHILDREN_GET or BAPI_XBP_GET_INTERCEPTED_JOBS the specially confirmed jobs are not returned again.

However, in some situations (such as after a breakdown) it might be useful to get a list of all intercepted or child jobs (including the confirmed ones). For this purpose, this function has the special indicator CONFIRMATION.

<table>
<tr><th>Function name</th><th>BAPI_XBP_SPECIAL_CONFIRM_JOB</th></tr>
<tr><td>Short description</td><td>This function module allows the caller to set special types of confirmation for a list of jobs.</td></tr>
<tr><td><b>New in XBP 2.0</b></td><td><b>The entire function module is new in XBP 2.0.</b></td></tr>
<tr><td>BAPI object name</td><td>BackgroundJob</td></tr>
<tr><td>BAPI method name</td><td></td></tr>
<tr><td>RFC interface</td><td><pre>
  Function BAPI_XBP_SPECIAL_CONFIRM_JOB

  Importing
    EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER
       type RFC_CHAR length 16
    CONFIRMATION TYPE CHAR1

  Exporting
   RETURN structure BAPIRET2 length 548
       number of fields 14

  Tables
    JOBS structure BAPIXMJOBS (jobname, jobcount)
       length 40.number of fields 2
</pre></td></tr>
<tr><td>Detailed function description</td><td>With this function module the external scheduler confirms jobs, of which he knows certain special characteristics (parent/child, intercepted). All jobs confirmed this way will not be returned again when jobs with this characteristic are requested. To specify the type of confirmation case-insensitive parameter CONFIRMATION should be used.</td></tr>
</table>

Parameter (Input)	- EXTERNAL_USER_NAME: The name of the XBP user. - CONFIRMATION: Currently, the following types of special confirmation are available: 1. Confirmation of intercepted jobs: CONFIRMATION = ’i’. 2. Confirmation of child jobs: CONFIRMATION = ’c’. - The JOBS table contains the jobs for which confirmation should be set. After BAPI_XBP_SPECIAL_CONFIRM_JOB is called, the JOBS table contains only those jobs, for which this operation failed, perhaps because the job was deleted or an internal error occurred.
Parameter (Output)	- BAPIRET2 is the standard return structure containing return values of the function.
Tables	- JOBS is a table with jobs, for which confirmation should be set.
MessageIDs	- MSG_EXT_USER_MISSING: External user name is missing. - MSG_CANT_LOG: Activity was terminated because the R/3 XMI logging mechanism returned an error. - MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. - MSG_PROBLEM_DETECTED: Impossible to find out whether special functionality is turned on. - MSG_INTERCEPTION_INACTIVE: Interception functionality is turned off. - MSG_PARENTCHILD_INACTIVE: Parent/child functionality is turned off. - MSG_JOB_CONFIRMATION_FAILED: Not all jobs were confirmed. The JOBS table contains the jobs that have not been confirmed. - MSG_WRONG_CONFIRMATION_TYPE: The type of confirmation is wrong.

7.8.3 Modifying the Criteria Table for Interception

With the following function module the user can add/modify the table with the intercept criteria. Only jobs matching this criteria will be intercepted.

Function name	**BAPI_XBP_MODIFY_CRITERIA_TABLE**
Short description	This function module is for modifying the criteria table (TBCICPT1) by replacing or updating its contents with the contents of the TBCICPT_TABLE table and returning the current contents of the criteria table. With the APPEND indicator you can specify whether the contents of the criteria table are replaced with the new contents or the new information is appended at the end. If the table with new criteria is initial (empty) all the data from the TBCICPT1 table is deleted. When returning the content of the criteria table, the CONTENTS indicator should be set. In this case the function module returns a copy of the criteria table in TBCICPT_TABLE.
New in XBP 2.0	**The entire function module is new in XBP 2.0.**
BAPI object name	**BackgroundJob**

BAPI method name	

RFC interface	function BAPI_XBP_MODIFY_CRITERIA_TABLE
	
	importing
	  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER
	     type RFC_CHAR length 16
	  APPEND type CHAR1 DEFAULT эXэ
	  CONTENTS type CHAR1 default SPACE
	
	exporting
	  RETURN structure BAPIRET2 length 548
	     number of fields 14
	
	tables
	  TBCICPT_TABLE structure TBCICPT1
	     length 47 number of fields 3

Parameter (Input)	• EXTERNAL_USER_NAME is the name of the XBP user. • APPEND is an indicator that determines whether the new criteria should be appended ('X') or replace old content (space). • CONTENTS is an indicator that specifies that the current call is a request for the contents of the criteria table. In this case, the TBCICPT_TABLE parameter is used for returning a copy of the criteria table.
Parameter (Output)	• BAPIRET2 is the standard return structure containing return values of the function.
Table	• TBCICPT_TABLE is the table with new criteria information.
MessageIDs	• MSG_DELETE_LINE_ERROR: Error deleting a line from the TBCICPT1 table. • MSG_CANT_LOG: Activity was terminated because the R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.9 Finding, Controlling, and Modifying Job Monitor Data

Using an external job management system, you can also monitor SAP R/3 jobs, display job logs and spool lists and determine the parent/child relations. This can be done using the function modules below:

7.9.1 Determining the Status of a Job

Function name	BAPI_XBP_JOB_STATUS_GET
Short description	Determine status of a job by reading SAP R/3 information on the job.
New in XBP 2.0	**The new parameter HAS_CHILD has been added (see description below).** **In addition the function can return now the status 'Intercepted'.**
BAPI object name	**BackgroundJob**
BAPI method name	**GetStatus**

RFC interface	`function BAPI_XBP_JOB_STATUS_GET`
	`importing`
	`  JOBNAME like BAPIXMJOB-JOBNAME`
	`    type RFC_CHAR length 32`
	`  JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`    type RFC_CHAR length 8`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`    type RFC_CHAR length 16`
	
	`exporting`
	`   RETURN structure BAPIRET2 length 548`
	`     number of fields 14`
	`   STATUS like BAPIXMJOB-STATUS`
	`      type RFC_CHAR length 1`
	`   HAS_CHILD like BAPIXMINFO-HAS_CHILD`
	`      type RFC_CHAR length 1`

Parameter (Input)	• JOBNAME is the name of the job whose status is to be determined. • JOBCOUNT is the ID number of the job whose status is to be determined. • EXTERNAL_USER_NAME is the name of the user in the external scheduler who called the function.
Parameter (Output)	• STATUS is the status of a job with the following possible values: 'R' - active 'I' - intercepted 'Y' - ready 'P' - scheduled 'S' - released 'A' - cancelled 'F' - finished • BAPIRET2 is the return structure used by BAPIs. • HAS_CHILD returns the information that specifies whether a job is a child, a parent, both, or neither. This is the parent/child information. The following values are possible as parent/child-relation:

	‘P’ – job is parent/has children ‘C’ – job is child ‘B’ – (‘both’) job is parent and child ‘ ‘ - (blank) job is neither parent nor child 💡 This information might change during the runtime of a job, because children are created at runtime. Before their creation they are not known to the system.
MessageIDs	• MSG_JOBID_MISSING: You did not specify a job ID number. • MSG_JOBNAME_MISSING: You did not specify a job name. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system. • MSG_CANT_LOG: Activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. • MSG_PARENT_CHILD_INCONSISTENCY: Inconsistency in the data found concerning parent/child-relation.

7.9.2 Determining the Status of a Job List

Function name	BAPI_XBP_JOBLIST_STATUS_GET
Short description	Determine status of a list of jobs by reading SAP R/3 information on all jobs.
New in XBP 2.0	**The entire function module is new in XBP 2.0.**
BAPI object name	**BackgroundJob**
BAPI method name	**GetAllStatus**

RFC interface	`function BAPI_XBP_JOBLIST_STATUS_GET`
	`importing`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	
	`exporting`
	`  RETURN structure BAPIRET2`
	`     length 548 number of fields 14`
	
	`tables`
	`  JOBLIST structure BAPIXMINIFO`
	`    length 42 number of fields 4`
	`    (fields: jobname 32, jobcount 8, status 1,`
	`    has_child 1)`

Parameter (Input)	• EXTERNAL_USER_NAME is the name of the user in the external scheduler who caused the function call.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
Tables	• JOBLIST is a table containing the job status consisting of JOBNAME, JOBCOUNT, STATUS and HAS_CHILD. STATUS is the status of a job with the following possible values: 'R' - active ‘I’ - intercepted 'Y' - ready 'P' - scheduled 'S' - released 'A' - terminated 'F' - finished 'N' - Job does not exist Possible values for the parent/child-relation HAS_CHILD: ‘P’ – job is parent/has children ‘C’ – job is child ‘B’ – (‘both’) job is parent and child ‘ ‘ - (blank) job is neither parent nor child
MessageIDs	• MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_CANT_LOG: Activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.9.3 Reading Job Logs

Function name	BAPI_XBP_JOB_JOBLOG_READ
Short description	Get job log (also called job protocol) for a particular job. This function has been enhanced after releasing XBP 2.0. The enhanced version contains a new optional importing parameter PROT_NEW and a new optional table JOB_PROTOCOL_NEW. With the enhanced version, it is possible to read the message type of a job log entry. See also note 603919. The enhanced version has been released with the following support packages: • 46B SAPKB46B52 • 46C SAPKB46C44 • 46D SAPKB46D33 • 6.10 SAPKB61032 • 6.20 SAPKB62021
BAPI object name	**BackgroundJob**
BAPI method name	**ReadJoblog**

RFC interface	`function BAPI_XBP_JOB_JOBLOG_READ`
	`importing`
	`JOBNAME like BAPIXMJOB-JOBNAME`
	`type RFC_CHAR length 32`
	`JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`type RFC_CHAR length 8`
	`EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`type RFC_CHAR length 16`
	`PROT_NEW like BTCH0000-CHAR1 (optional)`
	`type RFC_CHAR length 1`
	`exporting`
	`RETURN structure BAPIRET2 l`
	`length 552 number of fields 14`
	`tables`
	`JOB_PROTOCOL structure BAPIXMPROT`
	`length 572 number of fields 14`
	`JOB_PROTOCOL_NEW structure TBTC5 (optional)`
	`length 573 number of fields 15`

Parameter (Input)	• JOBNAME is the name of the job whose job log should be retrieved. • JOBCOUNT is the job ID number of the job whose job log should be retrieved. • EXTERNAL_USER_NAME is the name of the user in the external scheduler who caused the function call.

	• PROT_NEW is an optional flag. If it is not set, the function behaves as before the enhancement. If the flag is set to X, the function returns the table JOB_PROTOCOL_NEW that, in addition to the old table JOB_PROTOCOL, contains also the message type. The table JOB_PROTOCOL is empty in this case.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
Tables	• JOB_PROTOCOL is the table containing the job log. • JOB_PROTOCOL_NEW is a table filled only, if the flag PROT_NEW is set to X. The table JOB_PROTOCOL_NEW contains, in addition to the old table JOB_PROTOCOL, also the message type.
MessageIDs	• MSG_JOBID_MISSING: You have not entered a job ID number. • MSG_JOBNAME_MISSING: You did not specify a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system. • MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_NO_PRIVILEGE_GIVEN: The SAP R/3 user used by the external management system to log onto the SAP R/3 system, is not authorized to read the job log. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_NO_JOB_PROTOCOL: A log does not yet exist for the specified job. • MSG_JOB_PROTOCOL_IS_EMPTY: Job log is empty. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.
Note	- The format of the job log (MsgId) has changed in SAP R/3 Release 4.0. - The term job protocol was common for what is now known as job log. Don’t get confused. It is the same thing, but the correct term is **JOBLOG**.

7.9.4 Reading the Spool List of a Job

With XBP 1.0 you had the following possibilities to read spool lists:

- BAPI_XBP_JOB_SPOOLLIST_READ for reading the spool list of a job that has been run.
- BAPI_XBP_SPOOLLST_READ_RW for reading the spool list of a job in raw format, which is needed by some job scheduling tools to format lists correctly.

These functions modules can still be used as described later in this section. However, XBP 2.0 offers you a function with which you can choose if the list should be read in raw format or not:

- BAPI_XBP_JOB_SPOOLLIST_READ_20 contains a raw format indicator.

Function name	BAPI_XBP_JOB_SPOOLLIST_READ_20
Short description	Read spool list of a job that has been run. Some job scheduling tools need the raw format to format lists correctly. By setting the corresponding indicator, you can read the spool list also in raw format.
New in XBP 2.0	**The entire function module is new in XBP 2.0. It will replace the existing function modules BAPI_XBP_SPOOLLIST_READ and BAPI_XBP_SPOOLLST_READ_RW described later in this section.**
BAPI method name	**ReadSpoollist**

RFC interface	`function BAPI_XBP_JOB_SPOOLLIST_READ`
	`importing`
	`JOBNAME like BAPIXMJOB-JOBNAME`
	`type RFC_CHAR length 32`
	`JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`type RFC_CHAR length 8`
	`EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`type RFC_CHAR length 16`
	`STEP_NUMBER like BAPIXMJOB-STEPCOUNT`
	`type RFC_INT4 length 4`
	`RAW like BTCH0000-Char1 (optional)`
	`exporting`
	`RETURN structure BAPIRET2`
	`length 548 number of fields 14`
	`tables`
	`SPOOL_LIST structure BAPIXMSPOW`
	`length 4096 number of fields 1`

Parameter (Input)	• JOBNAME is the name of the job whose spool list should be retrieved. • JOBCOUNT is the job ID number of the job whose spool list should be retrieved. • EXTERNAL_USER_NAME is the name of the user in the external scheduler who caused the function call. • STEP_NUMBER determines from which job step the spool list should be retrieved. • RAW is the raw format indicator. If this indicator is set, you can read the spool list in raw format.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
Tables	• SPOOL_LIST is the internal table containing the spool list.

MessageIDs	• MSG_JOBID_MISSING: You have not entered a job ID number. • MSG_JOBNAME_MISSING: You did not specify a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system. • MSG_JOB_DOESNT_HAVE_STEPS: Job has no steps. • MSG_INVALID_STEP_COUNT: The job does not contain a step with the given number. • MSG_STEP_COUNT_MISSING: No step number was specified. • MSG_INVALID_SPOOLID: The spool ID contained in the step is invalid. • MSG_NO_SPOOLLIST: The spool request belonging to the step does not contain an ABAP list. The spool request cannot be displayed. • MSG_EXT_USER_MISSING: The name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_PRIVILEGE_MISSING: The user used by the external management system to log onto the SAP R/3 system is not authorized to read the spool list. • MSG_NO_PRIVILEGE_GIVEN: The user used by the external management system to log onto the SAP R/3 system is not authorized to read the spool list. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

Function name	**BAPI_XBP_JOB_SPOOLLIST_READ**
Short description	Read spool list of a job that has been run. This function module is replaced by the new function module BAPI_XBP_JOB_SPOOLLIST_READ_20, with which spool lists can also be read in raw format.
BAPI method name	**ReadSpoollist**

RFC interface	`function BAPI_XBP_JOB_SPOOLLIST_READ`
	`importing`
	`  JOBNAME like BAPIXMJOB-JOBNAME`
	`     type RFC_CHAR length 32`
	`  JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`    type RFC_CHAR length 8`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`  STEP_NUMBER like BAPIXMJOB-STEPCOUNT`
	`     type RFC_INT4 length 4`
	`exporting`
	`  RETURN structure BAPIRET2`
	`     length 548 number of fields 14`
	`tables`
	`  SPOOL_LIST structure BAPIXMSPOO`
	`     length 255 number of fields 1`

Parameter (Input)	• JOBNAME is the name of the job whose spool list should be retrieved. • JOBCOUNT is the job ID number of the job whose spool list should be retrieved. • EXTERNAL_USER_NAME is the name of the user in the external scheduler who caused the function call. • STEP_NUMBER determines from which job step the spool list should be retrieved.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
Tables	• SPOOL_LIST is the internal table containing the spool list.
MessageIDs	• MSG_JOBID_MISSING: You have not entered a job ID number. • MSG_JOBNAME_MISSING: You did not specify a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system. • MSG_JOB_DOESNT_HAVE_STEPS: Job has no steps. • MSG_INVALID_STEP_COUNT: The job does not contain a step with the given number. • MSG_STEP_COUNT_MISSING: No step number was specified. • MSG_INVALID_SPOOLID: The spool ID contained in the step is invalid. • MSG_NO_SPOOLLIST: The spool request belonging to the step does not contain an ABAP list. The spool request cannot be displayed. • MSG_EXT_USER_MISSING: The name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_PRIVILEGE_MISSING: The user used by the external management system to log onto the SAP R/3 system is not authorized to read the spool list. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

Function name	**BAPI_XBP_JOB_SPOOLLST_READ_RW**
Short description	Read the spool list of a job in raw format. Some job scheduling tools need the raw format to format lists correctly. NOTE: This function module is replaced by the new function module BAPI_XBP_JOB_SPOOLLIST_READ_20, which has an indicator to switch raw format on and off.

BAPI object name	**BackgroundJob**
BAPI method name	**ReadSpoolIst**

RFC interface	`function BAPI_XBP_JOB_SPOOLLST_READ_RW`
	`importing`
	`  JOBNAME like BAPIXMJOB-JOBNAME`
	`     type RFC_CHAR length 32`
	`  JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`    type RFC_CHAR length 8`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`  STEP_NUMBER like BAPIXMJOB-STEPCOUNT`
	`     type RFC_INT4 length 4`
	`exporting`
	`  RETURN structure BAPIRET2`
	`     length 548 number of fields 14`
	`tables`
	`  SPOOL_LIST structure BAPIXMSPOO`
	`     length 255 number of fields 1`

Parameter (Input)	• JOBNAME is the name of the job whose spool list should be retrieved as raw data. • JOBCOUNT is the job ID number of the job whose spool list should be retrieved as raw data. • EXTERNAL_USER_NAME is the name of the user in the external scheduler who caused the function call. • STEP_NUMBER determines from which job step the spool list should be retrieved as raw data.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
Tables	• SPOOL_LIST is an internal table containing the spool list.
MessageIDs	• MSG_JOBID_MISSING: You have not entered a job ID number. • MSG_JOBNAME_MISSING: You have not entered a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system. • MSG_JOB_DOESNT_HAVE_STEPS: Job has no steps. • MSG_INVALID_STEP_COUNT: The job does not contain a step with the given number. • MSG_STEP_COUNT_MISSING: You have not specified a step number. • MSG_INVALID_SPOOLID: The spool ID contained in the step is invalid. • MSG_NO_SPOOLLIST: The spool request belonging to the step does not contain an ABAP list. The spool request cannot be displayed. • MSG_EXT_USER_MISSING: The name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_PRIVILEGE_MISSING: The privilege for the specified user is missing. • MSG_NO_PRIVILEGE_GIVEN: The user used by the external management system to log onto the SAP R/3 system is not authorized to read the spool list. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling

	system has discovered an error. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.9.5 Checking the Status of a Job

Problems with the SAP R/3 system (database, network, termination of background work processes) can cause discrepancies between the actual status of a job and its recorded status in the database.

Example:

A background work process, with an active job, is terminated manually. The job runtime system cannot set the job status in the database to ‘terminated’.

The function module BAPI_XBP_JOB_STATUS_CHECK recognizes these cases and corrects the job status accordingly.

Function name	**BAPI_XBP_JOB_STATUS_CHECK**
Short description	Check whether the internally displayed status is up to date.
BAPI object name	**BackgroundJob**
BAPI method name	**CheckStatus**

RFC interface	`function BAPI_XBP_JOB_STATUS_CHECK`
	`importing`
	`  JOBNAME like BAPIXMJOB-JOBNAME`
	`     type RFC_CHAR length 32`
	`  JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`    type RFC_CHAR length 8`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`exporting`
	`  ACTUAL_STATUS like BAPIXMJOB-STATUS`
	`     type RFC_CHAR length 1`
	`  RETURN structure BAPIRET2`
	`     length 548 number of fields 14`
	`  STATUS_ACCORDING_TO_DB like BAPIXMJOB-STATUS`
	`     type RFC_CHAR length 1`

Parameter (Input)	• JOBNAME is the name of the job whose status should be checked. • JOBCOUNT is the job ID number of the job whose status should be checked. • EXTERNAL_USER_NAME is the name of the user in the external scheduler who caused the function call.
Parameter (Output)	• STATUS_ACCORDING_TO_DB contains the status of a job in the database. • ACTUAL_STATUS is the actual job status with following possible values: 'R' - active 'Y' - ready 'P' - scheduled 'S' - released 'A' - terminated 'F' - finished ‘X‘ - actual status cannot be determined • BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_JOBID_MISSING: You did not specify a job ID number. • MSG_JOBNAME_MISSING: You did not specify a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP

	R/3 system. - MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. - MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. - MSG_PRIVILEGE_MISSING: The SAP user used by the external job management system to log onto the SAP R/3 system is not authorized to use this function. - MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. - MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.9.6 Selecting Jobs

Function name	BAPI_XBP_JOB_SELECT
Short description	You can use the function module BAPI_XBP_JOB_SELECT to select a set of jobs in the SAP R/3 system that match the selection criteria given. At very least the username and the job name must be partly (using wildcards) specified. From XBP 2.0 on, it is possible to reduce the job list by selecting only non-confirmed jobs (see BAPI_XBP_CONFIRM_JOB and BAPI_XBP_SPECIAL_CONFIRM_JOB).
New in XBP 2.0	**The new parameter SELECTION has been added (see description below).**
BAPI object name	**BackgroundJob**
BAPI method name	**Select**

RFC interface	`function BAPI_XBP_JOB_SELECT`
	`importing`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`  JOB_SELECT_PARAM structure BAPIXMJSEL`
	`     length 196 number of fields 18`
	`  SYSTEMID like SY-SYSID optional`
	`     type RFC_CHAR length 8`
	`  SELECTION optional type CHAR2 default 'AL'`
	
	`exporting`
	`  RETURN structure BAPIRET2`
	`     length 548 number of fields 14`
	
	`tables`
	`  SELECTED_JOBS structure BAPIXMJOBS`
	`     length 40, number of fields 2`
	`  JOB_HEAD STRUCTURE BAPIXMJOB optional`
	`   length 388 number of fields 35`

Parameter (Input)	- EXTERNAL_USER_NAME is the name of the XBP user. - JOB_SELECT_PARAM are the selection parameters. Use this structure to transfer the job selection criteria to the background processing system. - SYSTEMID is the system ID.

	• Parameter SELECTION is optional and case-insensitive, and specifies what kind of confirmation (general or child) should be taken into account. The following values are possible: 1. AL (default) returns all child jobs regardless what confirmation they have. 2. NG returns only those child jobs that do NOT have general confirmation. 3. NC returns only those child jobs that do NOT have any confirmation. To make a special confirmation for a child job, use the BAPI_XBP_SPECIAL_CONFIRM_JOB function module with CONFIRMATION='C'. To make a general confirmation for a child job, use the BAPI_XBP_CONFIRM_JOB function module.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
Tables	• SELECTED_JOBS is the list of selected jobs. • JOB_HEAD lists job headers of selected jobs.
MessageIDs	• MSG_SELECT_PARAM_MISSING: You have not specified select options correctly. • MSG_SELECT_JOBNAME_MISSING: You have not specified a job name. • MSG_SELECT_USERNAME_MISSING: You have not specified a user name. • MSG_NO_JOB_FOUND: No jobs with the given name were found. • MSG_PROBLEM_DETECTED: The SAP R/3 job system has found an error. • MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_CANT_LOG: Activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. • MSG_WRONG_SELECTION_PAR: Inconsistent selection parameters.

7.9.7 Determining the Number of Jobs With Particular Job Names

Function name	BAPI_XBP_JOB_COUNT
Short description	You can use the function module BAPI_XBP_JOB_COUNT to determine the number of jobs which are defined in the SAP R/3 system with a particular job name.
BAPI object name	**BackgroundJob**
BAPI method name	**CountByName**

RFC interface	`function BAPI_XBP_JOB_COUNT`
	`importing`
	`  JOBNAME like BAPIXMJOB-JOBNAME`
	`     type RFC_CHAR length 32`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`exporting`
	`  NUMBER_OF_JOBS like BAPIXMAUX-INT4`
	`     type RFC_INT4 length 4`
	`  RETURN structure BAPIRET2 length 548`
	`     number of fields 14`
	`tables`
	`  JOB_TABLE structure BAPIXMJOB`
	`      length 388 number of fields 35`

Parameter (Input)	• JOBNAME is the name of a background job. • EXTERNAL_USER_NAME is the name of the user in the external scheduler who caused the function call.
Parameter (Output)	• NUMBER_OF_JOBS is the number of jobs found. • JOB_TABLE lists the resources available for background processing for each instance. • BAPIRET2 is the return structure used by BAPIs.
MessageIDs	• MSG_JOBNAME_MISSING: You have not specified a job name. • MSG_NO_JOB_FOUND: No jobs with the given name were found. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has found an error. • MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_CANT_LOG: Activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.9.8 Obtaining Key Job Parameters from Job Headers and Steps

Function name	BAPI_XBP_JOB_READ
Short description	This function module is intended for obtaining key job parameters from job header and job steps.
New in XBP 2.0	**The entire function module is new in XBP 2.0.**
BAPI object name	**BackgroundJob**
BAPI method name	

RFC interface	`function BAPI_XBP_JOB_READ`
	
	`importing`
	`  JOBCOUNT like BAPIXMJOB-JOBCOUNT`
	`     type RFC_CHAR length 8`
	`  JOBNAME like BAPIXMJOB-JOBNAME`
	`     type RFC_CHAR length 32`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`  JOB_HEADER_ONLY like BAPIXMINFO-JOBHEADER`

```
         optional type RFC_CHAR length 1
      STEP_NUMBER like BAPIXMJOB-STEPCOUNT optional
         type RFC_INT4 length 4

    exporting
      RETURN structure BAPIRET2 length 548
         number of fields 14
      JOBHEAD structure BP20JOB
         length 668 number of fields 58

    tables
      STEPS structure BP20STEP optional
         length 1028 number of fields 63
```

Parameter (Input)	• JOBNAME is the name of the job whose data need to be read. • JOBCOUNT is the ID number of the job whose data need to be read. • EXTERNAL_USER_NAME is the name of the XBP user. • JOB_HEADER_ONLY (optional) is an indicator that allows a choice between two options: retrieve job header only or retrieve job header and step descriptions. • STEP_NUMBER is the number of the step whose information should be retrieved. If STEP_NUMBER is 0 (zero), then information on all steps is retrieved.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs. • JOBHEAD is the return structure containing the job header.
Tables	• STEPS is a table consisting of the requested steps of the job.
MessageIDs	• MSG_JOBNAME_MISSING : You have not specified a job name. • MSG_JOBID_MISSING : You have not entered a job ID. • MSG_EXT_USER_MISSING : External user name is missing. • MSG_CANT_LOG : Activity was terminated because the R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON : The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. • MSG_JOB_DOES_NOT_EXIST : Specified job does not exist. • MSG_JOB_NOSTEPS : There are no steps in the job. • MSG_JOB_DOESNT_HAVE_THIS_STEP : Job does not have the specified step.

7.9.9 Determining Job Children

In general, a business process that is executed by a job, or rather a collection of jobs, does not only consist of static jobs, which are known in advance, but also of jobs that are created on the fly by the static jobs, such as to dynamically distribute workload. A job that is released by another job is called a child job and the releasing job is called a parent job.

For a job scheduling system it is important to know about the existence and current status of the child jobs of a certain parent job, because in the internal logic of many applications a parent job is considered as ‘finished’ only if the parent job itself **and** its child jobs are finished.

Up to now there is no proper way for an external scheduler to find out whether or not a job has child jobs. XBP 2.0 offers functionality to find all children created by a job.

<table>
<tr><th>Function name</th><th>BAPI_XBP_JOB_CHILDREN_GET</th></tr>
<tr><td>Short description</td><td>Get all children created by a job and return them in an internal table. Only the children are returned, not the grandchildren.
Note that this function returns only the children that have already been created at the time of the call. If the function is called while the job is still running, more children might be created after the call.
A child job will be returned even if it is has been deleted before the call. Therefore, a complete history of the job’s children can be retrieved.</td></tr>
<tr><td><b>New to XBP 2.0</b></td><td><b>The entire function module is new to XBP 2.0.</b></td></tr>
<tr><td>BAPI object name</td><td><b>BackgroundJob</b></td></tr>
<tr><td>BAPI method name</td><td><b>GetChildren</b></td></tr>
<tr><td>RFC interface</td><td><pre>function BAPI_XBP_JOB_CHILDREN_GET
importing
  JOBCOUNT like BAPIXMJOB-JOBCOUNT
     type RFC_CHAR length 8
  JOBNAME like BAPIXMJOB-JOBNAME
     type RFC_CHAR length 32
  SELECTION type CHAR2 default эALэ
     type RFC_CHAR length 2
  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER
     type RFC_CHAR length 16
exporting
  NR_OF_CHILDREN
     type RFC_INT4 length 4
  RETURN structure BAPIRET2 length 548
   number of fields 14
 tables
  JOB_CHILDREN structure BAPIXMJOBS
     length 40, number of fields 2
     (jobcount,jobname)</pre></td></tr>
<tr><td>Detailed description</td><td>This function module retrieves jobs that have CHILD status. Since you do not want to retrieve the names of the same jobs over and over again, you can reduce the list of the returned jobs to the ones which are not known to the caller. For this, confirmation is used.
The list of returned jobs may contain all child jobs or only those that have not been confirmed with one or more confirmations.</td></tr>
<tr><td>Parameter (Input)</td><td>• JOBCOUNT and JOBNAME of the job whose children are to be determined.
• EXTERNAL_USER_NAME is the name of the XBP user.
• Parameter SELECTION is optional and incase-sensitive, and specifies what kind of confirmation (general or child) should be taken into account. The following values are possible:
1. AL (default) returns all child jobs regardless what confirmation</td></tr>
</table>

	they have. 2. NG returns only those child jobs that do NOT have general confirmation. 3. NS returns only those child jobs that were NOT confirmed as child jobs. 4. NC returns only those child jobs that do NOT have any confirmation. To make a special confirmation for a child job, use the BAPI_XBP_SPECIAL_CONFIRM_JOB function module with CONFIRMATION='C'. To make a general confirmation for a child job, use the BAPI_XBP_CONFIRM_JOB function module.
Parameter (Output)	• NR_OF_CHILDREN of the given job. • BAPIRET2 is the return structure used by BAPIs.
Tables	• JOB_CHILDREN: Internal table containing the jobs consisting of JOBNAME and JOBCOUNT.
MessageIDs	• MSG_JOBID_MISSING: You have not entered a job ID number. • MSG_JOBNAME_MISSING: You have not entered a job name. • MSG_JOB_DOES_NOT_EXIST: Job does not exist in the SAP R/3 system. • MSG_EXT_USER_MISSING: The name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_WRONG_SELECTION_PAR: Inconsistent selection parameters. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.9.10 Determining Parent/Child Relation

You can use the following function to get information about the parent/child relations of any job.

<table>
<tr><th>Function name</th><th>BAPI_XBP_JOB_PARENT_CHILD_INFO</th></tr>
<tr><td>Short description</td><td>This function returns the child data of a given job (see BAPI_XBP_JOB_STATUS_GET) and the following information :<ul><li>If this job is a child, job ID number and job name of the parent job are returned. This is the only function that retrieves the parent of a job.</li><li>If this job is a parent, the number of children is returned.</li></ul>Note: The child information and the number of children may change during the runtime of a job.</td></tr>
<tr><td><b>New to XBP 2.0</b></td><td><b>The entire function module is new to XBP 2.0.</b></td></tr>
<tr><td>BAPI object name</td><td><b>BackgroundJob</b></td></tr>
<tr><td>BAPI method name</td><td></td></tr>
<tr><td>RFC interface</td><td><pre>function BAPI_XBP_JOB_PARENT_CHILD_INFO
  importing
  JOBNAME like BAPIXMJOB-JOBNAME
     type RFC_CHAR length 32
  JOBCOUNT like BAPIXMJOB-JOBCOUNT
    type RFC_CHAR length 8
  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER
     type RFC_CHAR length 16

  exporting
  HAS_CHILD like BAPIXMINFO-HAS_CHILD
     type RFC_CHAR length 1
  PARENT_NAME like BAPIXMJOB-JOBNAME
     type RFC_CHAR length 32
  PARENT_COUNT like BAPIXMJOB-JOBCOUNT
     type RFC_CHAR length 8
  NR_OF_CHILDREN type INT4 length 4
  RETURN structure BAPIRET2 length 548
     number of fields 14</pre></td></tr>
<tr><td>Parameter (Input)</td><td><ul><li>JOBNAME, JOBCOUNT of the job, for which the parent/child information is to be determined.</li><li>EXTERNAL_USER_NAME is the name of the XBP user.</li></ul></td></tr>
<tr><td>Parameter (Output)</td><td><ul><li>Possible values for the parent/child-relation HAS_CHILD:
‘P’ – job is parent/has children
‘C’ – job is child
‘B’ – (‘both’) job is parent and child
‘ ‘ - (blank) job is neither parent nor child</li><li>PARENT_NAME, PARENT_COUNT of the parent job, if there is one.</li><li>NR_OF_CHILDREN of this job at the time of the call.</li><li>BAPIRET2 is the return structure used by BAPIs.</li></ul></td></tr>
</table>

MessageIDs	• MSG_JOBID_MISSING: You did not specify a job ID number. • MSG_JOBNAME_MISSING: You did not specify a job name. • MSG_PROBLEM_DETECTED: The R/3 job scheduling system has discovered an error. • MSG_CANT_LOG: Activity was terminated, because the R/3 XMI logging mechanism returned an error. • MSG_EXT_USER_MISSING: Name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. • MSG_JOB_DOES_NOT_EXIST: The job does not exist in the R/3 database.

7.9.11 Reading and Changing Intercept Status and Parent/Child Relation

Function name	BAPI_XBP_NEW_FUNC_CHECK
Short description	This function module is intended for reading and changing the status of interception and parent-child functionality.
New to XBP 2.0	The entire function module is new to XBP 2.0.
BAPI object name	BackgroundJob
BAPI method name	

RFC interface	`Function BAPI_XBP_NEW_FUNC_CHECK`
	
	`Importing parameters`
	`  INTERCEPTION_ACTION type  CHAR1 OPTIONAL`
	`  PARENTCHILD_ACTION type  CHAR1 OPTIONAL`
	`  EXTERNAL_USER_NAME like  BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	
	`Exporting parameters`
	`  INTERCEPTION TYPE CHAR1`
	`  PARENTCHILD TYPE CHAR1`
	`  RETURN structure BAPIRET2 length 548`
	`     number of fields 14`

Detailed function description	This function module is for reading and changing the status of interception and parent-child functionality. It receives a request to execute an action and returns the statuses as results for this action. A status is a value that is either INITIAL (if functionality is switched off) or 'X'. An action can be one of the following values: • 'R' or 'r' or blank for reading the current status • 'S' or 's' for setting the status ON • 'C' or 'c' for removing ON (setting the status OFF) Actions are passed with the optional parameters INTERCEPTION_ACTION and PARENTCHILD_ACTION.

Parameter (Input)	- INTERCEPTION_ACTION specifies whether the status of the interception functionality should be read, set, or cleared. See detailed function description above for choosing correct action. - PARENTCHILD_ACTION denotes whether the status of parent/child functionality should be read, set, or cleared. See detailed function description above for choosing correct action. - INTERCEPTION_ACTION and PARENTCHILD_ACTION are independent from each other. Statuses are returned via the INTERCEPTION and PARENTCHILD parameters. - EXTERNAL_USER_NAME is the name of the XBP user.
Parameter (Output)	- INTERCEPTION is the current status of interception functionality. - PARENTCHILD is the current status of parent-child functionality. - BAPIRET2 is the return structure used by BAPIs.
MessageIDs	- MSG_EXT_USER_MISSING : External user name is missing (initial). - MSG_CANT_LOG : Activity was terminated because the R/3 XMI logging mechanism returned an error. - MSG_NOT_LOGGED_ON : The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. - MSG_WRONG_ACTION appears when the name of at least one action is invalid. - MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error.

7.10 Searching with Wildcards

Using the following function modules, you can search with wildcards for ABAP reports, external commands, output devices and print formats available in the current system:

7.10.1 Searching for ABAP Reports

Function name	BAPI_XBP_REPORT_SEARCH
Short description	This function module is a value help function for ABAP reports. It returns a list of ABAP reports available in the current system, which match a certain wildcard. It is possible to return the list in several parts. These parts can also vary in size from call to call.
New in XBP 2.0	**The entire function module is new in XBP 2.0.**
BAPI object name	BackgroundJob
BAPI method name	

RFC interface	`Function BAPI_XBP_REPORT_SEARCH`
	
	`importing`
	`  REPORT like TRDIRT-NAME optional`
	`     type RFC_CHAR length 40`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`  COUNT TYPE I OPTIONAL`
	
	`exporting`
	`  RETURN structure BAPIRET2`
	`     length 548 number of fields 14`
	
	`Tables`
	`  REPORTS structure TRDIRT`
	`     length 111 number of fields 3`

Detailed function description	With this function you can get a full list of ABAP reports available in the current system, whose names correspond to a certain wildcard and whose descriptions are written in the logon language. The wildcard is given with the optional and case-insensitive parameter REPORT. Valid wildcard special symbols are '*' (asterisk) and '.' (dot). If no wildcard is specified, all reports are returned. Names of reports, languages, and text descriptions are stored in the REPORTS table. If no description for a report is found in the logon language, the entry is returned with any available non-empty description. The optional parameter COUNT is used to limit the size of the output list. This parameter makes it possible to divide the output into parts. The size of each part can vary as well. When the list is divided into parts, each call to the current function module returns the next portion of the specified size unless the output is complete. If a sequence of calls with the same environment, that is, the same REPORT and EXTERNAL_USER_NAME parameters, is interrupted by a call with different parameters, the environment for the first call is lost, and only the interrupting call is able to proceed retrieving the rest of its output.
Parameter (Input)	• REPORT is a wildcard for names of required ABAP reports. • EXTERNAL_USER_NAME is the name of the XBP user. • COUNT is an optional parameter for limiting the size of the output

	list. This parameter enables you to devide the output into parts.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
Tables	• REPORTS is a table, that contains names of all the ABAP reports that will be returned by the function module.
MessageIDs	• MSG_EXT_USER_MISSING: External user name is missing (initial). • MSG_CANT_LOG: Activity was terminated because the R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. • MSG_WRONG_COUNTER: Counter has a negative value. • MSG_SELECTION_FINISHED: All output has been retrieved.

7.10.2 Searching for External Commands

Function name	BAPI_XBP_EXT_COMM_SEARCH
Short description	This function module is a value help function for external commands. It returns a list of external commands available in the current system, whose names match a certain wildcard.
New in XBP 2.0	**The entire function module is new in XBP 2.0.**
BAPI object name	**BackgroundJob**
BAPI method name	

RFC interface	`Function BAPI_XBP_EXT_COMM_SEARCH`
	
	`Importing`
	`COMMAND LIKE SXPGCOLIST-NAME`
	`optional default э*э`
	`type RFC_CHAR.length 18`
	`OPSYS LIKE SXPGCOLIST-OPSYSTEM optional`
	`type RFC_CHAR length 10`
	`EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`type RFC_CHAR length 16`
	
	`Exporting`
	`RETURN structure   BAPIRET2 length 548`
	`number of fields 14`
	
	`Tables`
	`EXT_COMMANDS structure SXPGCOLIST`
	`length 445 number of fields 9`

Detailed function description	With this function you can get a full list of external commands available in the current system, whose names and operating systems correspond to certain requirements. The wildcard is specified in the optional and case-insensitive parameter COMMAND. The OPSYS parameter specifies the name of the operating system for which the commands are valid. If no operating system name is specified, all are assumed. If no wildcard for command names is specified, then all commands are returned. Names, operating systems, types, underlying commands, etc. of external programs are placed in the EXT_COMMANDS table.

Parameter (Input)	• COMMAND is a wildcard for required names of external commands. • EXTERNAL_USER_NAME is the name of the XBP user. • OPSYS is the name of the operating system, for which external commands are searched.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
Tables	• EXT_COMMANDS is a table that contains all external commands, that correspond to the given wildcard and the name of the operating system.
MessageIDs	• MSG_EXT_USER_MISSING: External user name is missing. • MSG_CANT_LOG: Activity was terminated because the R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. • MSG_PROBLEM_DETECTED: Internal error.

7.10.3 Searching for Output Devices

Function name	BAPI_XBP_OUTPUT_DEVICE_SEARCH
Short description	This function module is a value help function for printer output devices. It returns a list of output devices available in the current system, whose names match a certain wildcard.
New in XBP 2.0	**The entire function module is new in XBP 2.0.**
BAPI object name	**BackgroundJob**
BAPI method name	

RFC interface	`Function BAPI_XBP_OUTPUT_DEVICE_SEARCH`
	
	`Import parameters`
	`  OUTPUT_DEVICE_SHORT like TSP03L-PADEST`
	`     optional type RFC_CHAR length 4`
	`  OUTPUT_DEVICE_LONG like TSP03L-LNAME`
	`     optional type RFC_CHAR length 30`
	`  EXTERNAL_USER_NAME LIKE BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	
	`Export parameters`
	`  RETURN structure BAPIRET2 length 548`
	`     number of fields 14`
	
	`Tables`
	`  OUTPUT_DEVICES structure RSPOLD`
	`     length 92, number of fields 4`

Detailed function description	With this function you can get a full list of output devices available in the current system, whose short and long names correspond to certain requirements. Long names are non-technical names, which are first converted into short names. The names should be given using the OUTPUT_DEVICE_SHORT and OUTPUT_DEVICE_LONG parameters. Both are optional, case-

	insensitive parameters, which accept wildcards. If both names are left blank, it is assumed that the long name is '*'. Output devices and some of their properties are stored in the table OUTPUT_DEVICES.
Parameter (Input)	• OUTPUT_DEVICE_LONG is a wildcard for long names of output devices. These long names will first be converted into short names. • OUTPUT_DEVICE_SHORT is a wildcard for technical names of output devices. • EXTERNAL_USER_NAME is the name of the XBP user.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
Tables	• OUTPUT_DEVICES is a table that contains all output devices that correspond to the given wildcards for short and long names.
MessageIDs	• MSG_EXT_USER_MISSING: External user name is missing. • MSG_CANT_LOG: Activity was terminated because the R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.10.4 Searching for Print Formats

Function name	**BAPI_XBP_PRINT_FORMAT_SEARCH**
Short description	This function module is a value help function for print formats. It returns a list of print formats available for a certain printer. The names of the print formats match a certain wildcard.
New in XBP 2.0	**The entire function module is new in XBP 2.0.**
BAPI object name	**BackgroundJob**
BAPI method name	

RFC interface	`Function BAPI_XBP_PRINT_FORMAT_SEARCH`
	
	`Importing`
	`  PRINTER like TSP03L-PADEST`
	`     type RFC_CHAR length 4`
	`  LAYOUT like RSPOLD-LAYOUT`
	`     optional DEFAULT э*э`
	`     type RFC_CHAR length 16`
	`  EXTERNAL_USER_NAME LIKE BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	
	`Exporting`
	`  RETURN structure BAPIRET2`
	`     length 548 number of fields 14`
	
	`Tables`
	`  LAYOUTS structure RSPOLD`
	`     length 92, number of fields 4`

Parameter (Input)	• PRINTER specifies a certain printer, for which print formats are retrieved.

	• LAYOUT defines a wildcard for required print formats. If this optional parameter is left blank, ‘*’ is assumed. • EXTERNAL_USER_NAME is the name of the XBP user.
Parameter (Output)	• RETURN is the return structure for function modules used for BAPIs.
Tables	• LAYOUTS is a table that contains all the print formats that were requested.
MessageIDs	• MSG_EXT_USER_MISSING: External user name is missing. • MSG_CANT_LOG: Activity was terminated because the R/3 XMI logging mechanism returned an error. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out. • MSG_WRONG_PRINTER_NAME: There is no printer of the given name in the system.

7.11 General Help Functions

Below are descriptions of the following functions which allow you to:

- see all variants of a given ABAP program (BAPI_XBP_VARIANT_INFO_GET).
- see the resources currently available for jobs in the SAP R/3 system (BAPI_XBP_ GET_CURR_BP_RESOURCES).
- determine whether resources are available for a job on a particular server at a particular time (BAPI_XBP_ GET_BP_SRVRES_ON_DATE).
- determine whether resources are available for a job on any server in the SAP R/3 system at a particular time (BAPI_XBP_GET_BP_RESRC_ON_DATE).
- read syslog entries in the SAP R/3 system.

7.11.1 Showing All Defined Variants of an ABAP Program

Function name	BAPI_XBP_VARIANT_INFO_GET
Short description	For a given ABAP the variants are read.
BAPI object name	**BackgroundJob**
BAPI method name	**GetVariantListForReportname**

RFC interface	`function BAPI_XBP_VARIANT_INFO_GET`
	`importing`
	`  ABAP_PROGRAM_NAME like BAPIXMREP-REPORTID`
	`     type RFC_CHAR length 40`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`  VARIANT_SELECT_OPTION like BAPIXMREP-`
	`  VARSELOPT`
	`     type RFC_CHAR length 1`
	`exporting`
	`  RETURN structure BAPIRET2 length 548`
	`    number of fields 14`
	`tables`
	`  ABAP_VARIANT_TABLE structure BAPIXMVAR`
	`    length 54 number of fields 2`

Parameter (Input)	• ABAP_PROGRAM_NAME is name of the ABAP program for which existing variants are to be determined. The program must have type 1 (interactively executable). • EXTERNAL_USER_NAME is the name of the XBP user • VARIANT_SELECT_OPTION defines which type of variables should be chosen. Permissible values are as follows: 'A' - variants that are available for batch as well as for dialog will be selected 'B' - 'batch only' variants will be selected
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
Tables	• ABAP_VARIANT_TABLE contains all defined variants of the ABAP program.

MessageIDs	• MSG_PROGRAM_DOES_NOT_EXIST: The specified ABAP program does not exist in the SAP R/3 system. • MSG_PROGNAME_MISSING: You did not enter an ABAP program name. • MSG_INVALID_SELECT_OPTION: Wrong code used to select variants of an ABAP. • MSG_PROGRAM_HAS_NO_VARIANT: The ABAP program has no variants. • MSG_NO_VARIANTS_DEFINED: You have not yet defined any variants. • MSG_PROG_NOT_EXECUTABLE: The program is not executable. • MSG_NO_EXECUTE_PRIVILEGE: The SAP user used by the external job management system to log onto the SAP R/3 system is not authorized to execute the ABAP program. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_EXT_USER_MISSING: The name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.11.2 Determining Current Resources for Jobs in the SAP R/3 System

Function name	BAPI_XBP_GET_CURR_BP_RESOURCES
Short description	Using this function module you can retrieve the following values from the SAP R/3 system: • Names of the servers which currently have background work processes • Number of background work processes on each server and their status (total number, free, working, class A background work processes)
BAPI object name	**SystemServiceInfo**
BAPI method name	**GetCurrentBackgroundResources**

RFC interface	`function BAPI_XBP_GET_CURR_BP_RESOURCES`
	`importing`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`exporting`
	`  RETURN structure BAPIRET2 length 552`
	`     number of fields 14`
	`tables`
	`  RESOURCE_INFO structure BAPIXMCRES`
	`     length 69 number of fields 5`

Parameter (Input)	• EXTERNAL_USER_NAME is the name of the XBP user.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs.
Parameter (Output)	• RESOURCE_INFO is a table containing the following information in each line: • Server name • Host name • Total number of background work processes on the server • Number of free background work processes on the server • Number of working background work processes on the server • Number of reserved class A background work processes on the server
MessageIDs	• MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered an error. • MSG_NO_RESOURCES_FOUND: There are no background processing resources in the SAP R/3 system. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_EXT_USER_MISSING: The name of the external user is missing. This is the name of a user in the external job scheduling system. • MSG_NOT_LOGGED_ON: The external management tool has not

	logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

7.11.3 Checking Available Job Resources at a Particular Time on a Server

In the SAP R/3 system, within the framework of switching operation modes on servers, you can assign different work process types at different times. For example, you might only have dialog processes during the day, but at night some of these switch to operating as background work processes. You can use the function module BAPI_XBP_GET_BP_SRVRES_ON_DATE to determine whether background work processes are available at a particular time **on a particular server**.

Function name	BAPI_XBP_GET_BP_SRVRES_ON_DATE
Short description	Get resource information for a particular server at a certain date and time.
BAPI object name	**SystemServiceInfo**
BAPI method name	**GetBgrdResourcesOnDateOnServer**

RFC interface	`function BAPI_XBP_GET_BP_SRVRES_ON_DATE`
	`importing`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR-EXTUSER`
	`     type RFC_CHAR length 16`
	`  SERVER_NAME like BAPIXMJOB-EXECSERVER`
	`     type RFC_CHAR length 20`
	`  DATE like BAPIXMJOB-SDLSTRTDT`
	`     type RFC_DATE length 8`
	`  TIME like BAPIXMJOB-SDLSTRTTM`
	`     type RFC_TIME length 6`
	`exporting`
	`  RESOURCE_INFO structure BAPIXMRES`
	`      length 66 number of fields 4`
	`  RETURN structure BAPIRET2 length 548`
	`      number of fields 14`

Parameter (Input)	• EXTERNAL_USER_NAME is the name of the XBP user. • SERVER_NAME is the name of the R/3 instance on which the availability of resources is to be determined. The name must be specified in the format <host_name> <SAP_system_name>_<SAP_system_number>. The instance name is contained in system profile parameter rdisp/myname. Example: host1234_C11_55 • DATE is the date on which the availability of resources is to be determined. You must specify the date in the format YYYYMMDD. Example 20000101. • TIME is the time of day at which the availability of resources is to be determined. The time must be specified as HHMMSS. Example: 231255.
Parameter (Output)	• BAPIRET2 is the return structure used by BAPIs. • RESOURCE_INFO is a table that lists the resources available for background processing for each instance: • Host name • Number of background work processes • Number of reserved class A background work processes
MessageIDs	• MSG_NO_RESOURCES_FOUND: There are no background work

	processes on the server at the given time and date. • MSG_PROBLEM_DETECTED: The SAP R/3 job scheduling system has discovered a problem. • MSG_INVALID_SERVER_NAME: The specified server name is invalid. • MSG_INVALID_DATE_TIME: The specified date and/or time is invalid. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging mechanism returned an error. • MSG_EXT_USER_MISSING: The name of the external user is missing. This is the name of a user in an external job scheduling system. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

The XMI log does not record any calls to this function, since it does not change or output any security-sensitive data.

7.11.4 Checking Available Job Resources at a Particular Time in the Whole SAP System.

Function name	BAPI_XBP_GET_BP_RESRC_ON_DATE
Short description	You can use this function module to determine whether background work processes are available at a particular time **on any server** in the SAP R/3 system.
BAPI object name	**SystemServiceInfo**
BAPI method name	**GetBackgroundResourcesOnDate**

RFC interface	`function BAPI_XBP_GET_BP_RESRC_ON_DATE`
	`importing`
	`  EXTERNAL_USER_NAME like BAPIXMLOGR_EXTUSER`
	`    type RFC_CHAR length 16`
	`  DATE like BAPIXMJOB_SDLSTRTDT`
	`    type RFC_DATE length 8`
	`  TIME like BAPIXMJOB_SDLSTRTTM`
	`    type RFC_TIME length 6`
	`exporting`
	`  RETURN structure BAPIRET2 length 548`
	`     number of fields 14`
	`tables`
	`  RESOURCE_INFO_TBL structure BAPIXMRES`
	`     length 66 number of fields 4`

Parameter (Input)	• EXTERNAL_USER_NAME is the name of the XBP user. • DATE is the date on which the availability of resources is to be determined. You must specify the date in the format YYYYMMDD. Example 20000101. • TIME is the time of day at which the availability of resources is to be determined. The time must be specified as HHMMSS. Example: 231255.
Parameter (Output)	• BAPIRET2 is the return used by BAPIs.
Tables	• RESOURCE_INFO_TBL is a table that lists the resources available for background processing for each instance. It contains the following values: - Server name - Host name - Number of background work processes - Number of reserved class A background work processes
MessageIDs	• MSG_NO_RESOURCES_FOUND: There are no background work processes in the SAP R/3 system at the given time and date. • MSG_INVALID_DATE_TIME: The specified time and/ or date is invalid. • MSG_PROBLEM_DETECTED: The job scheduling system has discovered an error. • MSG_CANT_LOG: The activity was terminated because the SAP R/3 XMI logging system returned an error. • MSG_EXT_USER_MISSING: The name of the external user is missing. This is the name of a user in the external job scheduling

	system. • MSG_NOT_LOGGED_ON: The external management tool has not logged onto the CCMS XMI interface. Therefore, the activity cannot be carried out.

8. Appendix

8.1.1 BAPI Return Structure

Each XBP function module has a RET structure of type BAPIRET2 as export parameter. With the help of this structure messages are reported from the SAP system to the caller.

STRUCTURE	BAPIRET2	
Field name	Type	Short description
TYPE	CHAR 1	Messagetyp: S Success, E Error ...
ID	CHAR 20	Message-ID
NUMBER	NUMC 3	Message-Number
MESSAGE	CHAR 220	Message Text
LOG_NO	CHAR 20	Appication-Log: Protocol Number
LOG_MSG_NO	NUMC 6	Application-Log: actual Number
MESSAGE_V1	CHAR 50	Message-Variable
MESSAGE_V2	CHAR 50	Message-Variable
MESSAGE_V3	CHAR 50	Message-Variable
MESSAGE_V4	CHAR 50	Message-Variable
PARAMETER	CHAR 32	Parameter Name
ROW	INT4 10	Line in Parameter
FIELD	CHAR 30	Field in Parameter
SYSTEM	CHAR 10	Logical System; origin of message
Notes	–	This structure is used for all BAPIs and is not related to XBP directly
Related to	–	There have been older structures with different names such as: BAPIRETURN1
	–	–

8.1.2 Message IDs and Their Meaning

In the table below you find a list of the message numbers as they are in the transaction SE91and the corresponding aliases that are used in the XBP functions described in chapter 7. Note that only messages from the message class XM are used.

You will be able to see the actual text of the message once you analyze the BAPIRET2 values. For your convenience we include here the constants which are referenced in the function module descriptions.

xmi_messages	like sy-msgid	value 'XM',
msg_logon_gen	like sy-msgno	value '9',
msg_logon	like sy-msgno	value '010'

msg_logoff_gen	like sy-msgno	value '011'
msg_logoff	like sy-msgno	value '012'
msg_auditlevel_set	like sy-msgno	value '013'
msg_versions_get_gen	like sy-msgno	value '014'
msg_versions_get	like sy-msgno	value '015'
msg_version_check	like sy-msgno	value '016'
msg_interface_describe	like sy-msgno	value '017'
msg_logmsg_enter	like sy-msgno	value '018'
msg_log_select	like sy-msgno	value '019'
msg_message_formats_upload	like sy-msgno	value '020'
msg_already_logged_on_gen	like sy-msgno	value '021',
msg_already_logged_on	like sy-msgno	value '022',
msg_unknown_interface	like sy-msgno	value '023',
msg_unknown_version	like sy-msgno	value '024',
msg_logon_denied_gen	like sy-msgno	value '025',
msg_logon_denied	like sy-msgno	value '026',
msg_not_logged_on_gen	like sy-msgno	value '027',
msg_not_logged_on	like sy-msgno	value '028',
msg_invalid_range	like sy-msgno	value '029',
msg_cant_select	like sy-msgno	value '030',
msg_cant_log	like sy-msgno	value '031',
msg_cant_upload	like sy-msgno	value '032',
msg_invalid_parameters	like sy-msgno	value '033',
msg_problem_detected	like sy-msgno	value '034',
msg_reorg	like sy-msgno	value '035',
msg_reorg_gen	like sy-msgno	value '037',
msg_jobname_missing	like sy-msgno	value '046',
msg_jobid_missing	like sy-msgno	value '047',
msg_ext_user_missing	like sy-msgno	value '048',
msg_job_does_not_exist	like sy-msgno	value '049',
msg_progname_missing	like sy-msgno	value '050',
msg_no_archive_info	like sy-msgno	value ‘051’
msg_invalid_print_params	like sy-msgno	value '052',
msg_invalid_archive_params	like sy-msgno	value '053',
msg_no_release_privilege	like sy-msgno	value '054',
msg_job_not_active	like sy-msgno	value '055',
msg_no_abort_privilege	like sy-msgno	value '056',
msg_no_job_found	like sy-msgno	value '057',
msg_targethost_missing	like sy-msgno	value ‘058’

msg_no_jobsteps	like sy-msgno	value '059',
msg_no_job_protocol	like sy-msgno	value '060',
msg_empty_job_protocol	like sy-msgno	value '061',
msg_step_count_missing	like sy-msgno	value '062',
msg_no_spoollist	like sy-msgno	value '063',
msg_privilege_missing	like sy-msgno	value '064',
msg_invalid_spoolid	like sy-msgno	value '065',
msg_no_immediate_start_poss	like sy-msgno	value '066',
msg_no_resources_found	like sy-msgno	value '067',
msg_invalid_date_time	like sy-msgno	value '068',
msg_invalid_server_name	like sy-msgno	value '069',
msg_prog_has_no_variant	like sy-msgno	value '070',
msg_prog_does_not_exist	like sy-msgno	value '071',
msg_no_execute_privilege	like sy-msgno	value '072',
msg_prog_not_executable	like sy-msgno	value '073',
msg_no_variants_defined	like sy-msgno	value '074'
msg_invalid_select_option	like sy-msgno	value '075'
msg_select_param_missing	like sy-msgno	value '076'
msg_trace_before_call	like sy-msgno	value '077'
msg_select_jobname_missing	like sy-msgno	value '078'
msg_select_username_missing	like sy-msgno	value '079'
msg_cant_del_in_jobtable	like sy-msgno	value '080'
msg_cant_del_joblog	like sy-msgno	value '081’
msg_problem_pred_succ	like sy-msgno	value ‘082’
msg_commit_failed	like sy-msgno	value ‘083’
msg_no_delete_privilege	like sy-msgno	value ‘084’
msg_job_running	like sy-msgno	value ‘085’
msg_interface_reorg	like sy-msgno	value '086'
msg_interface_reorg_gen	like sy-msgno	value '087'
msg_parent_child_inconsistency	like sy-msgno	value '088'
msg_child_register_error	like sy-msgno	value '089'
msg_mask_error	like sy-msgno	value '090'
msg_param_missing	like sy-msgno	value '091'
msg_event_does_not_exist	like sy-msgno	value '092'
msg_event_raise_failed	like sy-msgno	value '093'
msg_job_confirmation_failed	like sy-msgno	value '094'
msg_wrong_confirmation_type	like sy-msgno	value '095'
msg_wrong_selection_par	like sy-msgno	value '096'
msg_parentchild_inactive	like sy-msgno	value '097'

msg_interception_inactive	like sy-msgno	value '098'
msg_wrong_counter	like sy-msgno	value '099'
msg_wrong_printer_name	like sy-msgno	value '100'
msg_selection_finished	like sy-msgno	value '101'
**********		
msg_cant_enq_job	like sy-msgno	value '194'
msg_cant_read_jobdata	like sy-msgno	value '195'
msg_cant_release_job	like sy-msgno	value '196'
msg_cant_set_jobstatus_in_db	like sy-msgno	value '197'
msg_cant_start_job_immediately	like sy-msgno	value '198'
msg_cant_update_jobdata	like sy-msgno	value '199'
msg_eventcnt_generation_error	like sy-msgno	value '200'
msg_invalid_dialog_type	like sy-msgno	value '201'
msg_invalid_new_jobdata	like sy-msgno	value '202'
msg_invalid_new_jobstatus	like sy-msgno	value '203'
msg_invalid_startdate	like sy-msgno	value '204'
msg_job_edit_failed	like sy-msgno	value '205'
msg_job_modify_canceled	like sy-msgno	value '206'
msg_job_not_modifiable_anymore	like sy-msgno	value '207'
msg_nothing_to_do	like sy-msgno	value '208'
msg_no_batch_on_target_host	like sy-msgno	value '209'
msg_no_batch_server_found	like sy-msgno	value '210'
msg_no_batch_wp_for_jobclass	like sy-msgno	value '211'
msg_no_modify_privilege_given	like sy-msgno	value '212'
msg_no_release_privilege_given	like sy-msgno	value '213'
msg_no_startdate_no_release	like sy-msgno	value '214'
msg_invalid_targetgroup	like sy-msgno	value '216'
msg_conflicting_targets	like sy-msgno	value '217'
msg_job_doesnt_have_steps	like sy-msgno	value '218'
msg_wrong_step_type	like sy-msgno	value '219'
msg_job_doesnt_have_this_step	like sy-msgno	value '220'
msg_cannot_get_priarc_params	like sy-msgno	value '221'
msg_cannot_read_job	like sy-msgno	value '222'
msg_cannot_modify_job	like sy-msgno	value '223'
msg_wrong_step_number	like sy-msgno	value '224'
msg_error_modifying_worktable	like sy-msgno	value '225'
msg_job_nosteps	like sy-msgno	value '227'
msg_jobcount_missing	like sy-msgno	value '228'
msg_invalid_target	like sy-msgno	value '229'

msg_error_reading_worktable	like sy-msgno	value '230'
msg_delete_line_error	like sy-msgno	value '231'
msg_no_step_info	like sy-msgno	value '232'
msg_wrong_action	like sy-msgno	value '233'
msg_no_change_authority	like sy-msgno	value ‘234’
msg_invalid_jobclass	like sy-msgno	value ‘235’
msg_wrong_client	like sy-msgno	value ‘236’

9. INDEX