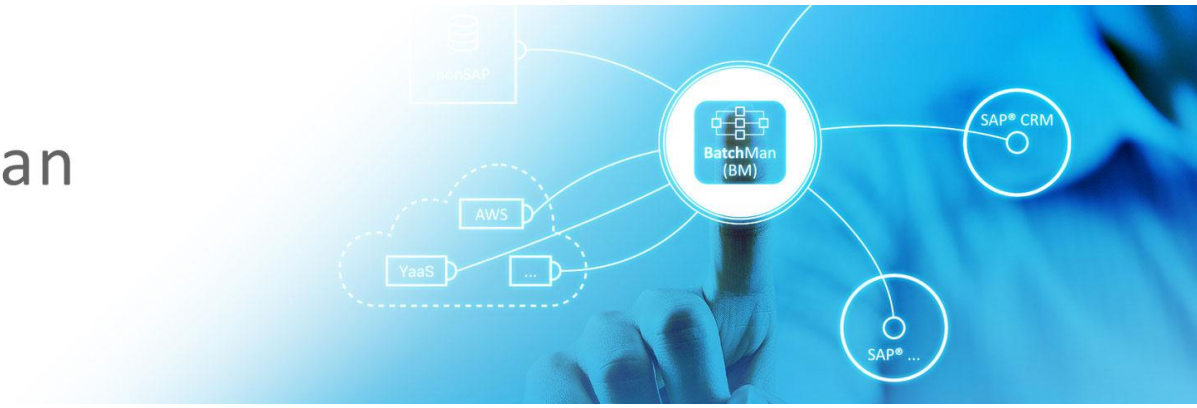
BatchMan



Automation & Monitoring of Business Processes in SAP & nonSAP

User Manual

BatchMan

Release 5

Product Info & Support

Phone: + 49 (0) 40/32 80 86 - 0

E-Mail: batchman@honico.com

Web: www.honico.com

Release: BatchMan 5.2

Date: 17.04.2026

SAP® Certified
Integration with SAP S/4HANA®

SAP® Certified
Powered by SAP NetWeaver®

HONICO

HONICO Systems GmbH | Mattentwiete 8 | 20457 Hamburg | Germany
Tel: +49 (0) 40/32 80 86 - 0 | Fax: +49 (0) 40/32 80 86 - 58 | www.honico.com | info@honico.com

Table of Contents

1. Introduction

This User Manual provides an overview of functions and handling of **BatchMan**, the SAP integrated process automation software for SAP and nonSAP processes.

Following a short description of the software, the Dynpros are displayed according to the menu and explanations are provided for fields and functions.

Knowledge of the SAP R/3®Standard system functions is a prerequisite and not explained further. Meaning and usage of these functions can be looked up in the SAP Documentation.

1.1. BatchMan: Functionality

In SAP, it is not possible to process all activities in the dialog mode. Performance-intensive processes should run better in the batch mode. SAP Job Management only provides limited options to control a large number of jobs.

BatchMan provides you with those familiar comfortable functions from the host world of job control systems for the SAP R/3 Client/Server environment.

A graphical interface for definition and clear display of job dependencies and the planning of operator orders also enable the management of complex job schedules with complicated structures and differentiated dependencies.

In the library, job steps, jobs and job networks are defined once and separated from the SAP environment by separate data retention. Flexible combinations of job steps for jobs and of jobs for job networks are possible.

Using extensive library functions, it is possible to carry out research and evaluations. Via a release procedure, it is possible to transfer jobs and networks directly to the SAP environment.

Using the job scheduling via an order, it is possible to define various order types such as individual or automatic orders.

An automatic generation of suggestions for job scheduling is possible. You may choose from an event control, file control or time control.

Job monitoring ensures a job control, in which it is possible to intervene in the background processing of the SAP system at any time. These functions may also be executed remotely.

1.2. Own name space

BatchMan was developed in SAP name space /BTCMAN/. This guarantees that overlapping with any customized client developments can not occur.

1.3. BatchMan: Basic Concept

The following outline provides an overview about the basic operation methods of BatchMan.

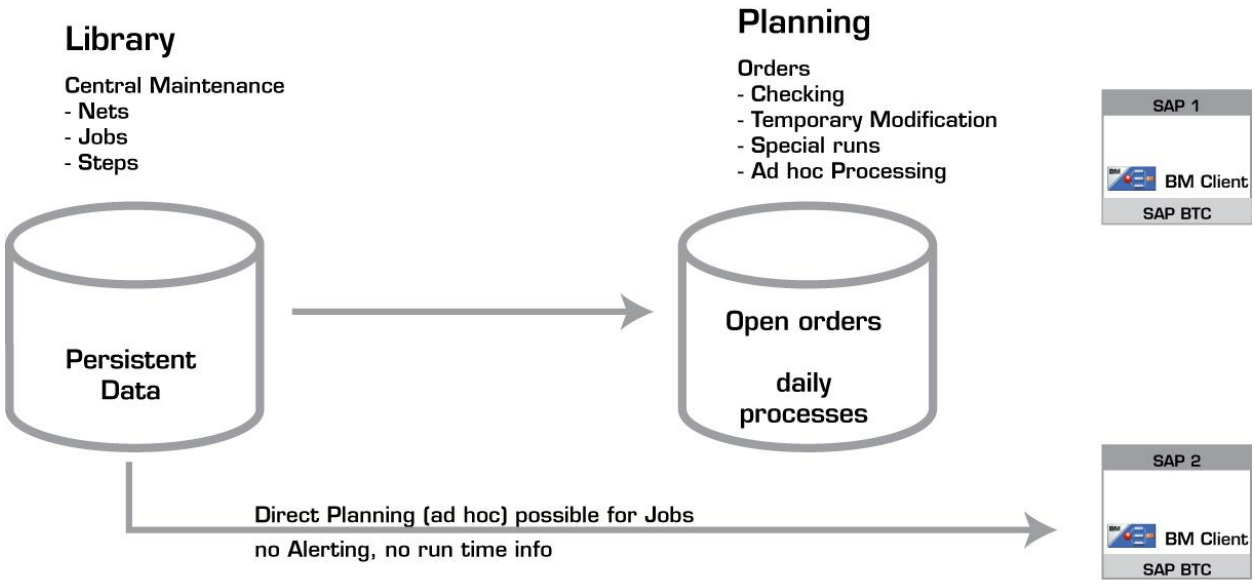
The master data objects (step, job and networks) are defined in the **library.**

Steps – A job step represents the lowest hierarchy level within the job management. A step corresponds to an ABAP-program (in most cases with a **variant** [a variant determines the parameter of a report]), an external program or an external command. A **BatchMan** step essentially corresponds to the step also used in SAP.

Job – A job represents the second hierarchy level and consists of a sequence of steps. Besides the SAP-available fields ‚job class' and ‚target host', **BatchMan** offers a number of additional fields that allow you to build up a very convenient job monitoring. Jobs can directly be transferred for processing from the **BatchMan** library to the SAP job control for processing. In this case, however, **BatchMan** functionality is not available. In addition to a job (described as normal job), processes are also scheduled in **BatchMan.** These processes only have one process step. A step list does not exist.

Job- Net – A net represents the highest hierarchy level of the master data and consists of a sequence of net-like (sequential and/or parallel) connected jobs and also networks them. In SAP, a corresponding object is not offered. In **BatchMan** however, jobs and networks can be summed up logically this way.

Job-Networks can not be transferred directly to the SAP job processing.



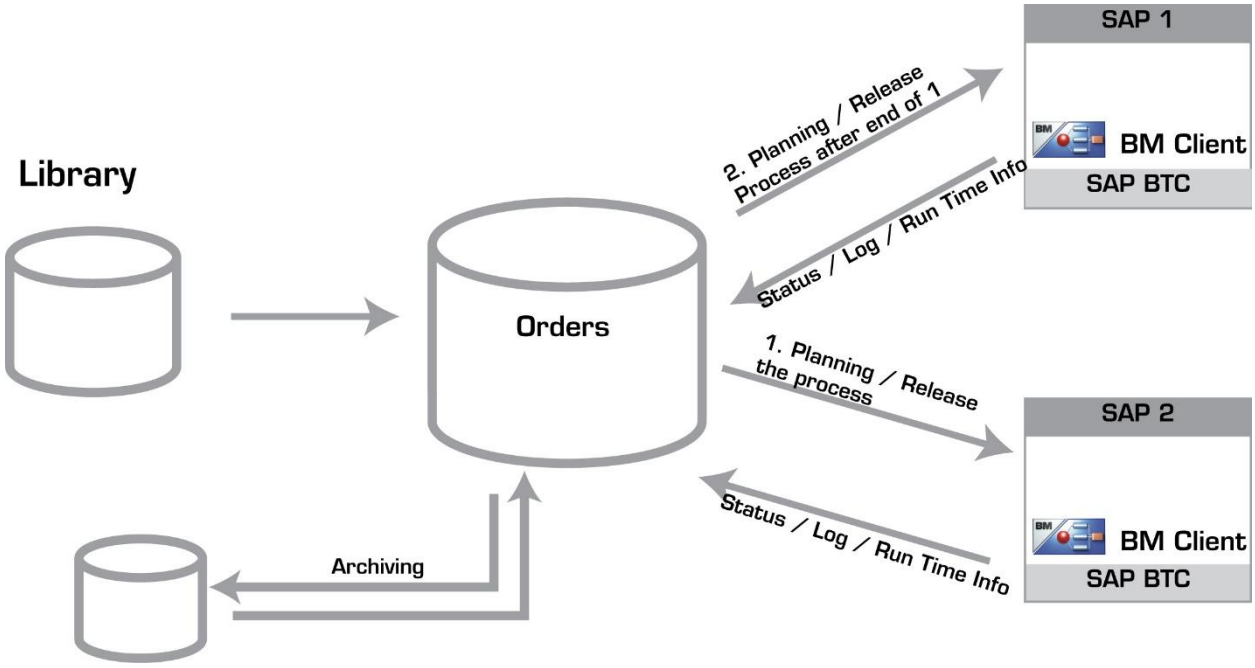
After creating the master data, the executable orders (**scheduling**) are generated by copying the corresponding master data objects. These orders are called **ORDERS**. This **BatchMan** term describes the formation of manually or automatically combined work packages for the processing. These work packages consist of sequences of jobs and job nets.

You can undertake manual modification in an order without changing the master data objects in the library, which means that the change is only temporarily valid for this order. If the master data objects are used in a different order, the original master data is copied into the new order.

After **Checking** and **Releasing** an order, the order objects are transferred into the SAP job control for processing.

While the order is processed, results of the SAP job processing are transferred to the BatchMan job control, so that all results are available in BatchMan. There is no need to switch back and forth between BatchMan and SAP.

At this point, **Monitoring** of the processes starts. List- or graphics-based monitoring is available, which offers an array of functions. The status of job and network processing can be monitored, aborted jobs can be re-started, and other activities be initiated.



After an order has been processed completely, the order itself can be archived including all objects; and also the results of the order (job log, running time). The **Archiving** within **BatchMan** is being done with an appropriate (own) **BatchMan** report.

1.4. BatchMan Master Job

In SAP, different scheduler types control the periodic scheduling and, particularly, the starting of the jobs.

Since user exits do not exist for these schedulers, a so-called **Master Job** is scheduled in **BatchMan.** This master job runs periodically in the SAP background scheduling. If this master job does not run, you can still create manual orders, but many functions of **BatchMan** are not executable anymore.

BatchMan – Master job tasks:

- Check order status of the active orders
- Start the daily order (Order of the type STD)
- Scheduling periodic orders that lasts less than 24 hours
- Start file-dependent jobs
- Check run time excess of jobs
- Start recovery jobs
- Send mail if job is aborted
- Check jobs outside their time window
- Check milestones
- Check resources
- Check AUTO orders generated from WDH orders
- Check jobs with status ‘READY’ and alarm if necessary
- Update watchdog-entry

1.5. BatchMan Scheduler Job

Since there are no User exits for schedulers, a so-called Master-scheduler (short: scheduler) is scheduled. This scheduler runs periodically (once a minute) in the SAP background scheduling. If this scheduler does not run, processing is not continued.

The scheduler calls up a function module that takes over the actual scheduling of the processing via the interface (XBP, BW.SCHE,BM-own).

1.6. Important terms

The following terms have special importance for the understanding of **BatchMan.**

LIBRARY

The master data of the objects step, job/process and job network are managed in the library.

JOB STEP

A job step is part of the library and represents the lowest hierarchy level within the job management system. A job step matches a program with its variant, an external program or command.

PROCESS

A process defines a workflow (group of steps or jobs). It is part of the library and consists of one or more processing steps. It is possible to define different processing types and (often also) sub types.

JOB

A job is part of the library and consists of a sequence of steps. A job is the process type ‘normal job’.

JOB- NET (also: NET or NETWORK)

A job network is part of the library and consists of a sequence of linked jobs.

VARIANT

A variant determines the flow parameter of a report.

ORDER

The term “order” in the sense of job management refers to the formation of manually or automatically compiled work packages to carry out a job scheduling. These work packages consist of sequences of jobs and job networks.

REFERENCE DATE AND PERIODICITY (Job and Job Net)

The scheduled date of a job or a job network is based on the reference date and periodicity information of the job or a job network.

This periodicity is always in reference to the reference date.

These specifications are relevant when jobs or job networks from the automatic scheduling should be based on an order of the type ‘standard’.

SCHEDULED DATE (Job and Job Net)

The scheduled date of a job or a job network is based on the reference date and the periodicity. If jobs and job networks are based on the automatic scheduling, their scheduled dates are compared to the scheduled day of the order of the order type ‘standard’. If both scheduled dates are the same, the corresponding job or job network will be added to the proposal list of an order.

START CRITERIA (Job and Job Net)

Jobs can be started by certain start criteria, which are

Checkpoints (mile stones) - they can be changed manually or through a job. They represent a numerical value (integer).

Resources - the job is started if the resource is available.

File-dependency - the job is started as soon as a file arrives.

TVARV (dynamic SAP-Variable) - can be used as date placeholder.

Earliest/Latest Start time - jobs should start within a certain time window.

SCHEDULED DAY (Operator Order)

The scheduled day of an order of the order type ‘standard’ is compared to the scheduled dates of the jobs and job networks that are subject to the automatic scheduling. If both scheduled dates are the same, the appropriate job or job network will be added to the proposal list of an order.

If orders of the type ‘individual order’ or ‘repeat order’ are present, the scheduled day serves as selection criteria.

SCHEDULING WITHOUT START DATE

This function can schedule a job in the SAP batch system without specifying a start date.

RELEASING WITH START DATE

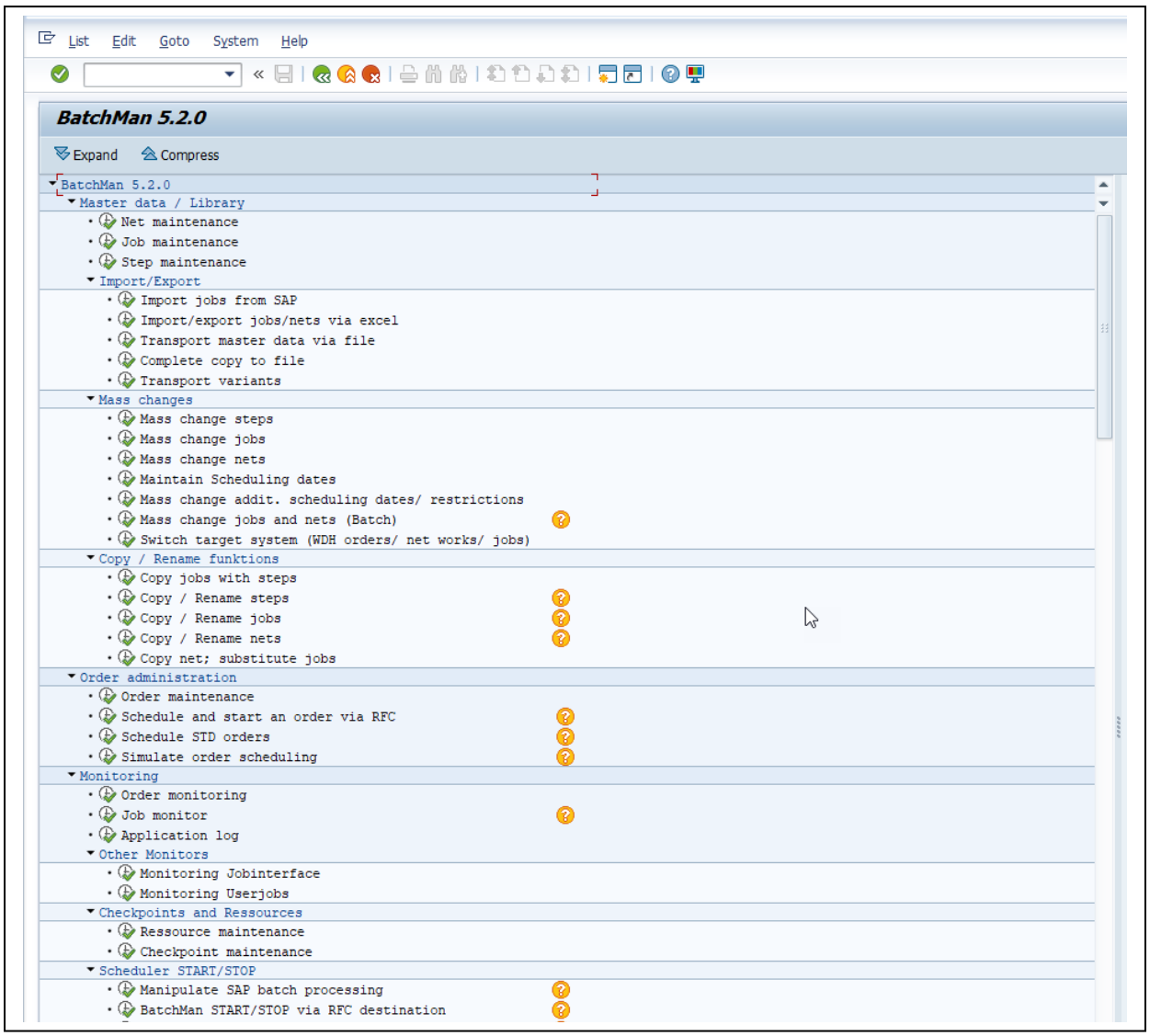
This function allows scheduling of a job in the SAP batch system by specifying a start date. Once this start date is reached, the job is started.

START DATE (Job, Operating Order)

The start date is the execution date of a job or order in the SAP batch system.

1.7. BatchMan Function tree vs. Cockpit

Transaction **J5HEN** will call up the **BatchMan** function tree

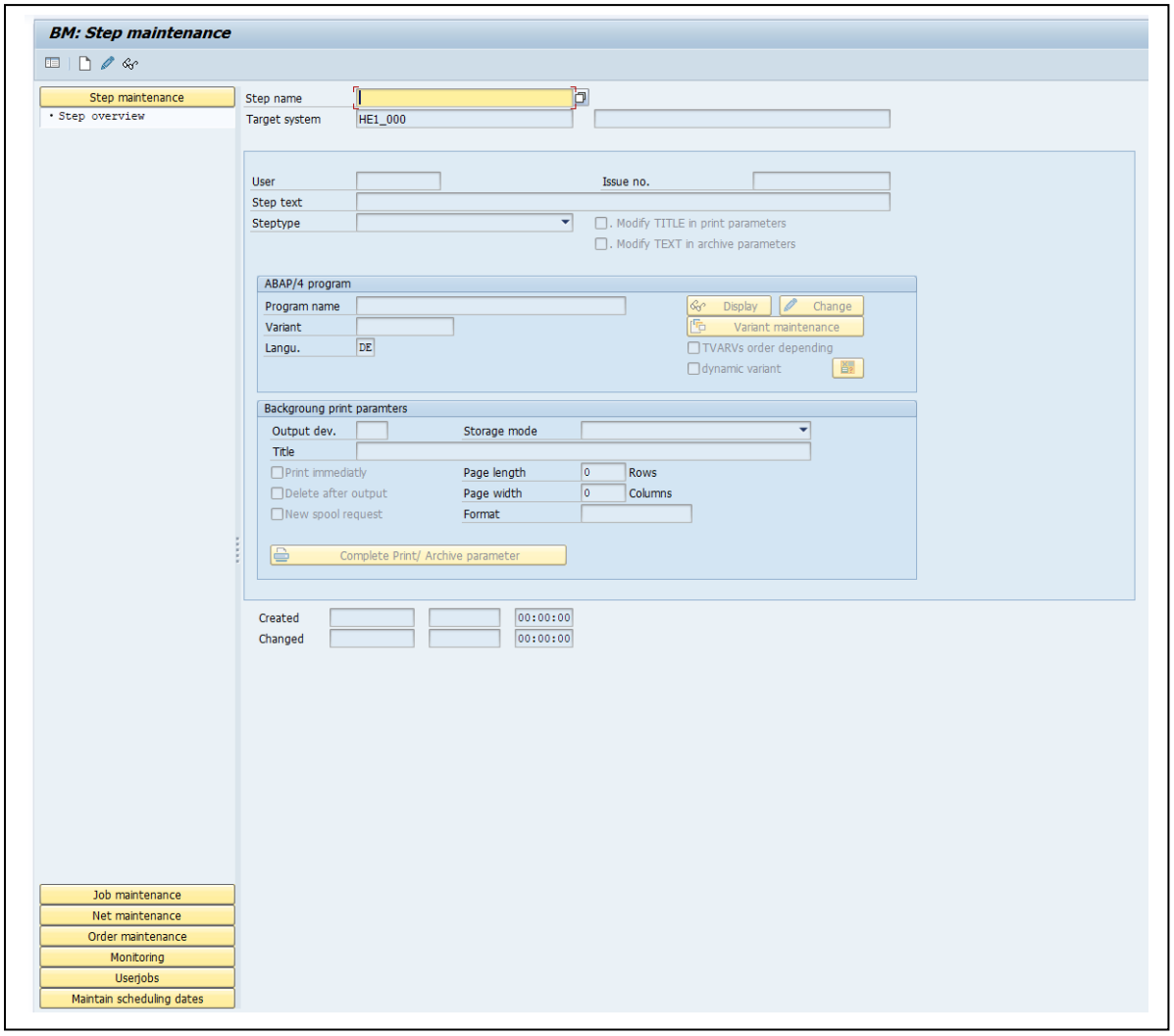


Roles are included in **BatchMan** authorization transports. If these roles are assigned to a user, the authorizations defined in the roles will appear as user menu.

A cockpit is provided for the most important transactions. While in the function tree, all functions can be called, only the most common functions can be called in the cockpit.

The cockpit is also available on the client systems. There it is the only option to maintain master data and to call monitoring.

Call the Cockpit with transaction **J5HJN**



Using this button, you can show or hide the navigation bar on the left side.

1.8. BatchMan System Information

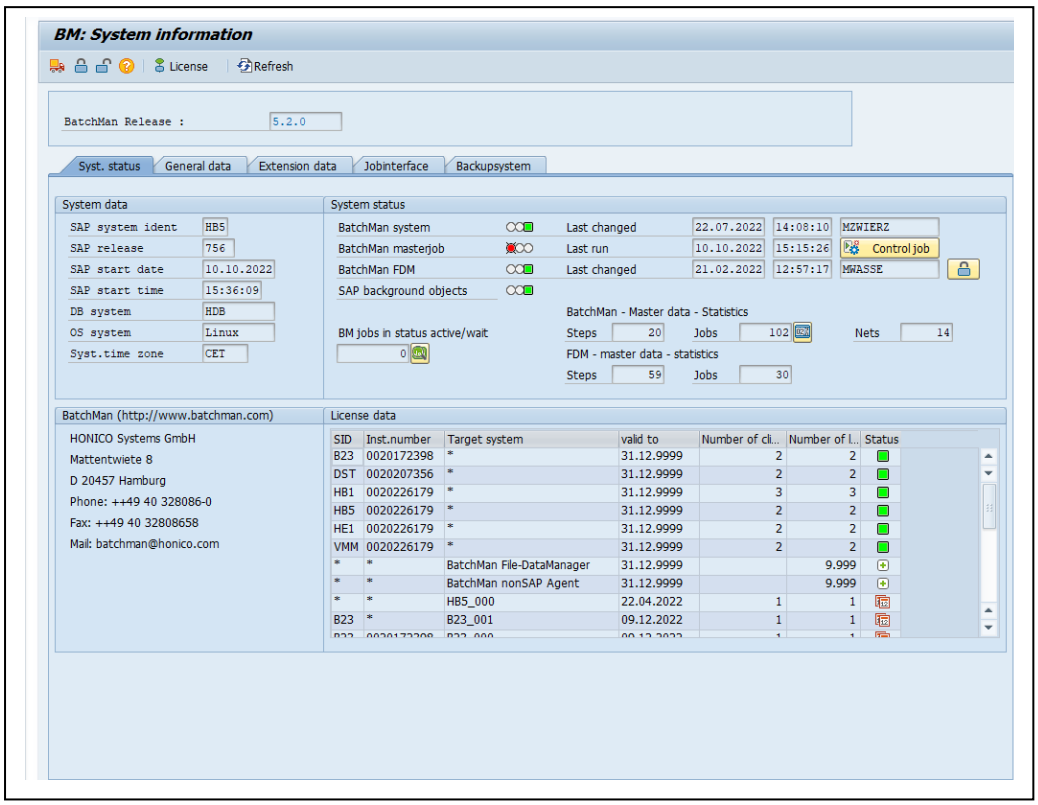
Pushing the system info button in the BatchMan user interface or entering the transaction **J5HIN** in the command field will call up system information for BatchMan.

First section on the left: general system data of the customer system.

First section on the right: BatchMan information is displayed. System status is shown as traffic light.

The upper traffic light shows the status of BatchMan. The BatchMan system can be switched off independently from the SAP-Batch system.

With the Button *Lock closed* all further activities in BatchMan are stopped. However, already started jobs will still be executed. With *Lock open* you can activate BatchMan again.



The middle traffic light shows the status of the **BatchMan** **master job**. It is set through a query of the last run of the master job. If this run was more than 10 minutes ago, the traffic light will be set on red. You can refresh the display via the green check mark (Enter-key).

The lower traffic light shows the status of the **SAP-scheduler**. It can be switched off via SAP Standard (call transaction SM61) or in the **BatchMan** function menu under Monitoring → Scheduler START/STOP. However, already started jobs will still be executed.

BM jobs in the active/ wait status: Number of BatchMan jobs that are “active” or are in the “wait” status. Usage for switch-over scenario if maintenance is due on the BatchMan master system.

Master data statistics: Number of created steps, jobs and job networks as master data

Left (section) below: Our **address** (development team). For error messages and suggestions/ideas, please use the *Mail-Address*.

On the right section below you can find your **BatchMan** license data with your **SAP Installation Number**. (see Installation Manual for more detailed information).

This button displays installed **transport orders** from table E070. Selection is done via transport orders with A* and , “**BatchMan**“

2. Library

2.1. Introduction

The master data of the objects job step, job and job network are managed in the library.

A job step is equivalent to a program or a variant of a program and represents the lowest hierarchy level within the job management system. Job steps can contain ABAP/4-programs and external commands just like in the SAP-Standard.

A job consists of a sequence of steps.

A job network consists of a sequence of network-linked jobs.

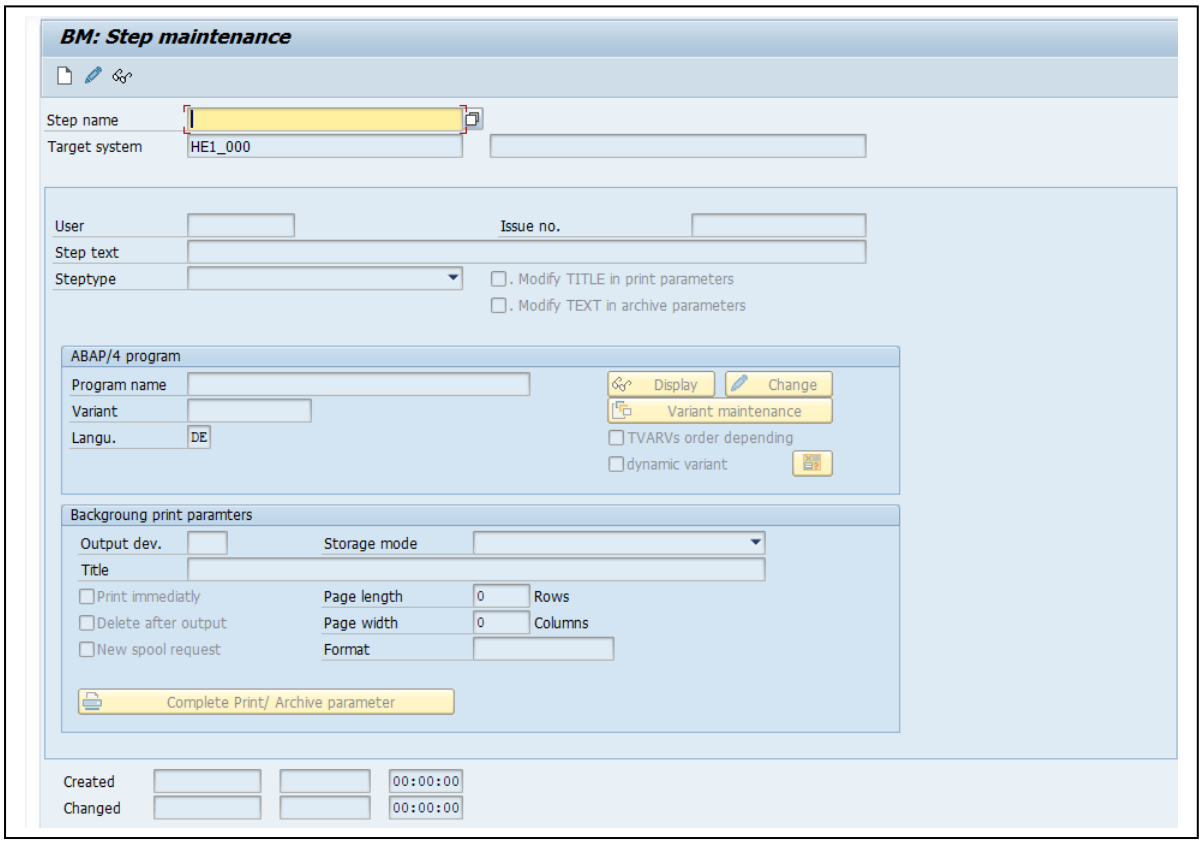
Flexible handling of the job management system is warranted by combining job steps to jobs and jobs to job networks.

The dataset in the library is kept separate from the dataset in the SAP environment. With the help of a special release process, the dataset in the library is transferred to the SAP batch system. This guarantees that changes in library objects, contrary to SAP standard, do not immediately become active.

2.2. Administration: Job Steps

The term job step is used in **BatchMan**, just like in the SAP standard, for the to-be-scheduled programs with their variants, external commands, or programs. Within the job management, job steps represent the lowest hierarchy level.

Step Maintenance can be called up in two different ways: directly with the transaction J5HCN or via the function tree: Master data/Library → Step maintenance.



The target system in the step maintenance serves to call up the search help, and to validate the values when steps are stored. It is a mandatory field.

The target system is not entered/stored in the step. This will make the steps available in several jobs (multiple usage).

The target system is stored according to user specification. The field is not maintained when the step maintenance is accessed for the first time. A Popup appears, and the user has to select a target system. When the step maintenance is used subsequently, then the last used target system will be read/appear in the user parameter.

The target system can be changed although it is grey (generally shows that it is not ready for input). One option is calling up F4 help in the field, a second option is available through the menu: Additions → Change RFC destinations

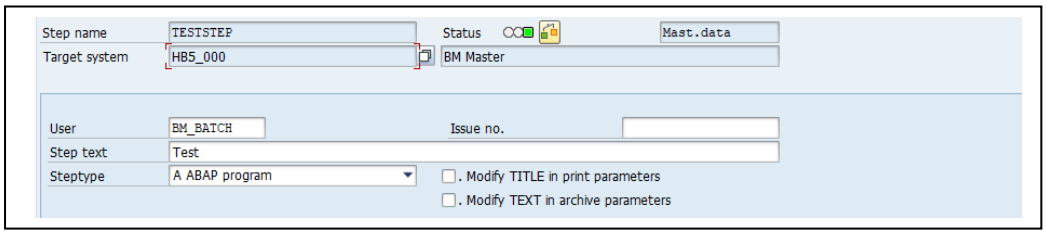
After entering the **step name** (also: Step ID), the desired function can be selected:


Create a new step


Change/ Edit an existing step


Display existing steps (also: ENTER key)

The individual steps are explained using a test step. Step name ‘Test step’ and button create. Afterwards the additional fields are displayed according to step type. Individual fields can only be entered when other fields were filled (example: print parameter).



User is the SAP user who is authorized to process the job step. That means that this user must also have the SAP authorization to execute, for example, an ABAP report.

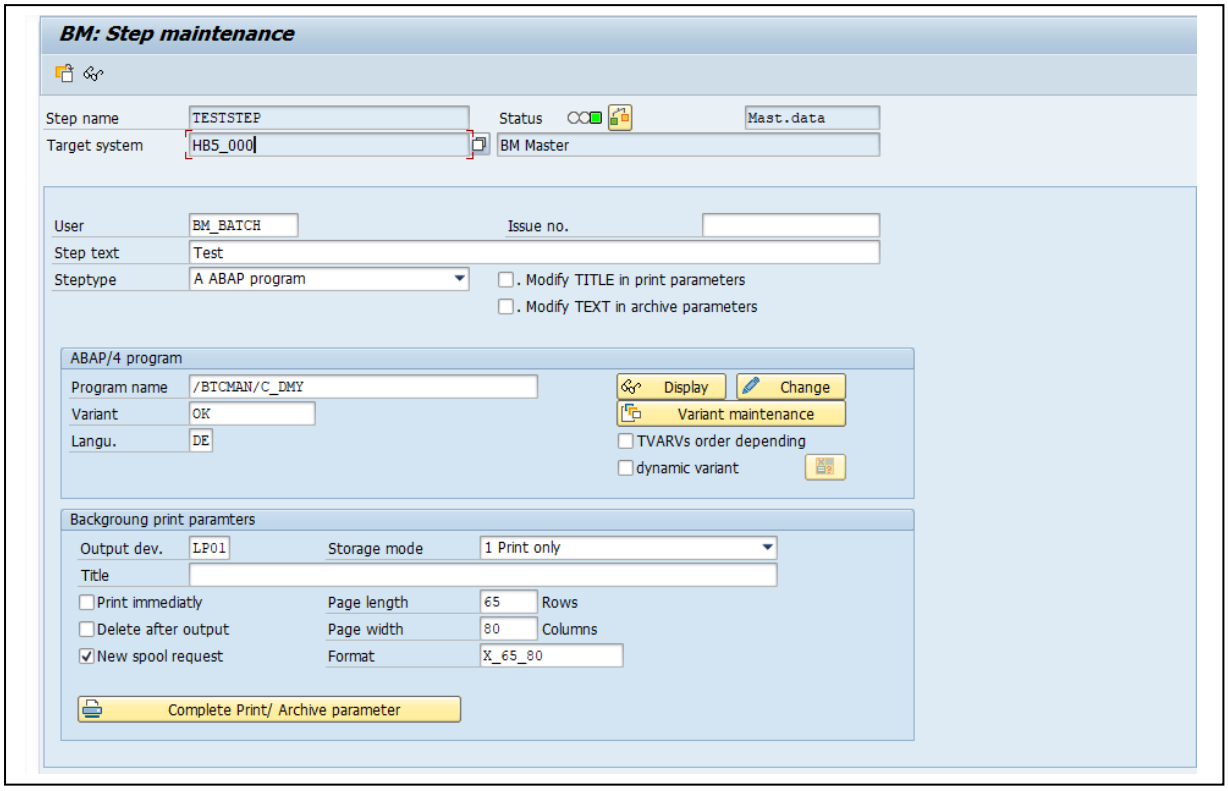
Analog to the target system, this user is taken from the user parameters. If no value is deposited there, the user from the setup will be used.

The **step text** is a long text to describe the text.

The **ISSUE number** is an unspecified field. Sort criteria of other software can be deposited here, like for example of HelpDesk systems.

Via the list box **Step type**, the type of step can be selected.

2.2.1. Step type: ABAP Programm



The **report name** is the name of an ABAP/4 program within a step, which should run in background. With the buttons 'Display' and 'Change' you can jump directly in the SAP maintenace of the program (if you have the authorization).

The **variant** (of the report name) determines the parameter that should be used for processing the report. Not all ABAP/4 programs need a variant. A variant only has to be entered if the report needs selection specifications.

The background processing takes the specifications from the variant, which normally would be entered interactively. In case of a direct call of the program in SAP, these parameters will be entered in the selection screen.

After entering the variant and ENTER, the button *'Variant maintenance'* becomes usable. This way you can go directly to the variant maintenance of the report. The target system determines in which SAP system and client the maintenance is called up.

The **TVARVs Order Dependent** flag can be used for variants that contain TVARVs that are updated by the BM Rule Editor. TVARVs, which normally have a global scope, are then becoming net or order specific.

Example: A TVARV is set in 3 different nets and then used -> each of these 3 nets only uses the value set in its own net (or order), even if this deviates from the global TVARV value. If an object is repeated later, the TVARV values originally set in this net (or order) also apply, even if the TVARV has already been updated to other values in the meantime.

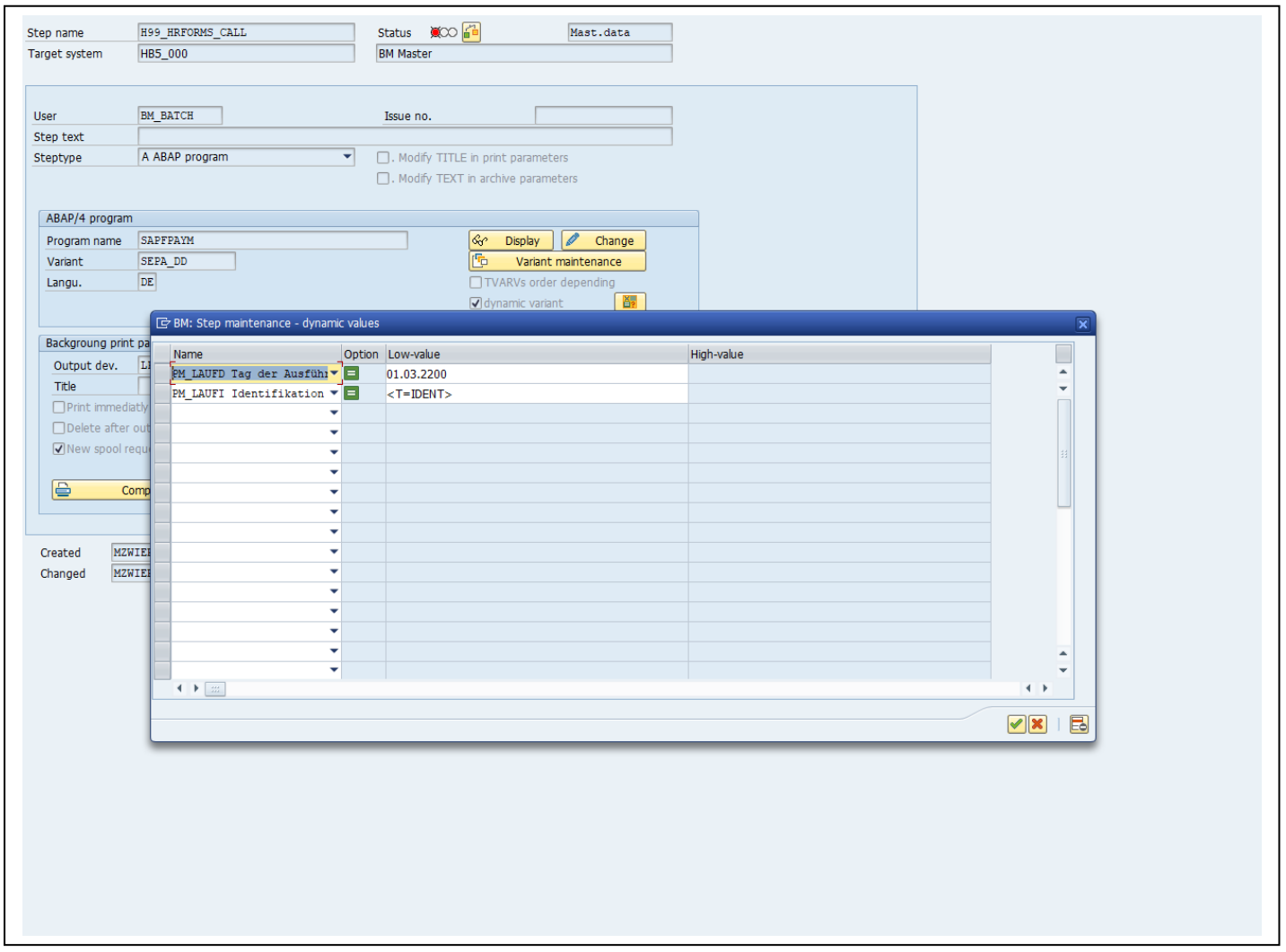
Dynamic Variant

By using **dynamic variants**, individual variant values can be set or overwritten (if an existing variant is specified at the same time) .

This has the advantage that dynamic values can be used without having to enter them in the variant or using the SAP variant-internal dynamization.

Dynamic variant values also have the advantage that they can be transported or exported/imported as components of a BM step and therefore do not require any additional variant transport. The BM Excel import/export offers a convenient way to change variant values on a large scale without having to laboriously adapt the actual variants in the SAP standard.

Note: Dynamic variants are set at the start of the job, with the variable values that existed at the time the job started. The TVARV updater should therefore not run within the same job in which the values used within the dynamic variants are set.



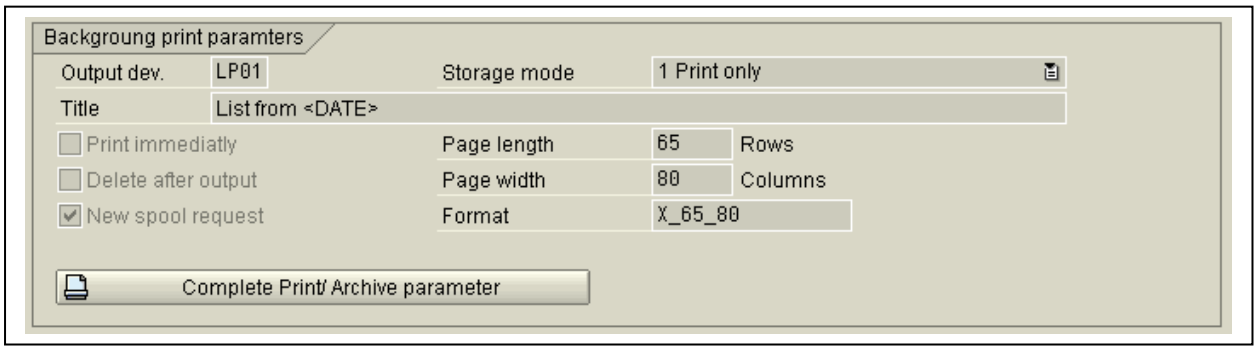
The **language** specifies in which language

- texts are displayed
- texts are entered
- system printout is created.

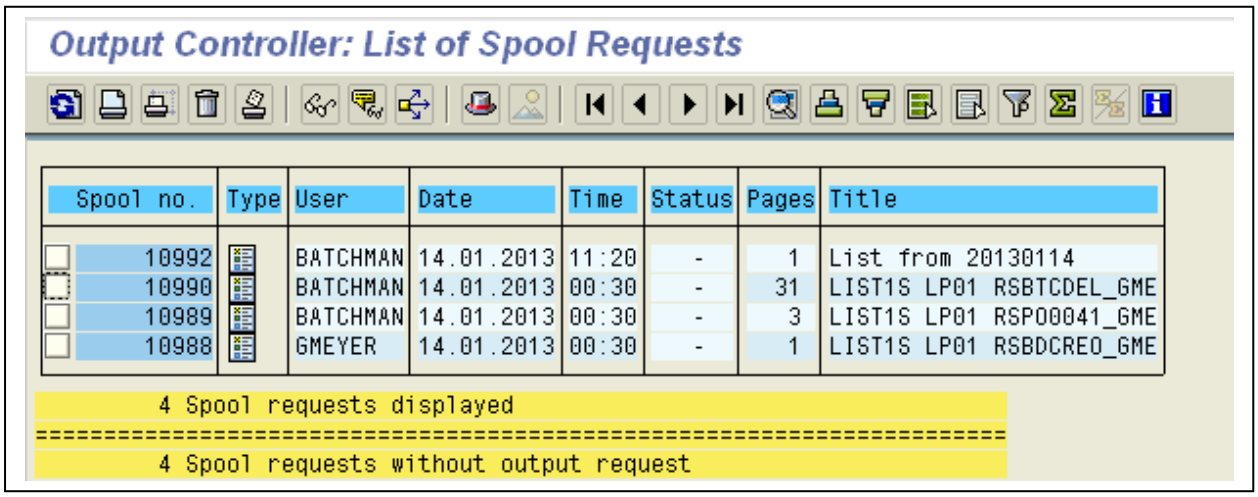
After entering the report name, the fields for print parameters become ready for input as well. These are already set up with a default value for some parameters. These parameters are already determined through user values (e.g. printer) or report values like report page and report width.

If additional print parameters and/or archive parameters shall be maintained, this is possible via the button *'Complete print/archive parameter'*. The fields for print-and archive parameters have the same meaning as in SAP.

You can also modify the title in the print parameters and the text in the archive parameters with variable fields.



For scheduling, the parameter is replaced and it then has in transaction SP01 the following title:



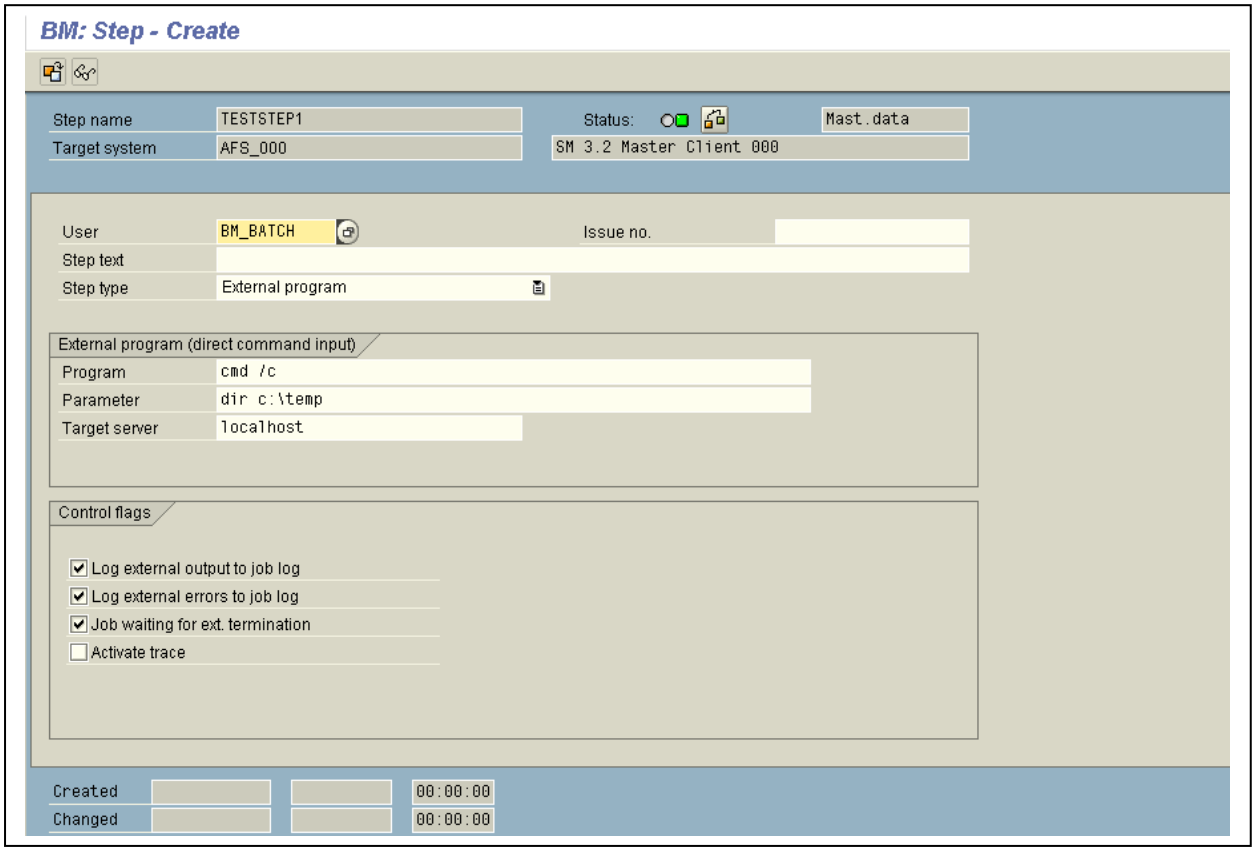
The displaying to happen in SAP internal date format !

2.2.2. Step type: Query

Obsolete, please no longer use.

2.2.3. Step type: External Program

This step type is used for the direct execution of operating system programs.



Program: Name of the program that should be executed by the background processing. All types of executable files are admitted.

You either have to enter the complete path name of the programs or you have to make sure that the program is in the search path of the SAP user. Per default this is the user <SID>adm, who is not a user of the SAP system, but the one defined in the host operating system.

Example:
Specification of a complete path name is: /usr/sap/C11/SYS/exe/run/r3trans. Options and parameter of the external programs can be entered in the field **Parameter**. If one specification contains a blank, then it must be set in ‘ ‘apostrophes.

Beispiel: - l 4 : = two values
'- l 4' : = one value
Hinweis: The character that must be used to set a value in apostrophes is operating system-dependent.

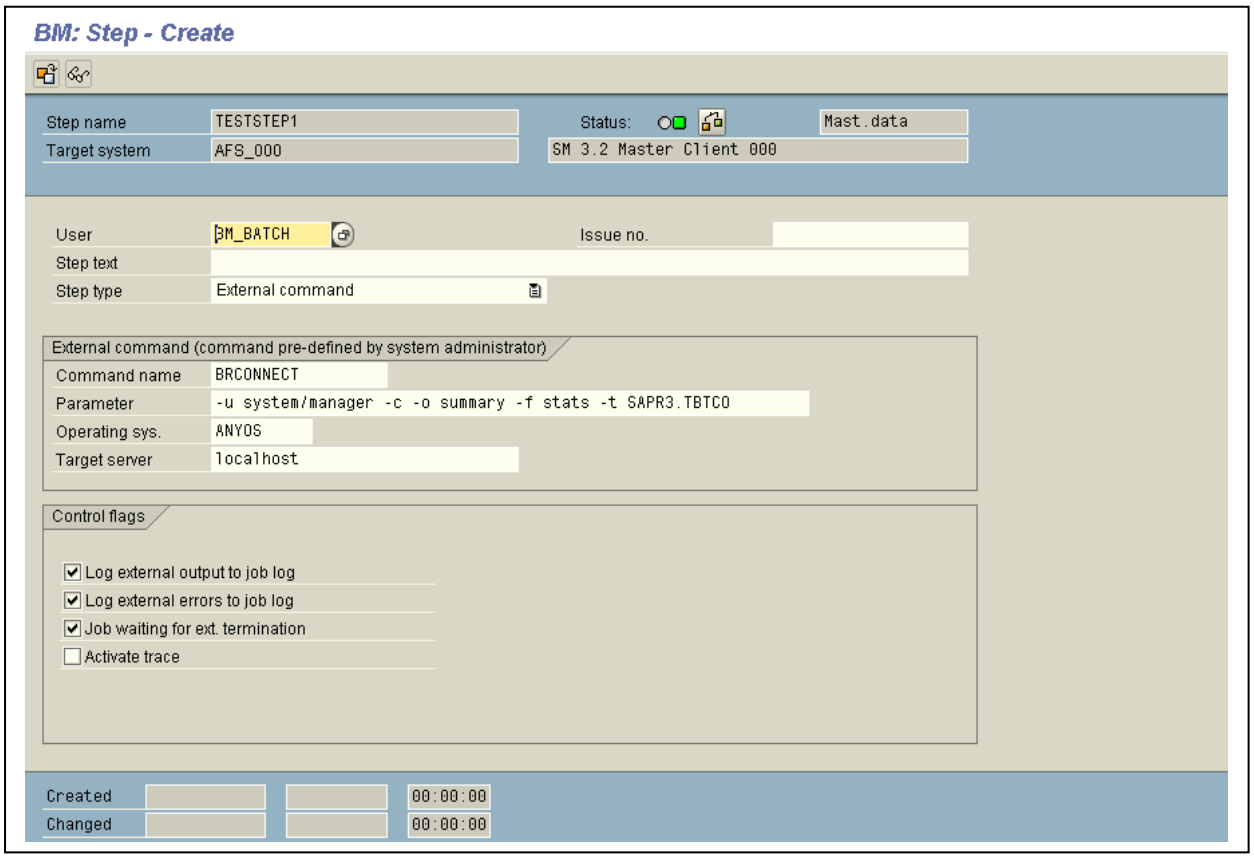
The **target computer** of the external programs is the host, on which a batch step should run. Under WindowsNT you can enter the local computer here. Under UNIX, remsh (remote shell) is used by SAP to start programs externally. Please pay attention to the appropriate OSS- Notes with the topic SAPXPG.

In general only ‚local host' should be entered in this field. You can define in the job on which host the program should be executed.

Alternatively the BatchMan Report **/BTCMAN/C_CMD** can be used. This has the advantage that TVARV variables are used there, which allow the creation of dynamic variants that are system-independent.

2.2.4. Step type: External command

This Step type is used for the indirect execution of operating system calls. The used commands must be predefined.



With an external **Command**, a normal user – user without administrator authorizations for the background processing system - can execute a host-system-command or program that was predefined by an SAP system administrator. In order to schedule the external command, the user must have the required authorization for the external command.

Options and parameters of the external command can be entered in the field **Parameter.** If one specification contains a blank, then it must be set in ‘ ‘apostrophes.

Example: - l 4 : = two values

'- l 4' : = one value

Note: The character that must be used to set a value in quotation marks is operating system-dependent.

The **Operating system** must be selected after entering the command, since the definitions of the external commands are operating system-specific.

The **Target computer** of the external command is the host, on which a Batch Step should run. Under WindowsNT you can enter the local computer here. Under UNIX, remsh (remote shell) is used by SAP to start programs externally. Please pay attention to the appropriate OSS- Notes with the topic SAPXPG.

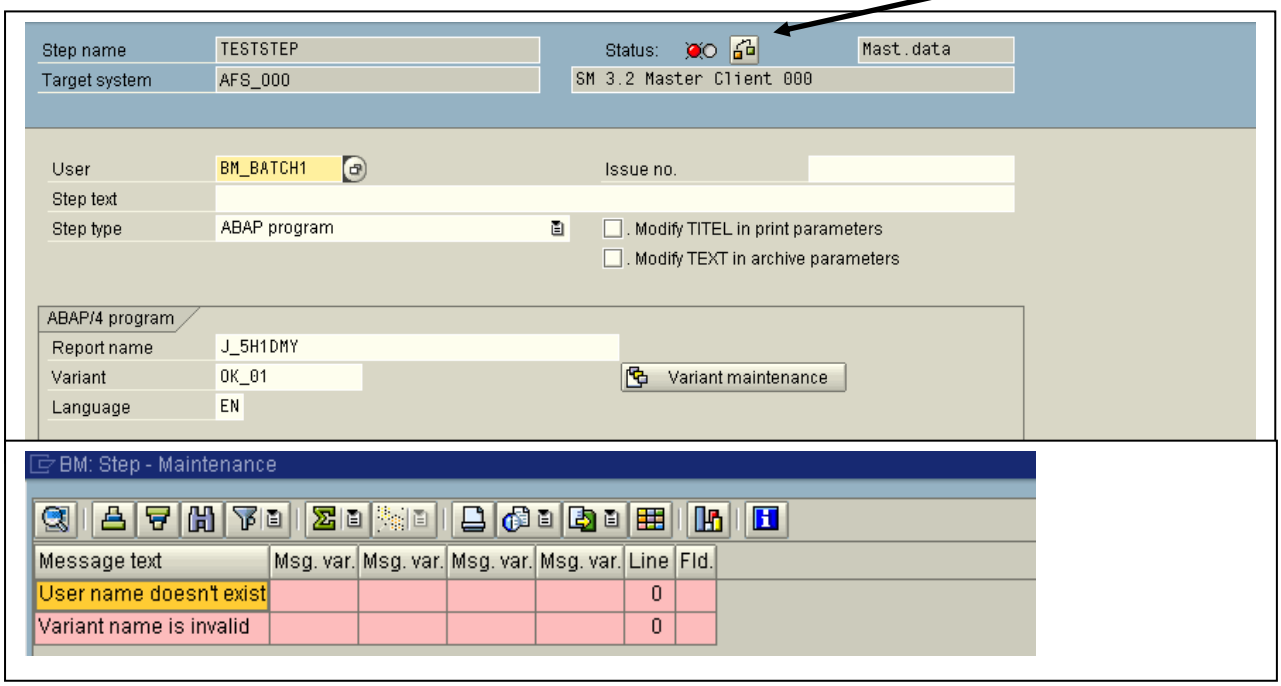
In general only ‚local host’ should be entered in this field. You can define in the job on which host the program should be executed.

2.2.5. Additional Functions for Step Maintenance

2.2.5.1. Checking a Step

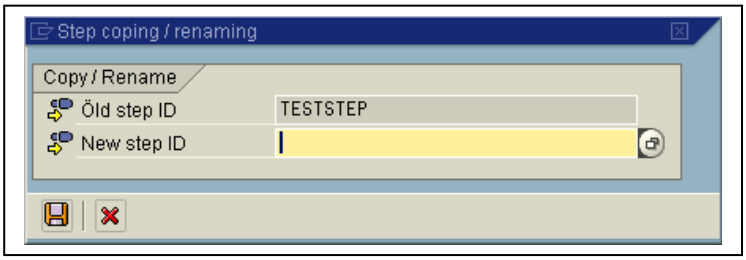
Every saving action checks the entered values against the SAP System (and therefore client). If all entries are without errors, the status becomes GREEN. If errors occur, the status becomes RED.

If the status is RED, the error list can be displayed by pushing the button ‚*Status'*.



2.2.5.2. Copy

Calling: function for copying the step. A window opens to request the new Step name.



2.2.5.3. Rename

Calling: function for renaming the step. A window opens to request the new step name.

2.2.5.4. Delete

Calling: function for deleting the step. This step can only be deleted if the step is no longer used in jobs. The step is deleted after a security query.

2.2.5.5. Other Object

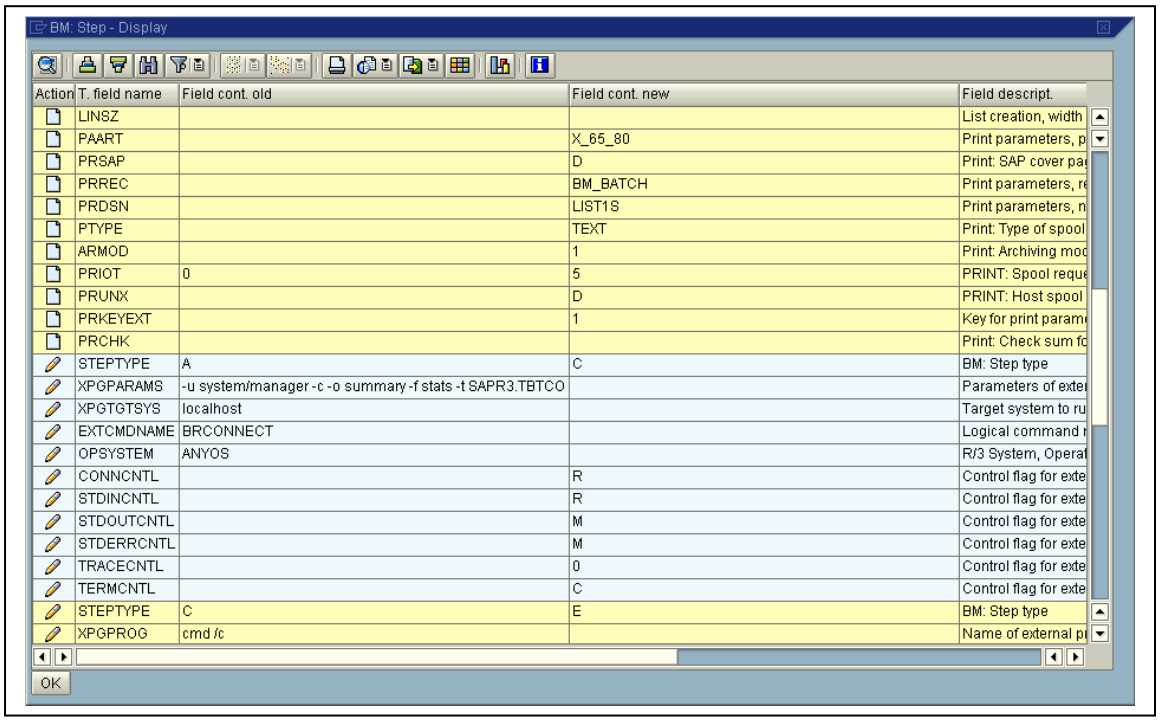
Calling: Init Mode of the step maintenance. All fields are deleted; the step name can be entered.

2.2.6. Menu Functions

The menu item **Step** contains the already described functions ‚Other *Object'*, ‚*Create*, ‚*Edit*, ‚*Display*, ‚*Delete'* and ‚*Save'*. The function ‚*Print* offers an option for printing all step values. See also ‚Reporting about Steps' below. Under **Edit** you can find the functions ‚*Copy'* and ‚*Rename'* (see above).

2.2.6.1. Go to → Change History

Calling the change history for a step. All changed fields are displayed. The icons of the action correspond to the ones of the buttons.



2.2.6.2. Goto → Deleted Steps

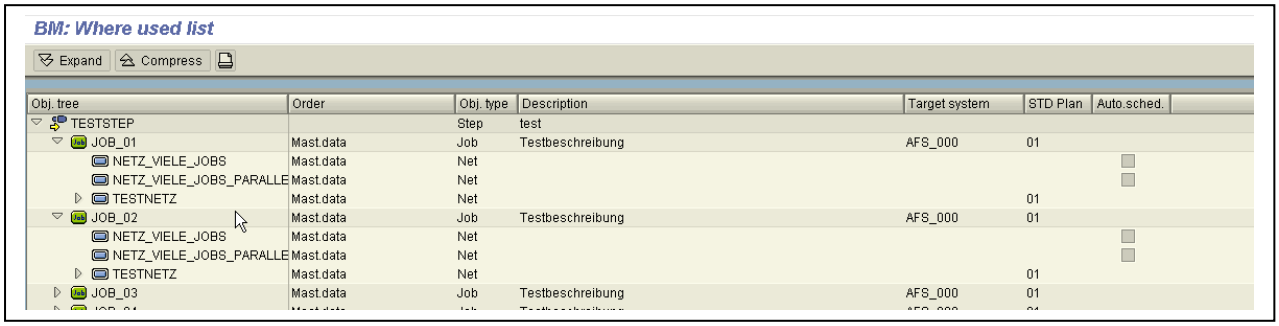
This function lets you display steps deleted in **BatchMan**. If a step is deleted, only one entry is created with the information that the step was deleted. If all other change documents for this step will be deleted depends on the setup parameter '<u>History will be kept if master data are deleted</u>'. If the value is active, then these can still be displayed via Reporting → Change history → Change History Steps also if the step is deleted.

2.2.6.3. Additions → Change RFC-Destination

The target system can be changed here. It is used in the step maintenance to call the search help and, when steps are checked and saved, to validate the values.

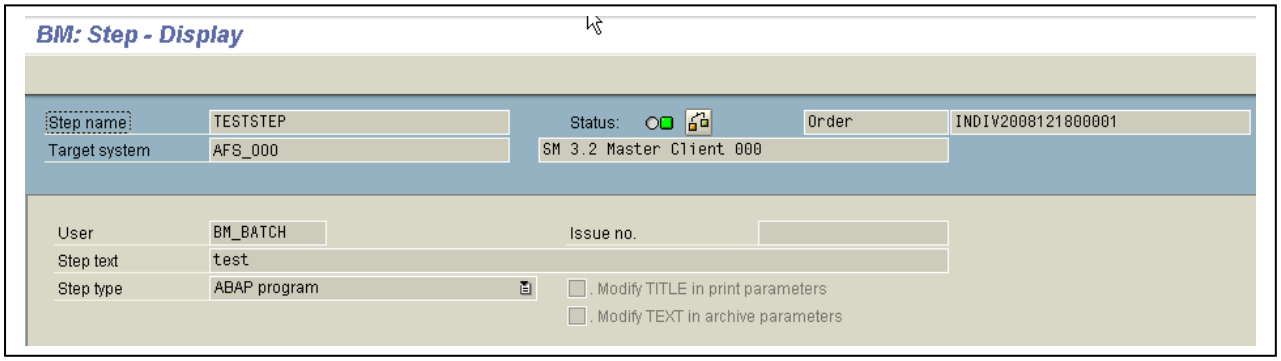
2.2.6.4. Environment → Usage only in Master Data

The function displays the usage of the current steps in jobs and all upper-level networks.

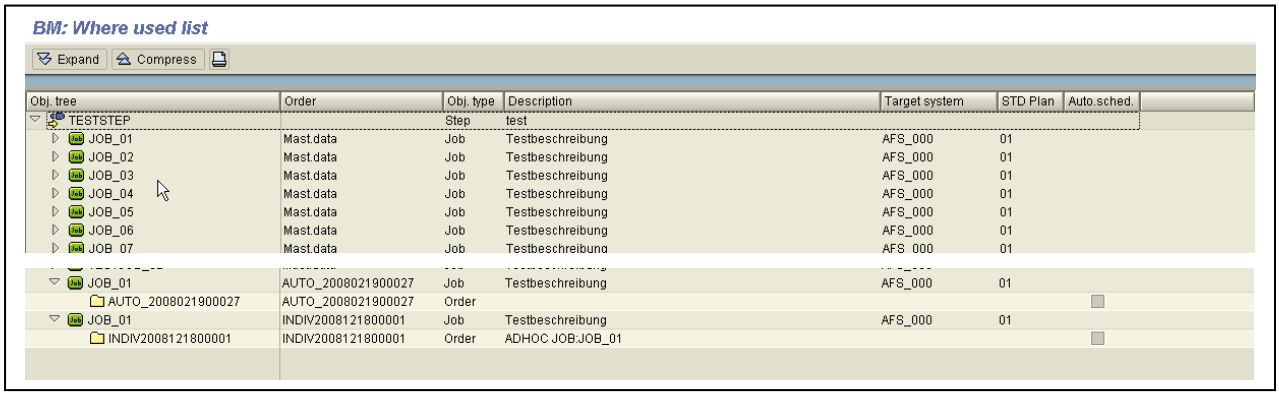


2.2.6.5. Environment → Usage of the Master Data and Order Data

The function displays the usage of the current steps in jobs and all upper-level networks, as well as the usage in the orders. If you branch from the order maintenance to the step maintenance, the system displays the step copied for the order. This is evident when “Order” and the related order number are now displayed in the fields next to the status.



The Where-Used-List displays the orders.

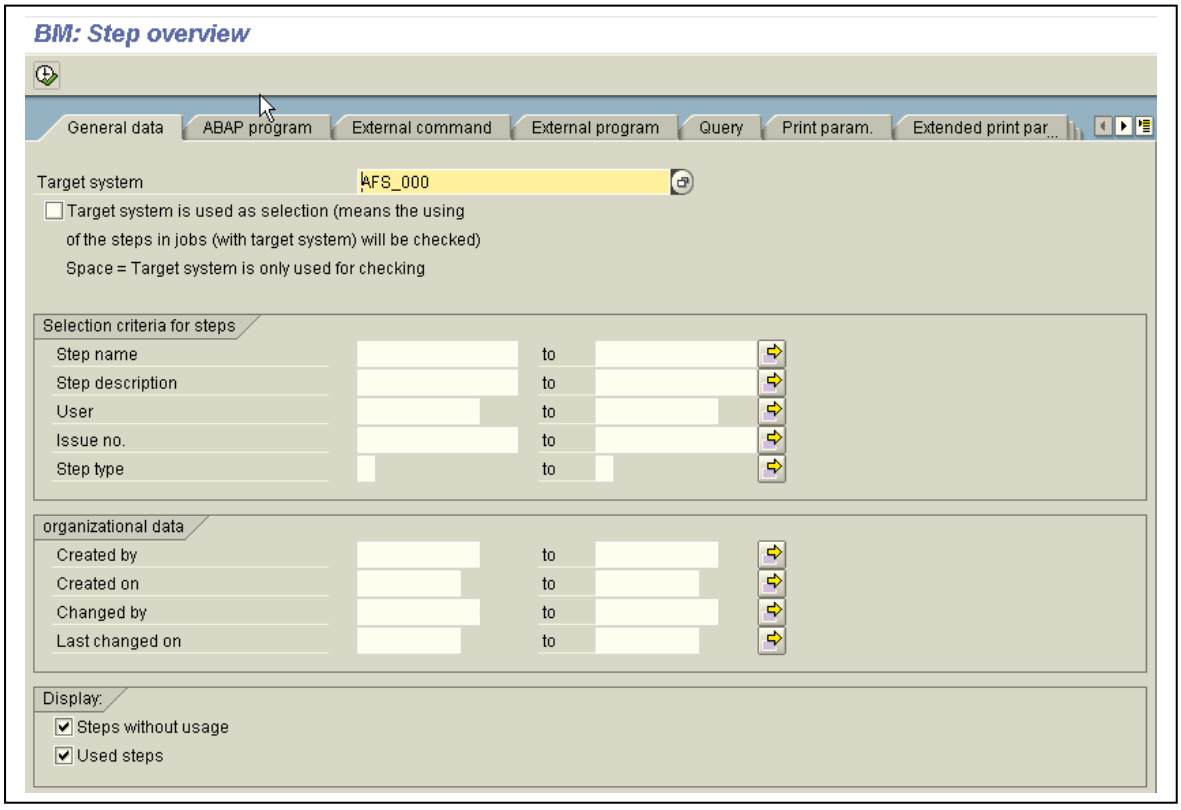


2.2.7. Reporting about Steps

The following evaluation options are available for job steps:

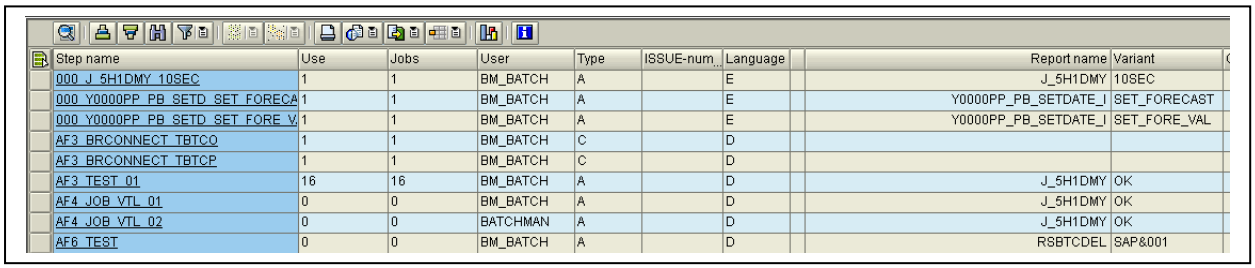
2.2.7.1. Step Overview

In this report, a step overview may be compiled by selective selection criteria. Here, it is possible to choose by step type. The target system entry is required. From the list, you can branch to step maintenance by double-clicking on the step selected.



With the "*Steps without usage*" flag active, only those steps are selected that are no longer used in any job.

The results list is output as an ALV list, and you can set it up as you like. You can change and save it using the layout management.



Step name	Use	Jobs	User	Type	ISSUE-num...	Language	Report name	Variant
000_J_5H1DMY_10SEC	1	1	BM_BATCH	A		E	J_5H1DMY	10SEC
000_Y0000PP_PB_SETD_SET_FORECA	1	1	BM_BATCH	A		E	Y0000PP_PB_SETDATE_I	SET_FORECAST
000_Y0000PP_PB_SETD_SET_FORE_V	1	1	BM_BATCH	A		E	Y0000PP_PB_SETDATE_I	SET_FORE_VAL
AF3_BRCONNECT_TBTCO	1	1	BM_BATCH	C		D		
AF3_BRCONNECT_TBTCP	1	1	BM_BATCH	C		D		
AF3_TEST_01	16	16	BM_BATCH	A		D	J_5H1DMY	OK
AF4_JOB_VTL_01	0	0	BM_BATCH	A		D	J_5H1DMY	OK
AF4_JOB_VTL_02	0	0	BATCHMAN	A		D	J_5H1DMY	OK
AF6_TEST	0	0	BM_BATCH	A		D	RSBTCDEL	SAP&001

Calling step maintenance; analog to a double click on a line

Detailed display of all step fields in one list (see ‚Step details'); the steps must be marked

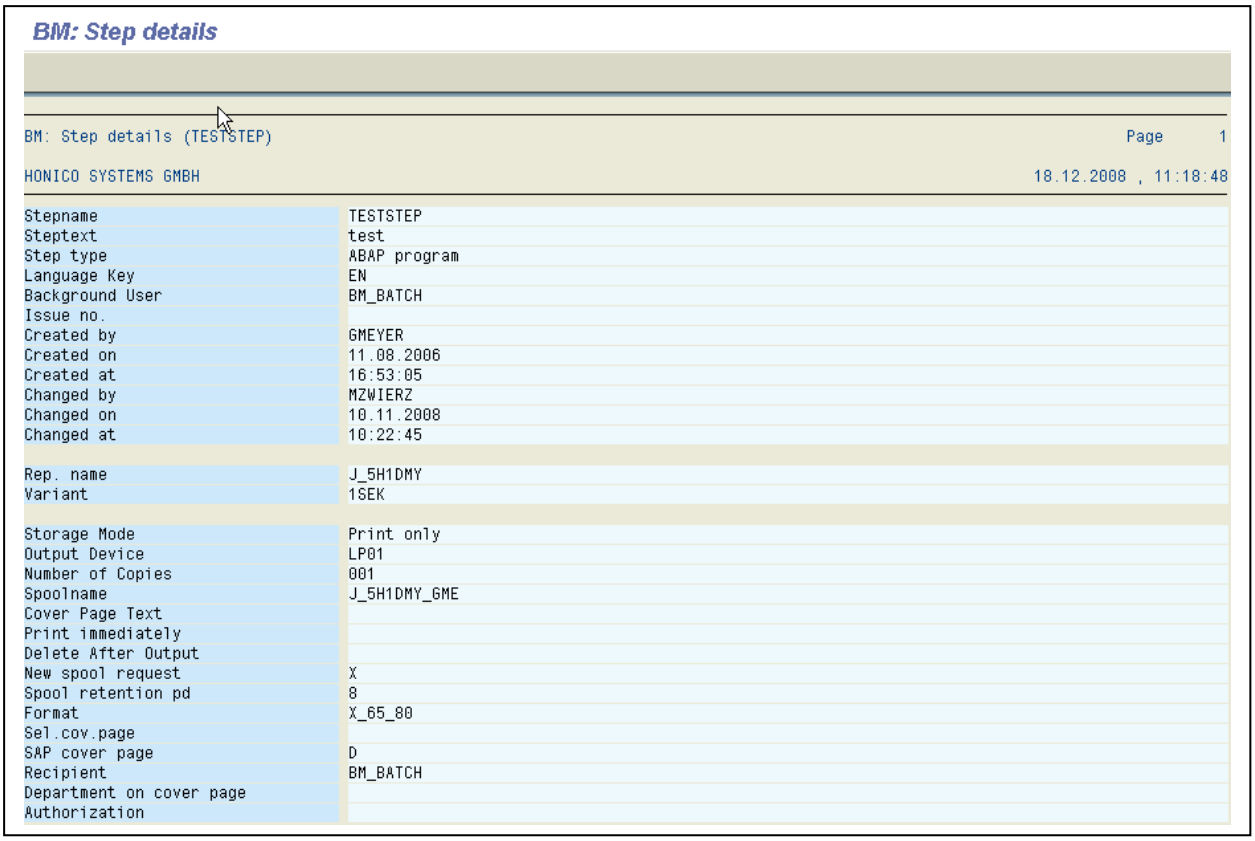
Where-used list of a step; the steps must be marked

Delete a step (not only from the list). The step usage is checked.

2.2.7.2. Steps: Detailed List

This function can be used for a complete printout of one step or of all steps. All fields of a step are specified. Only the fields are displayed that, for example, are relevant for the step type.

Excerpt:



BM: Step details

BM: Step details (TESTSTEP) Page 1

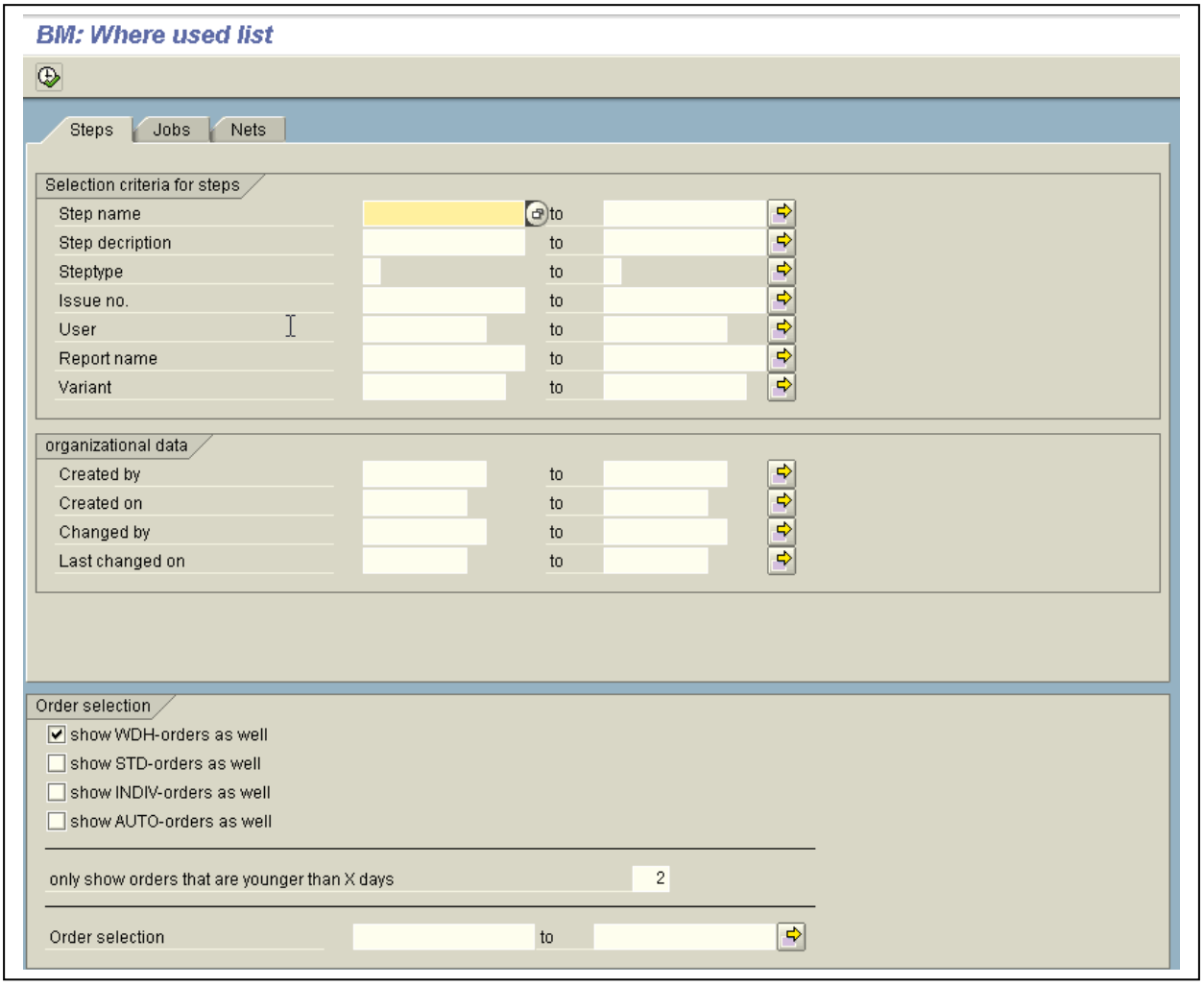
HONICO SYSTEMS GMBH 18.12.2008 , 11:18:48

Stepname	TESTSTEP
Steptext	test
Step type	ABAP program
Language Key	EN
Background User	BM_BATCH
Issue no.	
Created by	GMEYER
Created on	11.08.2006
Created at	16:53:05
Changed by	MZWIERZ
Changed on	10.11.2008
Changed at	10:22:45
Rep. name	J_5H1DMY
Variant	1SEK
Storage Mode	Print only
Output Device	LP01
Number of Copies	001
Spoolname	J_5H1DMY_GME
Cover Page Text	
Print immediately	
Delete After Output	
New spool request	X
Spool retention pd	8
Format	X_65_80
Sel.cov.page	
SAP cover page	D
Recipient	BM_BATCH
Department on cover page	
Authorization	

2.2.7.3. Where-Used-Lists for Steps

Calls where-used list for individual steps or all steps. This is possible both for master data and orders. For this, set the related flag for the order selection.

Using the order selection field (orders), you can restrict certain orders, e.g. via the number of the days you can display orders that are not older than today's date minus the number of the days.



Results list such as in 2.2.6.4. or 2.2.6.5. In all objects, you can branch directly to the display by double-clicking.

2.3. Administration of Jobs

You can call job maintenance with to options: directly with transaction J5HBN or via the function tree: *Master data/ Library → Job maintenance.*

2.3.1. Introduction to Jobs – General data

Even though the term job is further used in **BatchMan**, it should be more understood as process. Within the job definition the type of the job is defined via the process type.

Process type	**Sub-process type**	**Number of steps**
normal Job		1 – n
BW Process	Process chain	1
	Load InfoPackage	1
FI Process	Dunning run	1
	Payment run	1
	Copy dunning run	1
	Copy payment run	1
Other process	Archiving (SARA)	1
FDM JOB		1 – n

A job consists of a sequence of processing steps (steps) and is part of the library.

The steps of a normal job are managed in a step list that can be edited individually. In this step list, for example, the sequence of the steps can be determined and changed at will. All other process types do not have a step list, but are defined as individual steps.

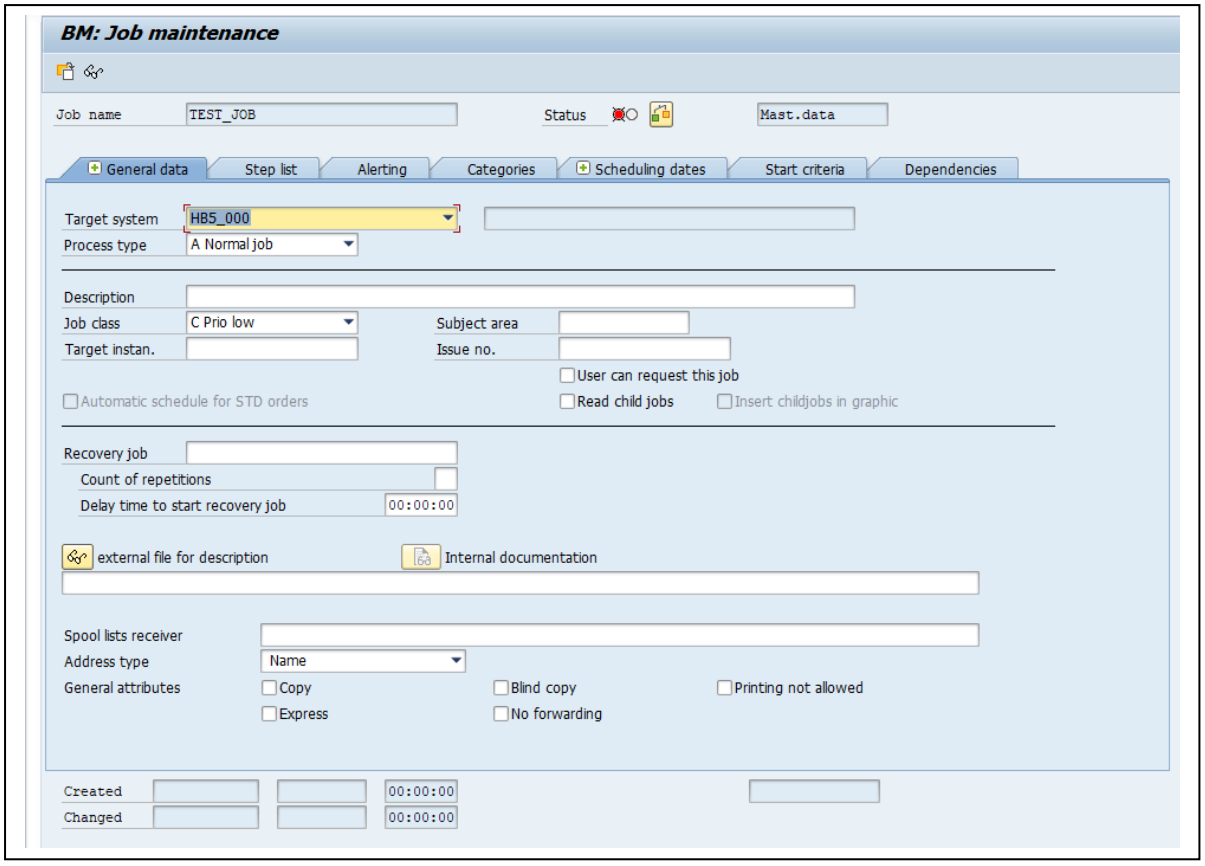
After entering the **Job name** you can select the desired function:


Create a new job


Edit an existing job


Display an existing job (also: ENTER key)

Create a job



The target system in the job maintenance is used to call the search help and to check/save the job with validation of the values. It is a mandatory field.

The main usage of the **target system** in jobs, however, is the specification on which system the job execution should take place. It is saved with/to the job.

The target system is saved user-specific. If you first enter the job maintenance, this field is not maintained. That is the reason why the default system form the definition of the remote connections is used. If this is not specified, then a popup appears in which a target system must be entered by the user. When you use the job maintenance afterwards, the last used target system is read from the user parameters.

Via the list box **process type** you can then determine the special process type. The table above gives a rough overview on the existing process types.

The **description** is a long text to describe the job.

The **Job class** (Default 'C') enables you to classify jobs according to priority. Indicate one of the following classes for your job:

A	Jobs with high priority, whose execution is mandatory for continuing systems operations. Jobs of class A are always processed before jobs of lower classes.
B	Jobs which should be executed regularly, for example jobs for the transmission of performance statistics. Jobs of class B are processed before jobs of class C.
C	Jobs that neither fit in class A nor class B and which don't require higher priority. Class C is the standard class.

With the help of **subject areas** you can group jobs and networks or classify them. In reporting, an evaluation/analysis can be created on the bases of subject areas.

A target instance or target server group existing on the target system can be entered in the **Target instance** field.

If the SAP system consists of several instances (entities) and no target system is specified, then the SAP scheduler determines in which host the job will be processed. All batch-capable target instances are displayed with a match code function.

The **ISSUE Number** is a blank field. You can deposit sort criteria of other software here, e.g. of HelpDesk Systems.

Via the flag ‚**Automatic schedule for STD orders'**, this job will be activated for the automatic scheduling of the STD orders. When an operator order of the type **STD** is created only those jobs will be suggested automatically for selection in which the checkbox automatic scheduling is marked (see also Order Administration).

"**Users can request this job"** is a special enhancement to the interception interface. With this it is possible to predefine in BatchMan jobs, which are supplied with only a start option by the end users of the specialist departments. In order to be able to use this function, you have to release the function in SAP using report **INITXBP2**.

If the flag **, Read child jobs'** is activated, then those jobs are read at the end of the defined job that were generated by this job (SAP calls these Parent Child Jobs). In order to use this function, the functionality has to be released in SAP with the report **INITXBP2**.

Using **"Insert child jobs in graphic,"** you can also insert the child jobs found into the graphic display.

The best known example might be the report RV60SBAT. It generates batch jobs for die billing. These jobs all begin with INVOICE*. Subsequent processing is only an issue when all INVOICE* jobs are finished. Then this flag will be activated.

Using the **External file for job description** link, you can define an external file for the documentation of this object. (also see menu function: Goto Object documentation). You can call the **Internal documentation** via the change button (or the display button). This is only possible after the first save.
The **job documentation solution manager** branches into the job documentation of the SAP Solution Manager. Depending on the setting, the documentation is created/changed or just displayed.

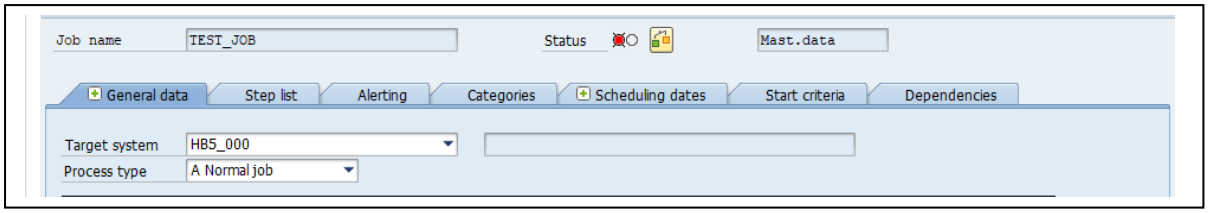
Via the **Spool lists receiver**, the receivers of the spool lists generated by the job (print list) are specified. This may be an SAP Office user, an Internet e-mail address or also a general SAP Office distribution list (**recipient type**).

Recovery job: A recovery job is a job that is automatically triggered when the original job is terminated. Via **Count of repetitions,** you can restrict the number of repetitions of a recovery job. This only works if the recovery job is the same job as the job itself. **Delay time to start recovery jobs** determines the delay time between when the termination is detected and when the recovery jobs is started.

If you use this function, bear in mind that only **one** step is defined in the step list for a normal job. The reason for this is that is it not possible to determine in an SAP table which step has terminated in a job. This can only be scanned and detected using the job log.

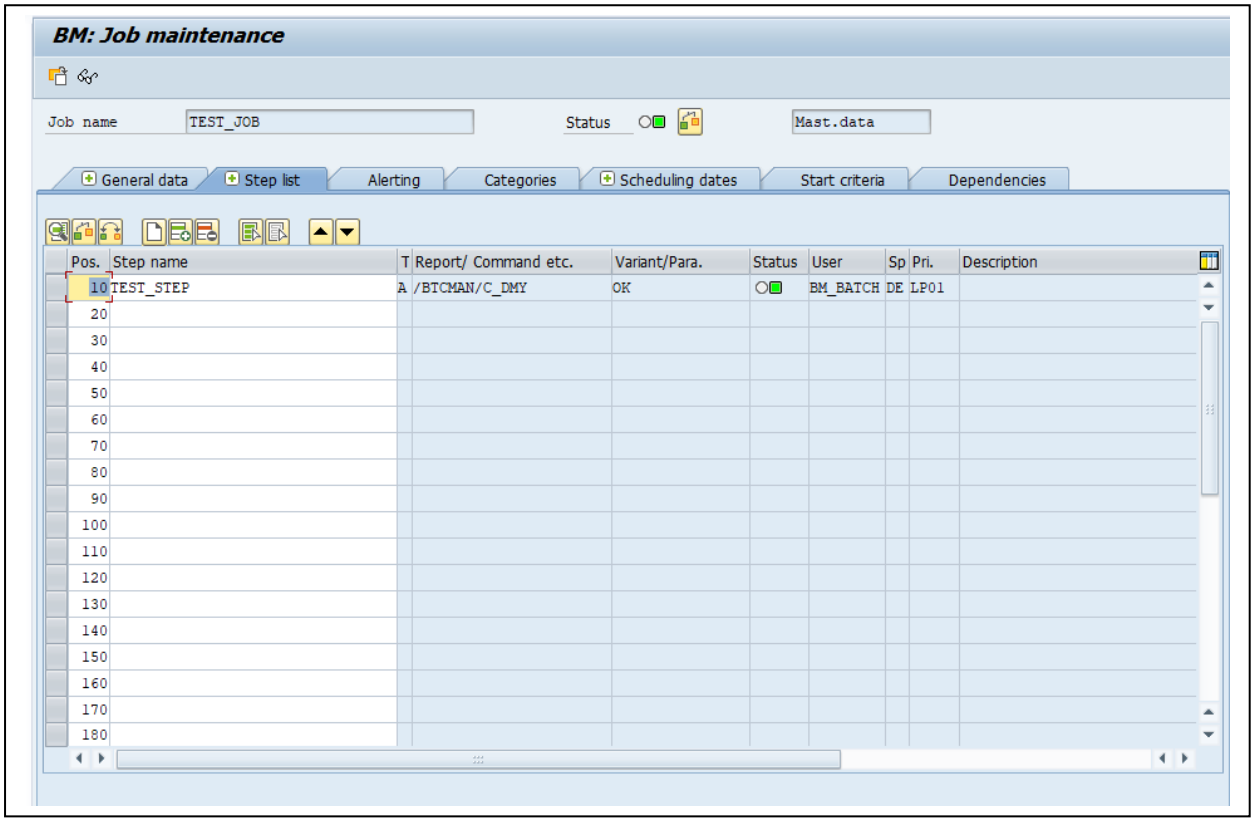
The scheduling of the recovery job takes place parallel to the original job. Also, the status of the original job is switched from "aborted" to "aborted/ confirmed".

2.3.2. Job Type: Normal Job



A normal job corresponds to the job known in SAP. It consists of one to n steps. In **BatchMan,** these steps are those objects already described above.

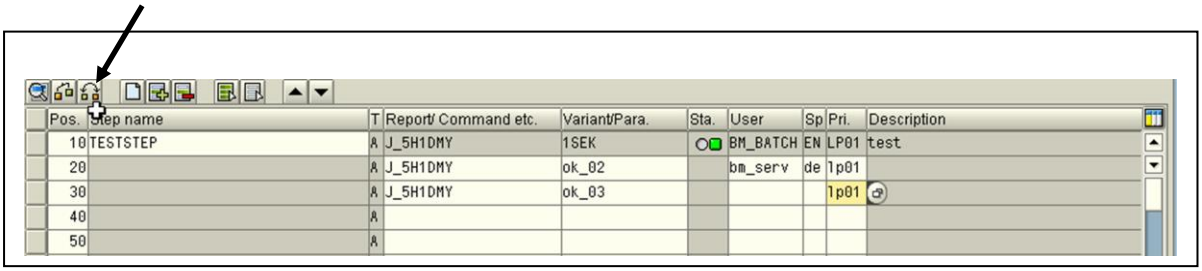
Like the name implies, the step list is maintained using the “Step list” tab page. When you create a job, the step list is empty.



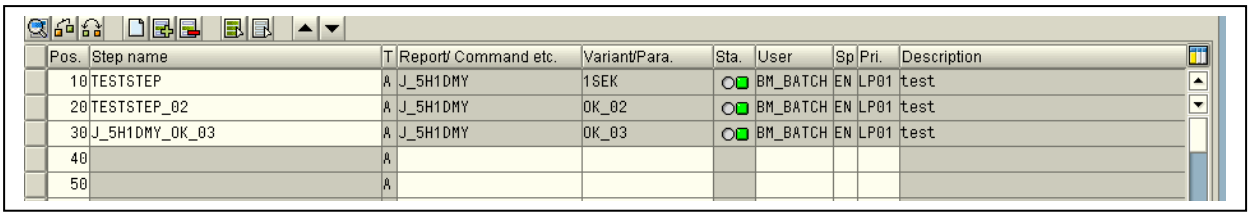
Using input help (F4) in the step name, you may choose from steps already created. If the name is known, it is entered in the step names. When you choose ENTER, the system reads the step values.

Another option is to directly enter the report and the variant – just switch between step name or report/variant with the button as indicated below.

This function is only available for the ABAP step type.



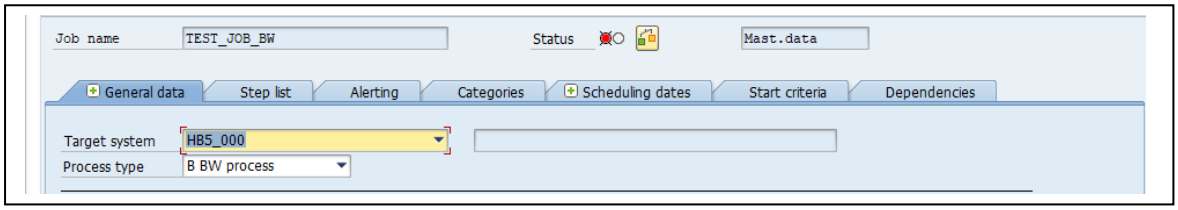
If a step is found with the report and the variant (Pos 20), this is used. If no step exists (Pos 30), one is created upon saving the job. If the fields for the user, language and printer are not maintained, the system uses the values from the setup.



Pos.	Step name	T	Report/ Command etc.	Variant/Para.	Sta.	User	Sp	Pri.	Description
10	TESTSTEP	A	J_5H1DMY	1SEK		BM_BATCH	EN	LP01	test
20	TESTSTEP_02	A	J_5H1DMY	OK_02		BM_BATCH	EN	LP01	test
30	J_5H1DMY_OK_03	A	J_5H1DMY	OK_03		BM_BATCH	EN	LP01	test
40		A							
50		A							

2.3.3. Job Type: BW Process

The “BW process” job type schedules special processes in an SAP BW system. The scheduling occurs via BAPI or by interfaces released by SAP. The job class in the definition has no significance.

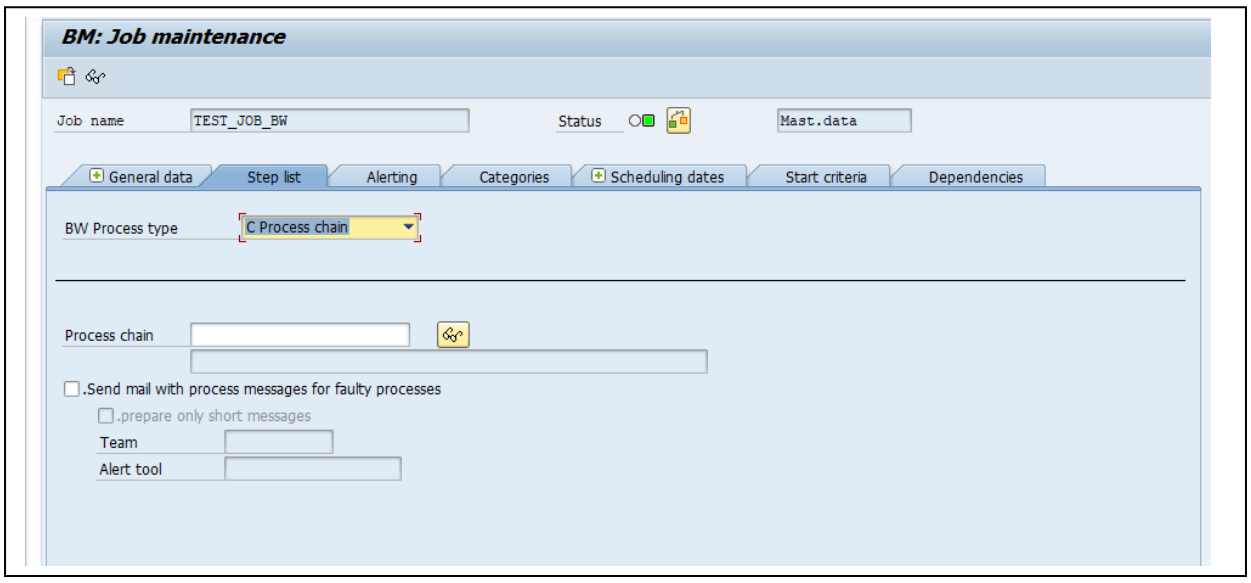


Within the process type ‚BW Process', so-called sub-types must be maintained instead of the step list (or if only one step is present here).

Since within the BW system a number of internal entries are saved for the scheduling of processes, this scheduling is the only option (especially the use of process chains) to carry out scheduling properly. Scheduling BW reports directly is not recommended.

2.3.3.1. BW Process Sub-Type: Process Chain

A process chain within an SAP BW System is a locked (in itself) processing chain. Within the chain, a multitude of different BW internal processes with the respective dependencies is defined.



The name of the **process chain** can be selected in the field with function key F4. The displayed list of the available process chains is called on the target system. This way the values are always current. The text of the process chain is displayed below.

As already mentioned, there are a number of BW internal processes mapped in a process chain. These processes are executed by BW-internal mechanisms as SAP jobs. The messages arising in the process are written to the application log of the SAP BW system.

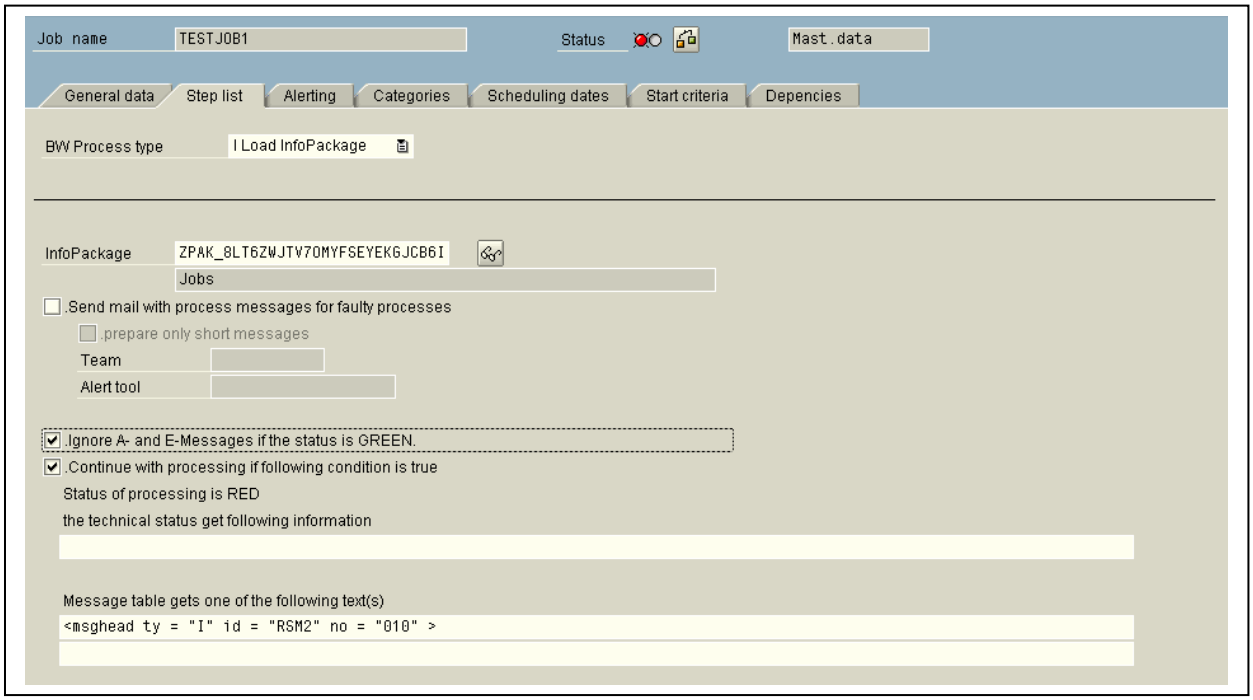
When you activate the "**Send mail with process messages for faulty processes"** flag, it is possible to switch on a separate alert independent of the other alerts at the job. This makes it possible for instance to inform the BW team in more detail than the job scheduling team, which only receives a message about the failure.

If you enter a **BatchMan team** and **alert tool** accordingly, then the system reads out and sends messages from the application log for the faulty processes. This is not the case if the "**Prepare only short messages"** flag is activated. Then the system only outputs the process.

2.3.3.2. BW Process Sub-Type: Load Info Package

Obsolete, please no longer use.

With the help of an InfoPackage you determine the selection conditions for the data request of a source system in the BW.



The name of the **InfoPackage** can be selected with the F4 function in the field. The displayed list of the available InfoPackages is called on the target system. This way the values are always current. The text of the InfoPackage is displayed below.

When you activate the "**Send mail with process messages for faulty processes"** flag, it is possible to switch on a separate alert independent of the other alerts at the job. This makes it possible for instance to inform the BW team in more detail than the job scheduling team, which only receives a message about the failure.

If you enter a **BatchMan team** and **alert tool** accordingly, then the system reads out and sends messages from the application log for the faulty processes. This is not the case if the “**Prepare only short messages”** flag is activated. Then the system only outputs the process.

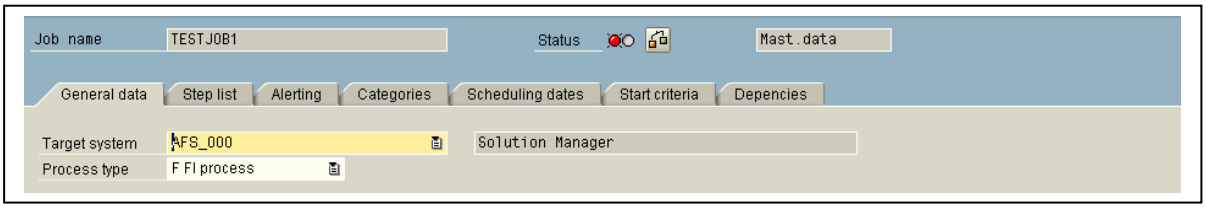
After an InfoPackage has finished being processed, the messages of the InfoPackage are evaluated. Here, A (termination) or E (error) messages may arise, even though the status of the process displays a green traffic light. If these messages are to be ignored, activate the “**Ignore A and E messages when status is GREEN”** flag.

The technical status text then delivers a language-dependent text (German: “Verarbeitung ist überfällig”; English: “Processing is overdue.” The message table available as of BW 3.x delivers a language-independent entry as an entry. By activating the **“Continue with processing if following conditions are true”** flag, you can continue the processing.

In the **technical status text,** you may enter the language-dependent text and or fill the **lines of the message table**. The output message for a process still active reads in the message table as follows: "**<msghead ty = "I" id = "RSM2" no = "010" >**". (The language-dependent message for this reads: "Request is still running".) The bold text is entered in one of the two lines.

2.3.4. Job Type: FI Process

This job type helps schedule special processes in financial management. These processes may only be scheduled on one target system that also contains the FI component (i.e. ERP or ECC; but not a pure Netweaver system). You can find a detailed description of this job type with examples in BM4_FiProcesses_YYMM.

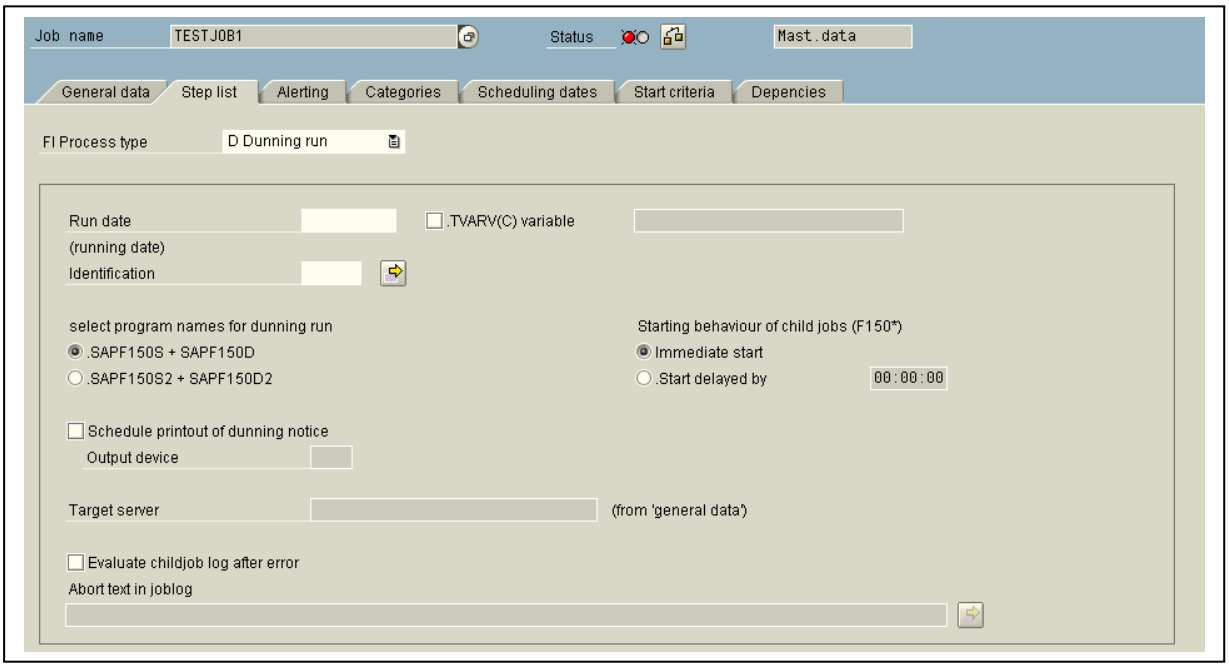


Choose the “FI process” process type.

2.3.4.1. FI Process Sub-Type: Dunning Run

Scheduling and monitoring dunning runs. These are normally scheduled by the specialist department via transaction F150.

Here, a fixed date is used; it is recommended to use a TVARV(C) variable.



This process schedules the dunning run analogous to transaction F150. Within BatchMan, a function module is used to implement the scheduling.

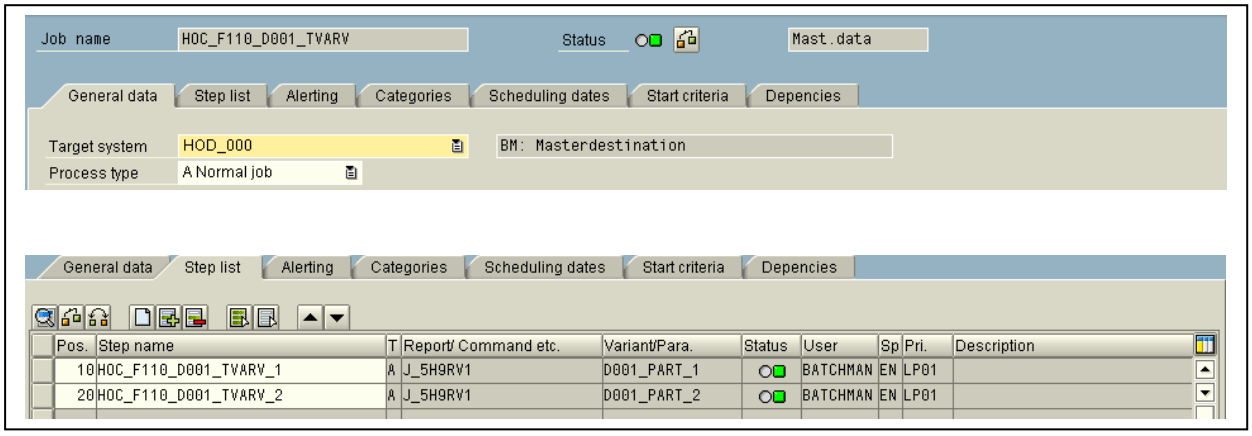
If the job F150-<SY_DATE>-<IDENT>* (sub-job = child job) is scheduled, the process goes in BatchMan to “wait.” The master job monitors the child job until the job is ended.

The dunning notice printout is scheduled as well. Otherwise, the dunning notice printout is to be scheduled as a “normal job.”

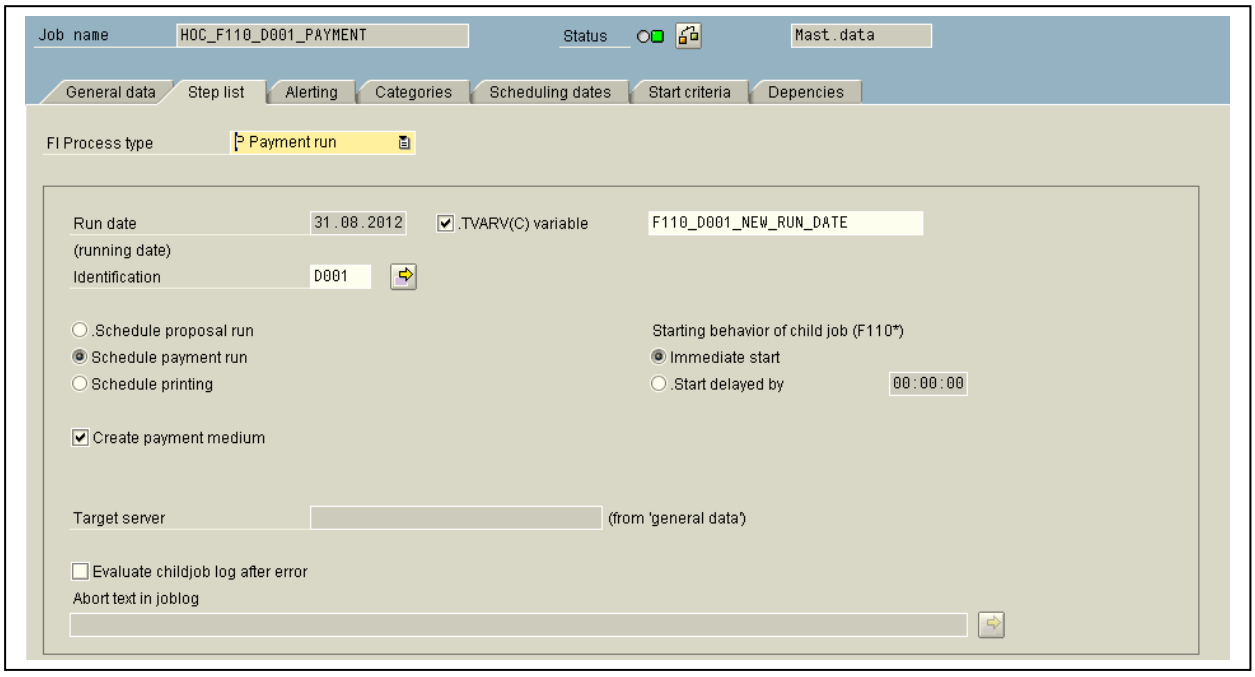
2.3.4.2. FI Process Sub-Type: Payment run

Scheduling and monitoring payment runs. These are normally scheduled by the specialist department via transaction F110.

This short example uses TVARV(C) variables. For this reason a “normal job,” which updates the variable, is required as the first process in the process chain.

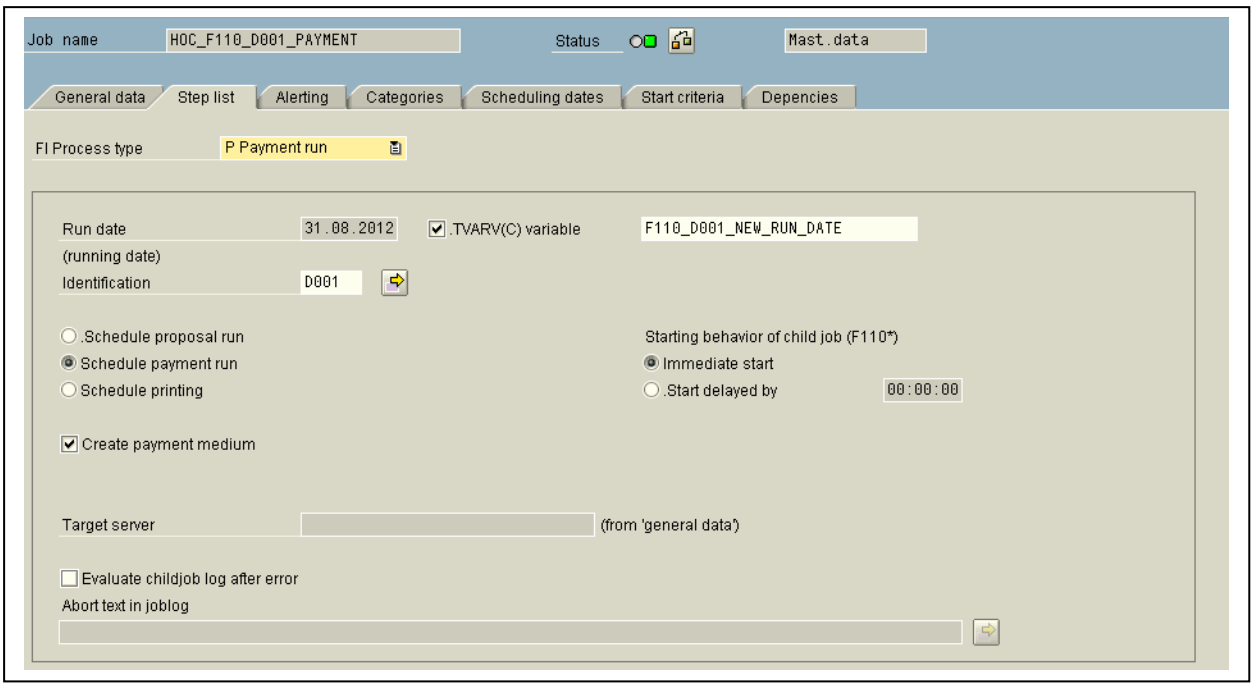


After the variables have been updated, the contents of these variables are used in scheduling the processes. By setting the “**TVARV(C) variable”** flag, the input fields are toggled. After you input the variables, the system reads the value from table TVARV(C).



This process schedules the proposal run analogous to transaction F110. Within BatchMan, a function module is used to implement the scheduling.

If the job F110-<SY_DATE>-<IDENT>* (sub-job = child job) is scheduled, the process goes in BatchMan to “wait.” The master job monitors the child job until the job is ended.

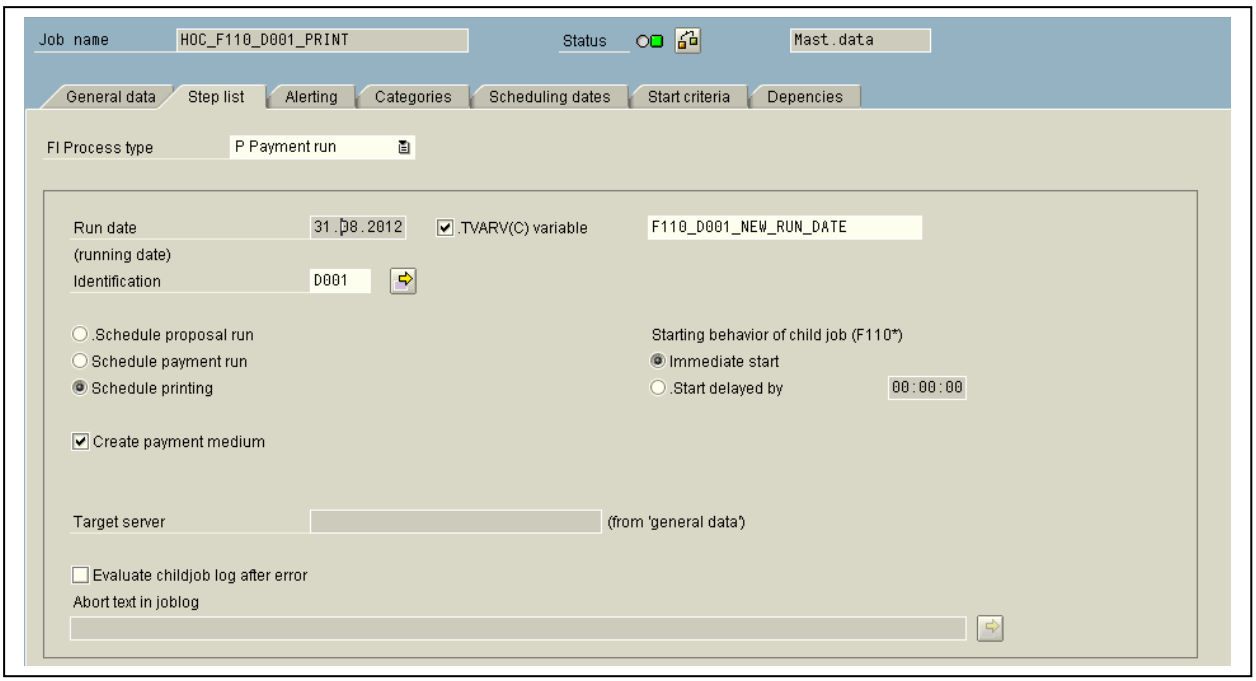


This process schedules the payment run analogous to transaction F110. Within BatchMan, a function module is used to implement the scheduling.

If the job F110-<SY_DATE>-<IDENT>* (sub-job = child job) is scheduled, the process goes in BatchMan to “wait.” The master job monitors the child job until the job is ended.

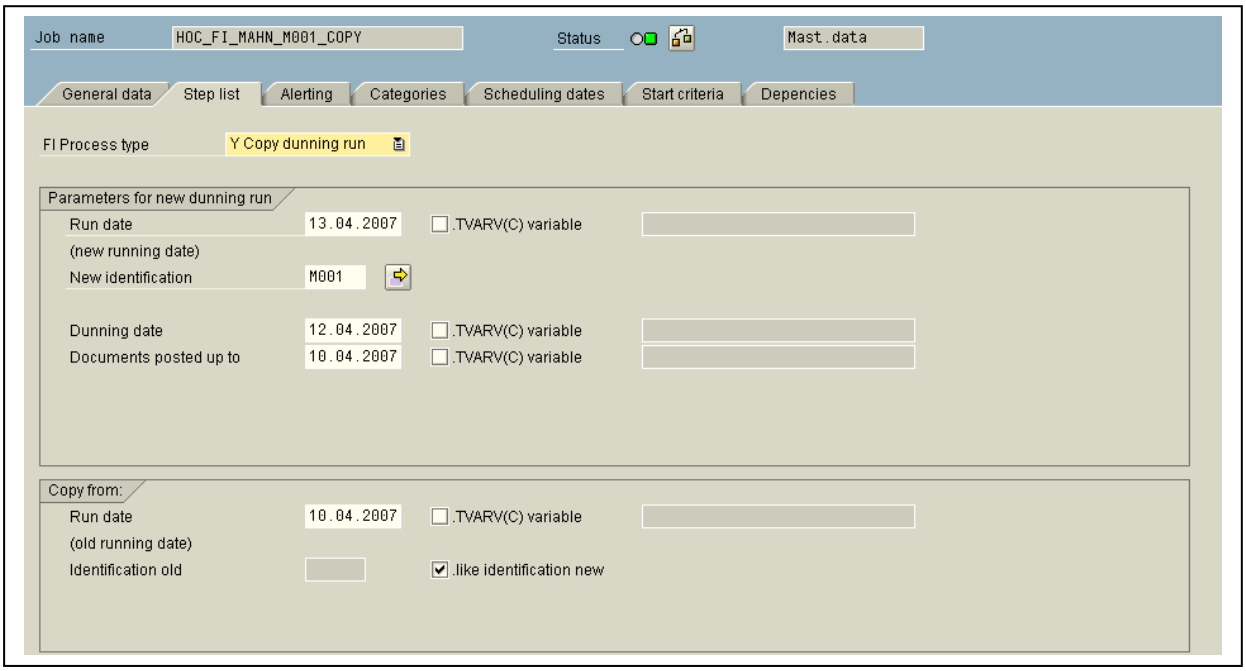
This process schedules analogous to transaction F110 the printout and creation of the payment media. Within BatchMan, a function module is used to implement the scheduling.

If the job F110-<SY_DATE>-<IDENT>* (sub-job = child job) is scheduled, the process goes in BatchMan to “wait.” The master job monitors the child job until the job is ended.



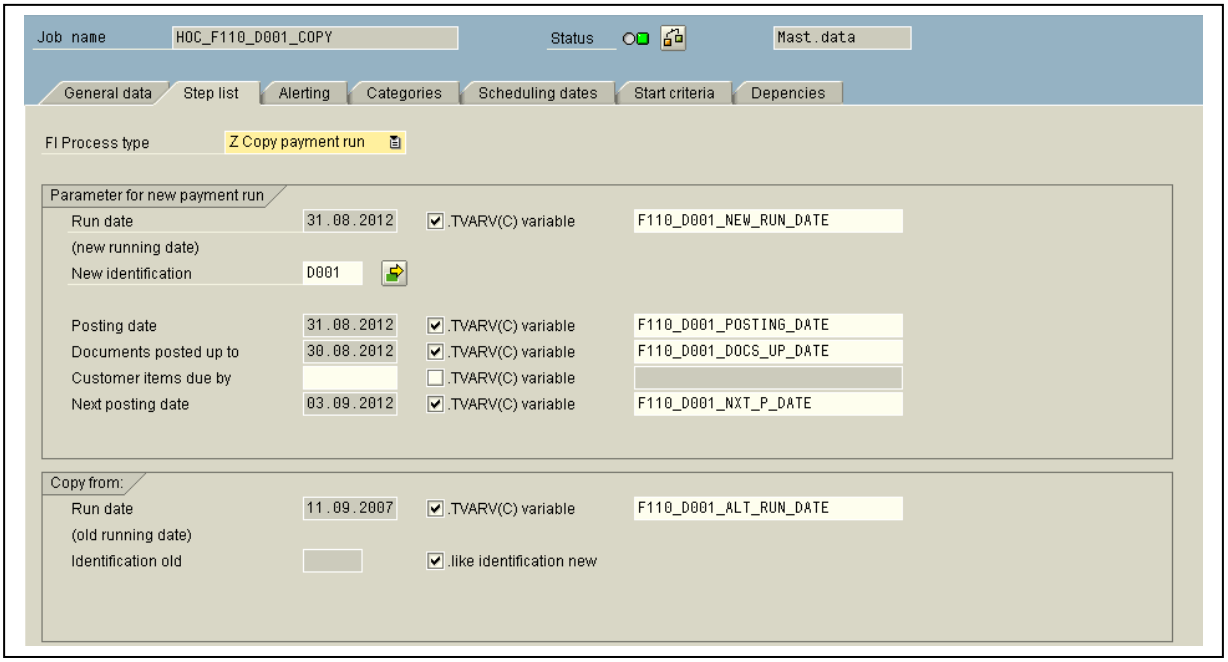
2.3.4.3. FI Process Sub-Type: Copy Dunning Run

Copy an “old” dunning run to a “new” dunning run while keeping the run ID. Here, fixed date fields are used; it is recommended to use a TVARV(C) variable.d



2.3.4.4. FI Process Sub-Type: Copy Payment Run

Copy an “old” payment run to a “new” payment run while keeping the run ID. Here, TVARV(C) variables are used again, so that the flow of the payment runs shown is completely dynamic.



2.3.5. Job Type: Other Process

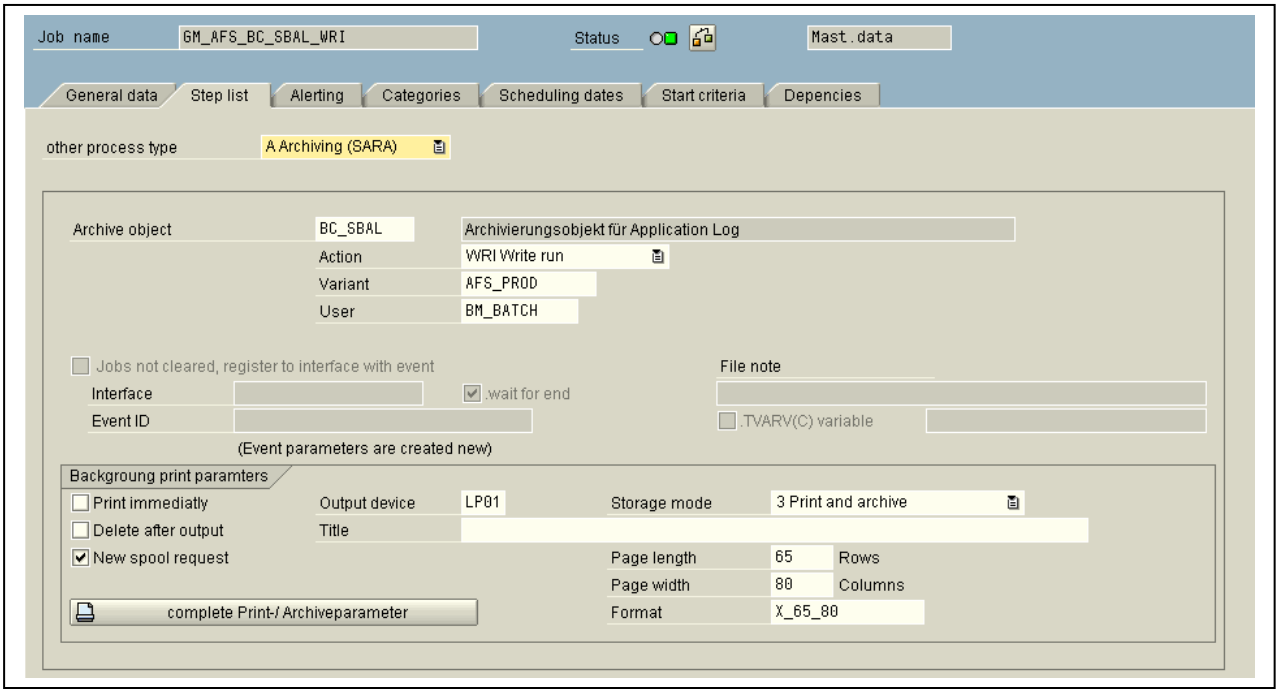
2.3.5.1. Other process sub-type archiving (SARA)

This process type can only be used as of SAP BASIS 6.20 and higher.

Schedule archiving runs that normally are controlled via transaction SARA. It is possible to schedule the preliminary run, writing, subsequent run, deletion run and storage run. These processes all have in common the fact that they generate additional jobs (from one to n). Having BatchMan monitor all jobs (also sub-jobs = child jobs) ensures that a scheduled process (archiving) is ended when all sub-jobs = child jobs are also ended.

Any additional required runs (e.g.: index build) may be scheduled as normal jobs using the programs available for each archiving object.

The use thereof assumes a maintained Customizing of the archiving objects (transaction AOBJ). The maintenance and thus scheduling using BatchMan is oriented with many of the settings and fields in the archiving Customizing and transaction SARA.



Choose the **archiving object** that is to be archived. Selection via input help (F4) is possible. Confirm with the Enter key; the system then reads the Customizing settings for the object and fills the actions possible. Afterwards, the other fields are ready for input.

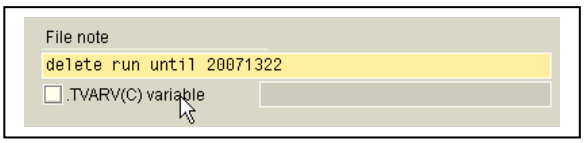
The **actions** possible depend on the Customizing of the archiving object. Maximally available are *preliminary run, write, subsequent run, deletion run* and *store*.

Depending on the action selected, the **variant** is to be specified. For instance, *write* and *delete* require a variant, while *store* does not.

The **user** is the name of the user who starts the archiving. This corresponds to the user who is entered in transaction SARA.

The **print parameters** are to be maintained for each run set. Since these are stored in the same table as the print parameters of the steps, the job names should be unique to the step names.

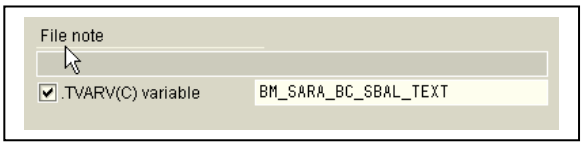
File note: If it is set in the archiving Customizing that the storage or the deletion run are to be executed separately from the writing run, then the note for all files of a storage run or deletion run may be updated.



The setting in Customizing of the archiving object is to be set or left set to “no scheduling” for the deletion jobs. Analogously, the “**Start automatically”** flag is not to be set for the storage action.

So that the content of this field does not have to be adjusted for each run, a TVARVC variable is to be used. This variable contains the corresponding text.

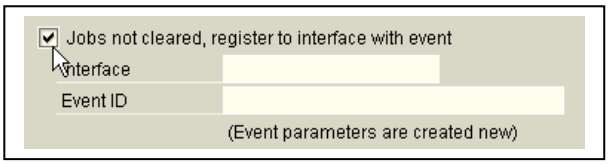
The variable is to be held for the system (and clients) on which the archiving runs.



The notes are updated for ALL files of a storage run or deletion run. That means if more than one writing run (two runs) were executed and then for instance the storage action is run, the files of both writing runs are stored and the notes are updated.

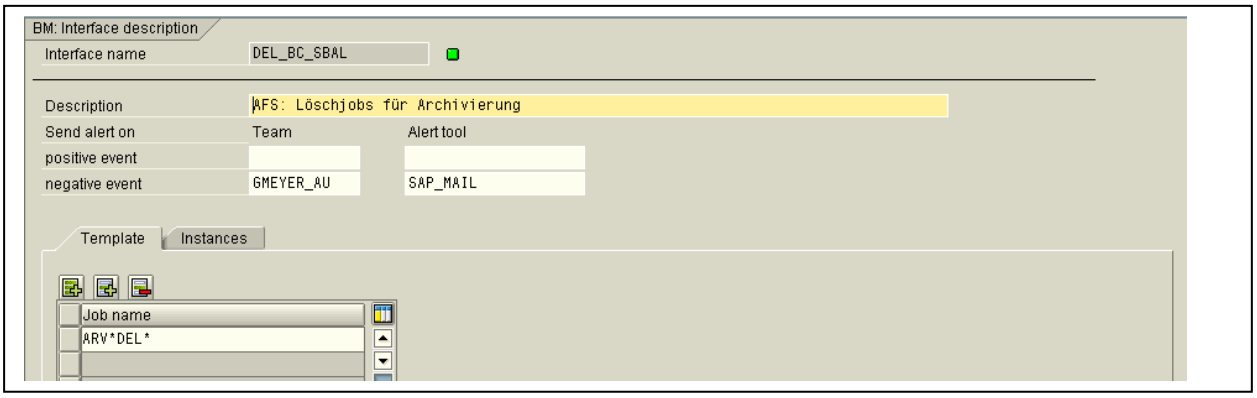
Load restriction when archiving is only possible for storage runs or deletion runs if they are not executed together with the writing run.

For storage runs or deletion runs, it is possible to activate the “**Jobs not cleared, register to interface with event”** flag



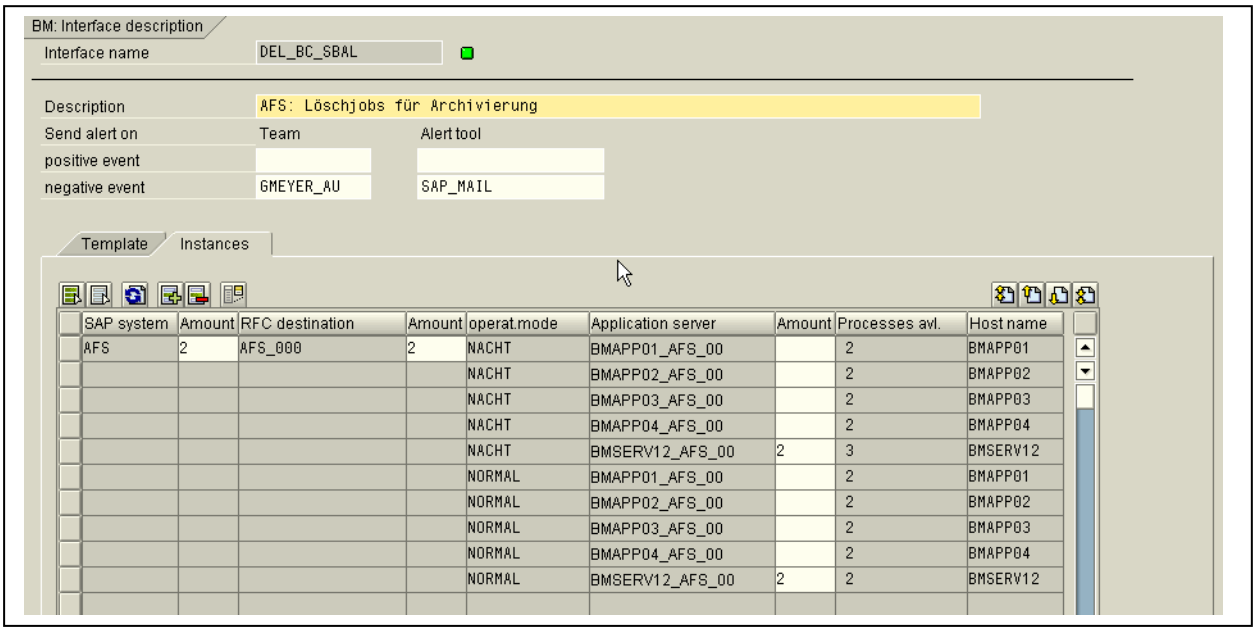
This way, all sub-jobs = child jobs of a storage run or deletion run may be registered to a BatchMan interface. This makes it possible to distribute the number of child jobs to the individual instances of an SAP system.

Example of an interface deletion for deletion job: BatchMan *Function Tree (J5HEN) → Customizing → Interface Customizing → Define Interfaces.*



For each interface, the job muster of the jobs to be monitored is defined (here ARV*DEL* since it is to be used not only for one archiving object, but for all archiving objects).

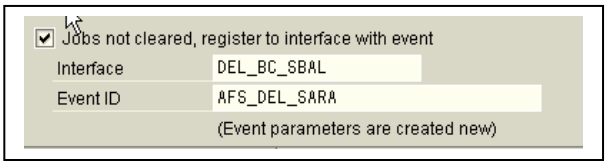
The load for the individual instances is set via the instances. That means, how many jobs may/should run for an instance in which operating mode.



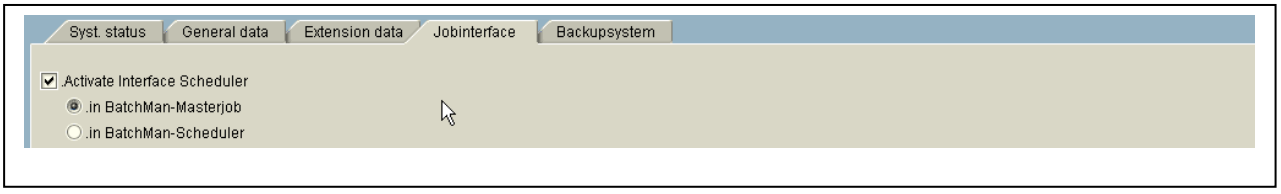
Here, only two jobs at the most should run for the central instance. Also in the NACHT operating mode if three batch processes are available.

Maintain the values in BatchMan: Enter interface.

Determine with which event the jobs are to be started. So that not all jobs immediately start upon triggering the event, the event parameter is used. The parameter is formed from the job name and the job count upon registering the jobs at the interface.



So that this function can be used, the processing of the interface is to be activated in the BatchMan setup.



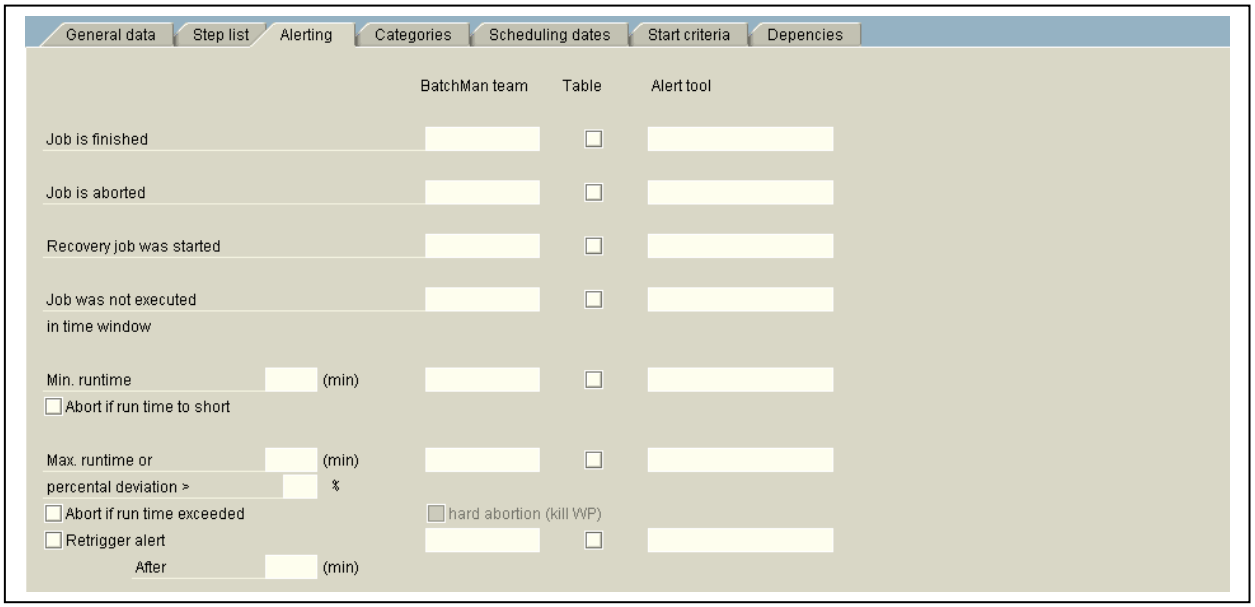
In the Customizing of the archiving object, the scheduling of the deletion job is to be set or left set to "*no scheduling*" upon writing. Also if the jobs are started at the interface via event, the "*after event*" option is NOT to be set in the archiving Customizing for the deletion jobs. The deletion jobs are scheduled by BatchMan via a function module.

2.3.6. Alerting/ Monitoring Run Time

Alerting can be triggered for several system conditions. If you correspondingly enter a **BatchMan team** and **alert tool**, messages are sent for the jobs.

The teams are maintained via the ***BatchMan-Function tree → Customizing → Basis Customizing → Team-/ Recipient Definitions***.

The documentation **BM Alerting** describes the customizing of the different alerting types (alert tools), using examples.



Job is finished (positive alert): is in general used more for informing individual users or user groups that the processing was successful. This is not important for the job control, because a reaction usually only is required in case of process abort.

However, if a certain process should be finished at a fixed time, then a positive alert might be useful. In this case, more emphasis should be given to checkpoints rather than milestones. A positive or a negative alerting can be deposited in a checkpoint as milestone.

Job is aborted (negative alert) is used for the alerting of aborted jobs.

Recovery Job was started is triggered when the original job was aborted and a job is indicated in the field recovery job.

Job was not executed in time window is triggered when the job was not started up to the time when it was defined in the start criteria (no start after). This way a future monitoring of late processing is made possible.

Alerting for **minimal runtime** is triggered when the entered minimal run time was too short. If the flag **, Abort if run time to short'** is set, then the job will be set to 'aborted'. If the flag is not active, then only the alerting is triggered.

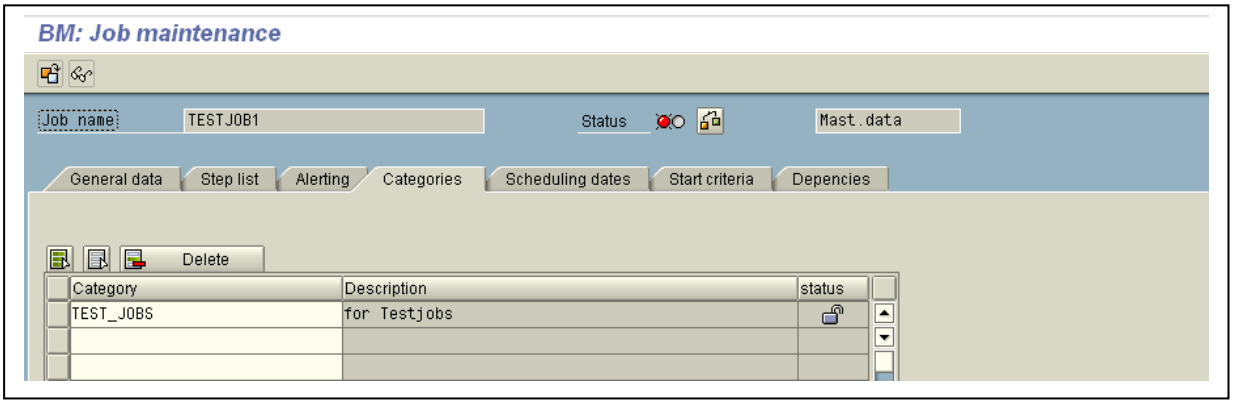
The alerting for **maximum run time** is triggered when the entered maximum run time has been exceeded. If the flag ‚**Abort if run time exceeded'** is set, then every effort is made to abort the job hard/break it off hard from the system.

Note: The system indeed tries to abort the job. However, since handling was changed by SAP, an abort is not secured. During the selection process from the databank it is hardly possible to abort a job during/at that time.

If the flag is not active, only alerting is triggered. This alerting can be triggered again after a second time period (also with different alerting). Then the alert is triggered n times x minutes.

You can find more information on alerts in the BatchMan Alerting – Manual.

2.3.7. Categories



Here you can assign categories to jobs. With the categories, you can classify jobs.

For the jobs with the categories, you can also:

- define maintenance windows,
- stop and start manually via the objects.

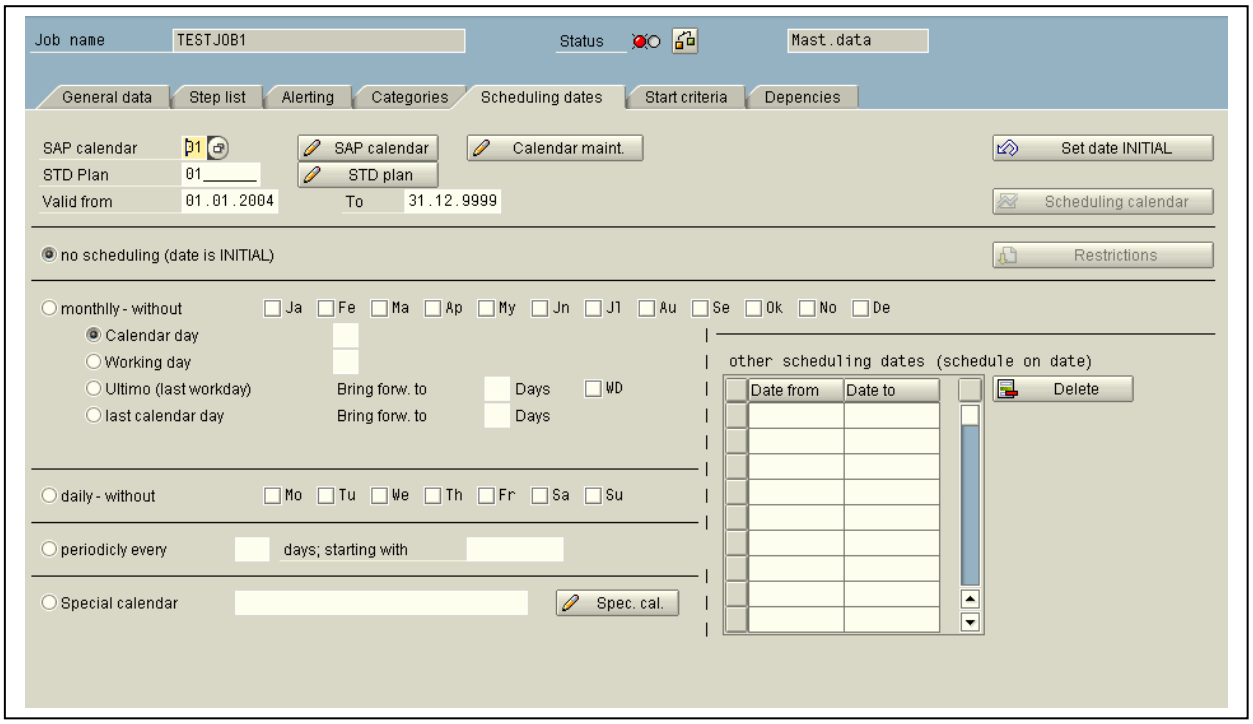
2.3.8. Scheduling Dates

The periodic scheduling (periods longer/same as one day) are maintained with the file card **Scheduling dates**.

The operator order of the order type „**Standard**“ searches, based on the specified scheduled day in the operator order, for all jobs in which:

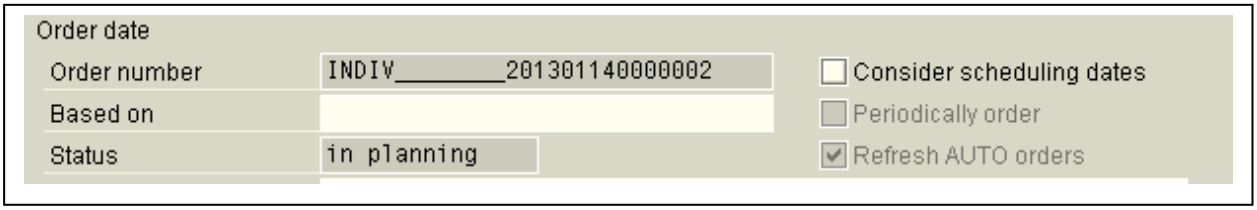
- The field „**Automatic scheduling**“ was marked,
- The **Scheduled date** of the job is in agreement with the scheduled date of the operator order
- The appropriate **Plan** is assigned and
- The scheduled date is within the **Validity range (Valid from – valid to)**

Jobs selected in this way are automatically entered in the order graphics display and combined with each other. The jobs are scheduled so that the predecessor at in the beginning, and the successor at the end of the order.



The shortest time period for the start date in the master data is ONE DAY. Shorter time periods are possible with the order type ‚WDH‘(repeat order).

In INDIV and WDH /AUTO orders, the scheduling dates are taken into account if the “Consider Scheduling Dates” flag is set for the order maintenance.



Calendar

The system can recognize work days and holidays with a help of a factory calendar. Factory calendars allow the creation/customizing of start restrictions based on state/country or plant specifications/differences. Entering a calendar is mandatory.

Factory calendar maintenance can be accessed in the following way:

- Transaction-code SCAL
- Button **'SAP calendar'** in the maintenance of the periodicity (internal call of transaction SCAL)
- Function tree BatchMan® → Tool-Box → Calendar tools → SAP Factory calendar(SCAL)

In addition to the option available in transaction SCAL of tediously maintaining special periods, there is also the option via the "Maintain calendar" pushbutton to toggle individual days between working days and non-working days by double-clicking and flagging. (detailed description can be found under *Function tree BatchMan → Toolbox → Calendar Tools → Maintain factory calendars)*

STD-Plan

The plan determines in which STD-type order the job is included. The individual plans are maintained in the BatchMan Customizing. Using the **STD plan** pushbutton, you can branch from periodicity maintenance directly to the Customizing of the plans.

Valid from - to

The validity period determines when (‚from') or until when ('to') a job is considered for scheduling.

Periodicity of job scheduling

There are several options available for periodicity of job scheduling:

No scheduling (Date is INITIAL)

This field should be marked if the job is not supposed to run by itself, but within a job net. It then receives its scheduling from the higher ranking network.

This option should be chosen for all jobs which run in orders of the type 'WDH'.

MONTHLY ....WITHOUT Ja, Fe, Ma, (JANUARY; FEBRUARY; MARCH....)

If you mark the selection button 'Monthly' the job will be executed monthly. Additionally, you can specify **calendar day**, **workday, ultimo or last calendar day** of the monthly execution date of the job by selecting one of the four selection buttons located below the monthly button.

Marking of the fields **'Ja, Fe, Ma....'** makes sure that the job will **NOT** be executed in the marked / specified months. These fields are valid for all four following possibilities:

1. If you enter ***„calendar day X"*** and the corresponding day, then the job will be executed monthly on the entered day. The execution can still be affected with the button ‚restrictions'.

2. *Workday: X*

The field 'Workday: x' means, that the job is executed each month on the x th workday.

Example: June 1996

1 st Workday	June 03, 1996	Scheduled on ⇒ June 03, 1996
20th Workday	June 28, 1996	Scheduled on ⇒ June 28, 1996

3. *Ultimo – postb. backw. to X days*

Ultimo' is defined as the **last workday** in a month. By marking the field '**Ultimo - postb. backw'**, the job will always be executed on the last workday of the month or X calendar days before ultimo. If the field 'workday' is selected in addition, the former specification '...**Ultimo - postb. backw** ' only refers to workdays.

Example: June 1996

⊗ Ultimo	⇒ Scheduled on June 28, 1996 (last workday in June of 1996)
⊗ Ultimo - change to **7** days	⇒ Scheduled on June 21, 1996
⊗ Ultimo - change to **7** days **X** Workday	⇒ Scheduled on June 19, 1996

4. *Last calendar day- postpone backw.*

The job will be executed either on the last day of the month or on the x-day 'postponed backw.'

DAILY.....WITHOUT Mon,Tue, Wed, Thu, Fri, Sat, Sun

Selecting the field 'Daily' will execute the job **every** day.

EXCEPTION: If fields from the list of weekdays located nearby are marked, the job will **NOT** be executed on the marked weekdays

PERIODIC EVERY..... X DAYS, starting with

Marking of this selection field will cause the job to be executed periodically, corresponding to the specified entered number of calendar days in the nearby field. To find the execution day, the **reference date (…days, starting with….)** is used. The scheduled date is therefore derived from reference date and periodicity.

Example:

Execute every day	Enter ⇒'1'
Execute every second day	Enter ⇒'2'
Execute every third day	Enter ⇒'3'

Note: The field "Periodic every x- week " was dropped. If the job should be executed in weekly intervals, just use ‚Periodic every X days' and multiply it by 7 (seven).

Example:

Execute periodic every week	Enter ⇒'7'
Execute every second week	Enter ⇒'14'
Execute every third week	Enter ⇒'21'

Special Calendar / Special Rule

The special calendar is based on the SAP factory-/logistics calendars and provides the tool to define your own executions.

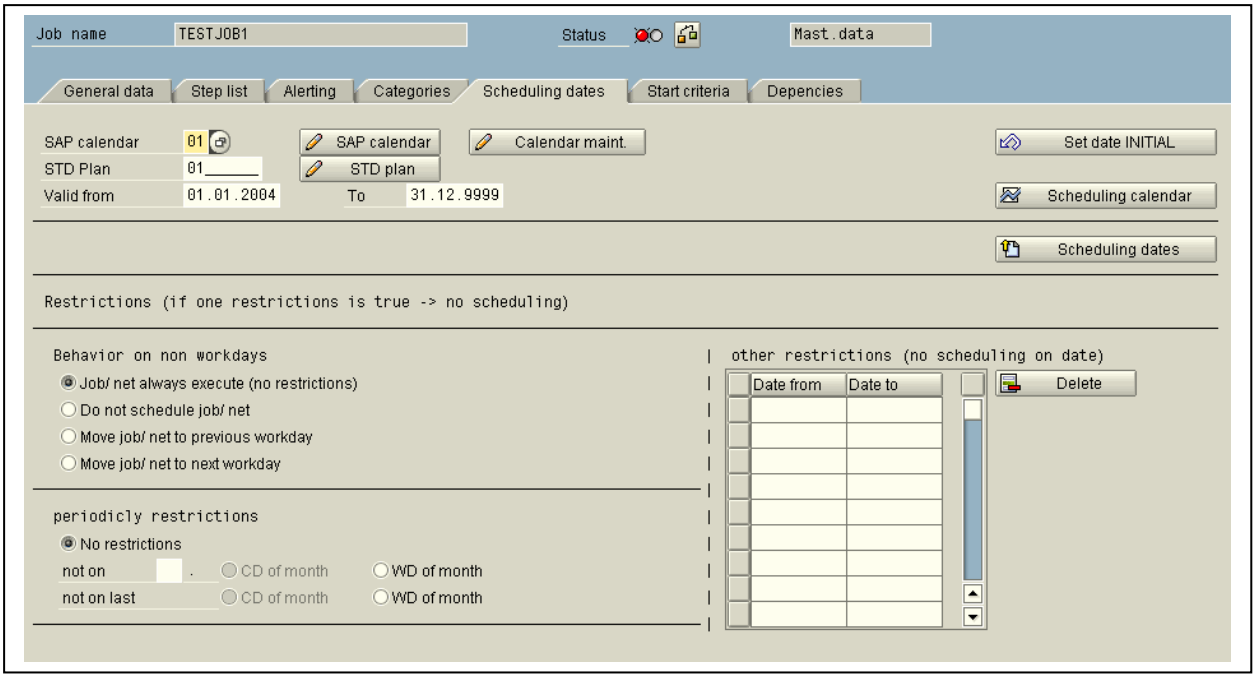
Section 6.2.6 describes how to create and maintain special calendars.

The button '**Spec. cal.**' gets you directly into the maintenance of the special calendars.

If scheduling is necessary for additional days (in addition to periodicities), these dates can be defined in the table ‚Other scheduling'.

Note: Prerequisite, however, is a selected periodicity unequal ‚no scheduling'.

If a scheduling date was maintained, then the button **Restrictions** becomes active. It calls an additional screen (instead of scheduling dates), in which start date restrictions can be entered.



Here you can specify if a job should **always be executed** on non-workdays, **cancelled, moved to workday before** or **moved to the next workday**.

Additionally, you can specify in **Periodic restrictions** that the job should not run on specified days of the month (CD-calendar day; WD-workday).

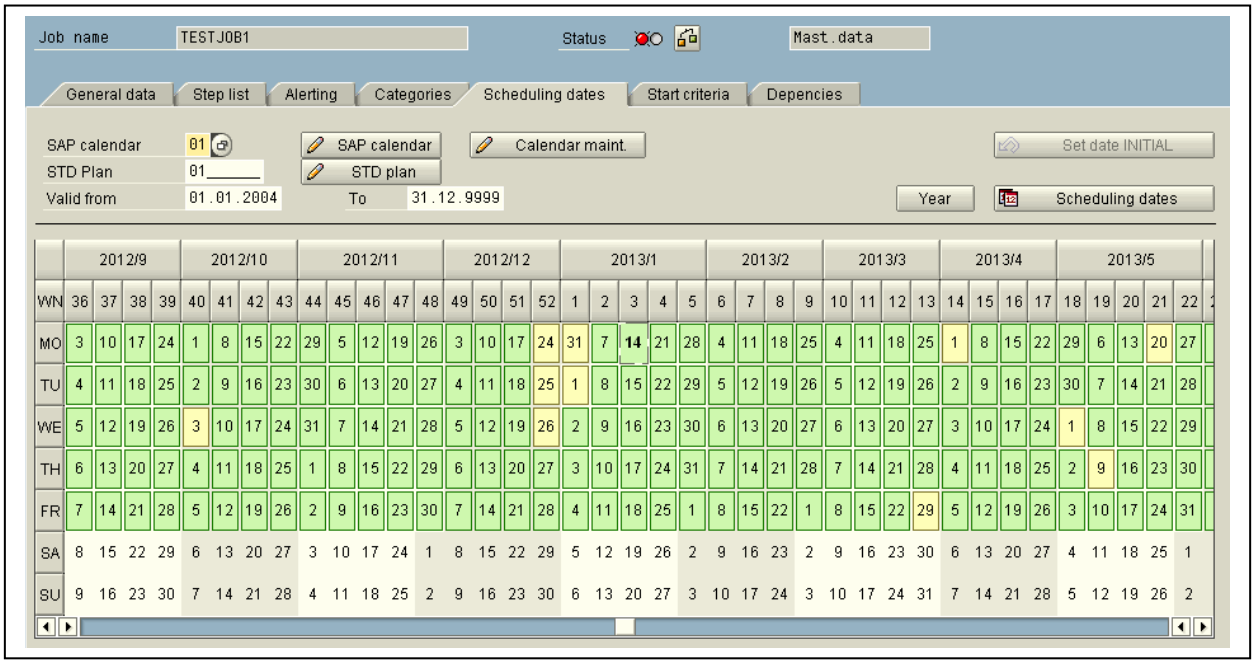
Example: A processing should run monthly. Only on the last work day an expanded processing should run which executes the month-end closing. Then, not on last WD of month' should be activated.

If more than the selected restrictions are necessary, these can be defined in the table ‚Other restrictions'.

Note: Prerequisite, however, is a selected periodicity unequal ‚no scheduling', so that the values are taken into consideration.

Via the active button **Scheduling dates** you can switch back to scheduling.

With maintained scheduled dates and restrictions, the button **Scheduling calendar** is active there also. This calendar shows a preview of the specified period values. This function should always be executed as security measure, especially in the case of special scheduling.



Example with periodicity beginning every 7 days on the 15.08.2006 and restriction not on the last work day of the month.

The display appears through the SAP calendar controls. It calculates (if not indicated otherwise) always 7 months from the current month back in the past and to the future. This prevents that the formatting takes too long.

If the view is changed further into the future with the lower runner, then the control is expanded correspondingly.

In the calendar display, the days of the SAP factory -/ logistic calendar is displayed in the following colors:

black	working day
red	non-working day

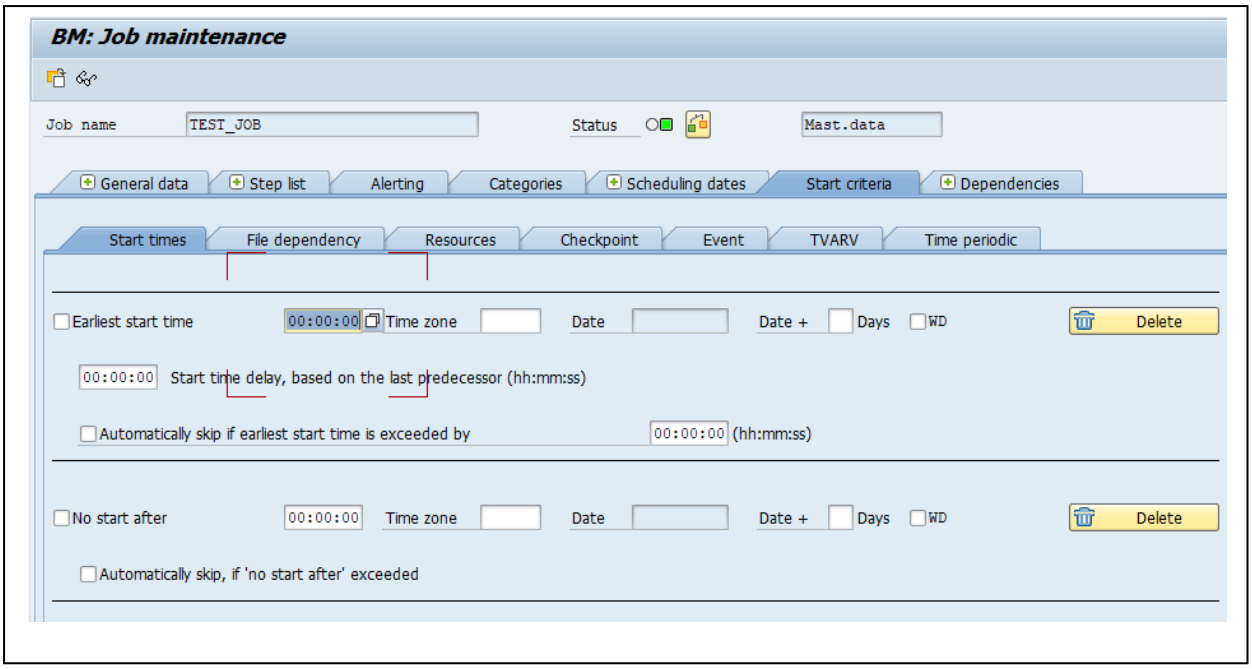
The scheduling that results from the period values are shown in the following colors:

Marked green	Scheduled on working days
Marked yellow	Scheduled on non-working days

The button **Set date INITIAL** sets the settings in the periodicity back to **No scheduling (date is initial).**

2.3.9. Start Criteria

By defining start conditions, jobs can be scheduled on more than n-predecessor relationships: the start conditions are AND-linked.



2.3.9.1. Start times

Earliest start time / No start after

In these fields, the start window can be defined in which the job may start to run.

Since the field for date should be dynamic (the appropriate reference date must be taken), it has to remain empty in general. If the setup parameter ‘flag, if earliest/ latest start date will be fixed set’ is activated, then the field ‘date’ is not ready for input.

STD orders:

The scheduled date is the deciding factor here. The order number is created from the STD plus plan plus scheduled date plus sequence number.
The scheduled date is also the date that is used to search for the objects (networks and jobs) in the master data, which have to be put in the STD order at the scheduled date.

For example, if on Wednesday a STD is generated for Saturday and also released on Saturday, then Saturday (scheduled date) is used as reference date for the calculation of the start date.

WDH orders:

Here the function is not important, because the WDH orders are not executed, always as copy via the AUTO order. Objects contained in the WDH orders should not have start dates because the dates are defined in reference to the WDH orders.

If objects are contained in WDH orders, then you receive a respective message (that objects contain set start times) at the time of checking.

INDIV and AUTO orders:

If an INDIV is created via the order administration, the scheduled date can be used as reference date here as well. There is also the option to use the function 'Submit to Batch' when you want to

create an INDIV order. And there, the current day is used as scheduling date. This way, start times are calculated in reference to the system date.

Therefore the start date of the order is used as reference date. This will ensure that the start dates will also be properly calculated for future scheduling.

Dynamic Moving of the Start Window:

Using the fields **Date plus X Days** (flag **work days**): To the date, the number of the entered days is calculated considering the flag work days (basis is the calendar from the scheduling periods) and entered in the date field for this operator order. If this function should be used, then the Setup-Parameter 'flag, if earliest/ latest start date will be fixed set' should be set to ‚X'.

Time-zones: If the fields are filled, the start time (earliest or latest) with the time zone is converted to the current system time when the job is released. Monitoring is always done with the system time!

Start time delay, based on last predecessor (hh:mm:ss): Execution time and date within programs and networks can be dynamically shifted depending on previous objects. The time shift is always dependent on the last terminated object.

The start time is set during execution. If all predecessors are finished, then the previous object that was finished last changes the start time (date and time) of the job/network; and the relative start time move/change is deleted.

This field therefore has higher priority than any preset ‚earliest start time'!!!

Automatically skip if earliest start time is exceeded by (hh:mm:ss): This defines a maximum wait time for the job/net to start. The reason for the start time delay (e.g., maintenance window, other start criteria not met, etc.) is irrelevant.

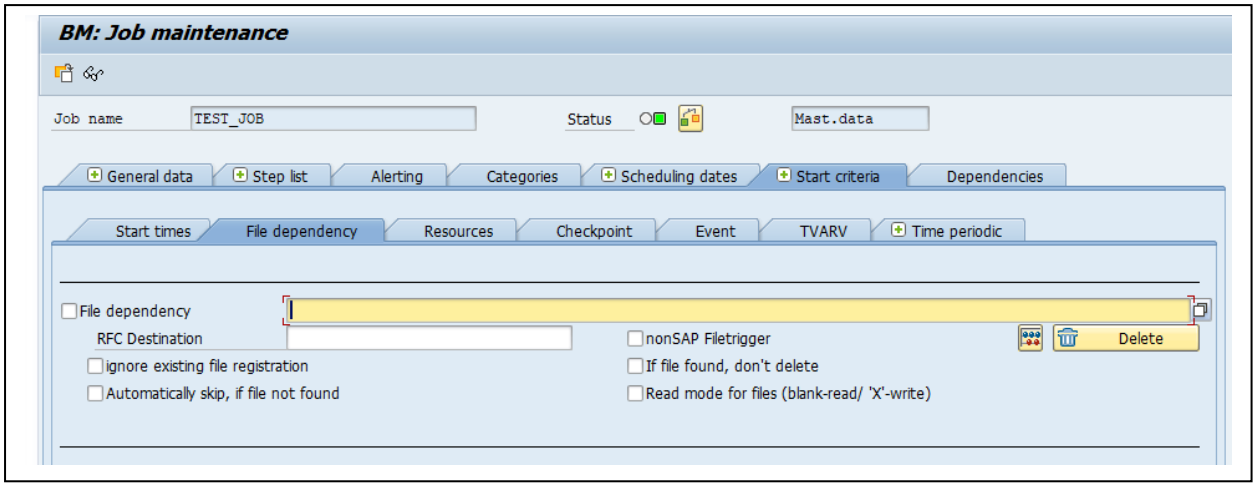
For jobs/net that are time-periodic, the start time is set dynamically based on the corresponding period. The start time exceedance offers the option of skipping a period after a defined wait time (e.g., the previous period is still active). Unlike other "skips," this skip does not stop the entire period, but merely postpones the scheduling to the next period execution.

For jobs/nets that are not time-periodic, this field should be used together with the **start time delay**, as this dynamically sets the earliest start time to the end of the respective predecessor.

Example: Wait for file for max. 30 minutes

Auto. Skip, if 'No start after' exceeded: If the flag is active and the latest start time has been exceeded, the job will be skipped. This means that the content of the job (the steps) will not be executed.

2.3.9.2. File dependencies



Here the job may only be started if the file exists in the file system. The BatchMan master job checks this in every run. If a file is found, the system tries to open this file for READING. That means that the SIDADM must have writing authorization in the file system!!!

File names can be chosen according to three different types:

- Direct entry of the file with path, name and extension –Type '**D**'. Generic entry of the file name is possible. However, only one file is processed, even if more files are prevailing.
- Servicing a file name via a TVARV-variable – Type '**T**'. For example, a file with a current date of day can be processed. The chapter about TVARV's provides more information about creating a TVARV, supplying a TVARV with rules, and updating a TVARV. Generic entry of a file name within the TVARV is possible.
- Servicing the file name via a logical file name – Type '**L**'. These logical file names have to be maintained via the SAP transaction 'FILE'.

In a normal case, the file that has triggered this order is immediately deleted after recognition by the master job. Setting the flag **'If file found, don't delete'** will prevent that.

Via the **Read mode** of a file you can determine how the file is read:

SPACE – Reading of the file in shared mode, that means no exclusive reading

‚X' - Reading in the append mode, that means exclusive reading

Of particular importance within this procedure is the ‚**OK-file'.** Following the successful transmission of the data file, the ‚OK-file' is also transferred from the source server to the file system of the target server. The target server should be used as trigger file.

It must be entered with the path in the field ‚file-dependent'. The data file is only processed by the SAP system, if the file scheduler can find the correct OK-file. After processing, the date file should be either archived or deleted.

With the **nonSAP Filetrigger** flag, file triggers can be triggered outside of the SAP system landscape via the FDM agent. Various file protocols such as SFTP, FTP, SMB etc. are available here.

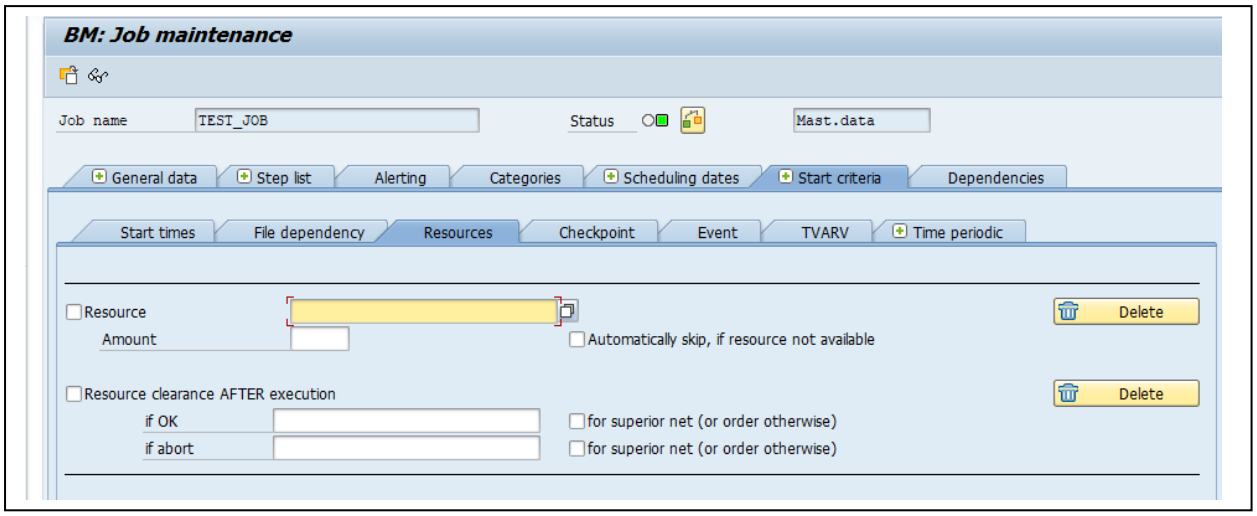
After a file trigger has been triggered, the triggering file is registered in BatchMan with the file name and last modification date. After the file registration in the BM this file or a file with the same file name and modification date will not trigger any further triggers.

If one and the same file is required as a file trigger in several different jobs, you can use the option '**ignore existing file registration**', which will trigger a file even if that file had already triggered a file trigger elsewhere and was registered in the process.

Note: If you use the flag '**ignore existing file registration**' in a periodic job/net, the file trigger will always (endlessly) fire until the file is deleted or moved.

The flag '**Automatically Skip, if file not found**' causes the file trigger to be queried exactly once at the start time of the job/net - if the result is positive, the job/net will be started, if the result is negative, the job/net will be skipped.

2.3.9.3. Resources



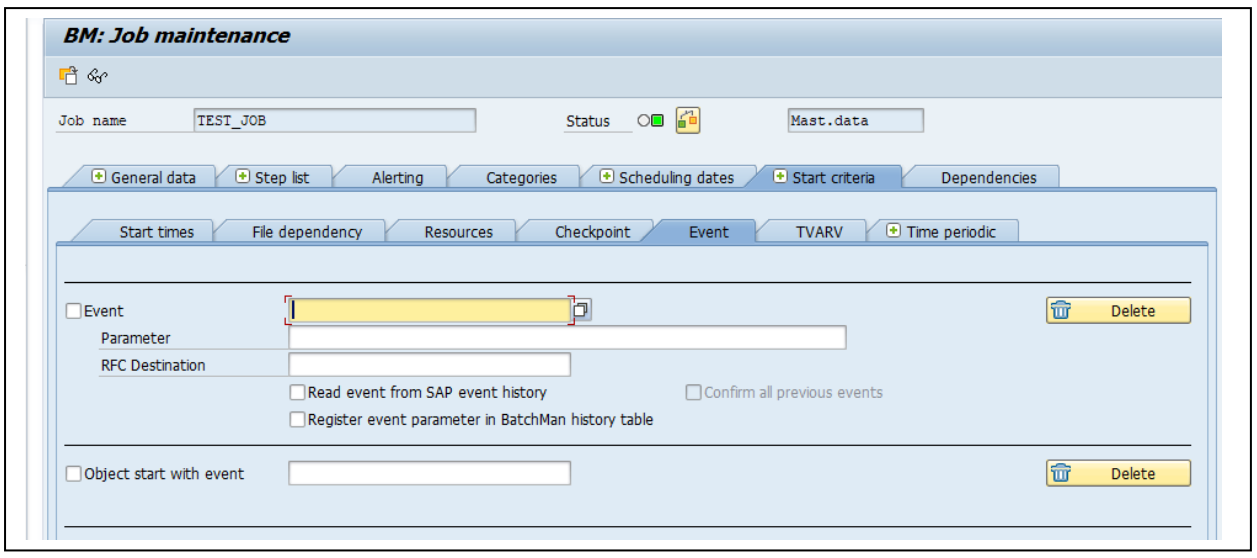
A resource is also a symbolic field, which is used by the job control for the determination/analysis for start attributes. Contrary to the checkpoints, resources are forced down at the time of job start and forced up at the job end. Resources can, for example, be used for the administration of tape drives or also for batch processes.

See the sections *Checkpoints* and *Resources* for a more detailed description. Checkpoints and resources in networks are also valid when jobs are used in networks!

The flag '**Automatically Skip, if resource not availabe**' causes the resource check to be queried exactly once at the start time of the job/net - if the result is positive, the job/net will be started, if the result is negative, the job/net will be skipped.

A job/net occupies a resource when it starts and automatically releases it again at the end of the job/net. If the automatic resource release is to take place before the actual job/net end, you can use the '**Resource clearance after execution**' function in another job/net. If this job/net is part of the network that occupies the resource in question, you can limit the search for the resource to be released to the higher-level network with the option '**for superior network**' – otherwise the entire order is searched for the resource to be released.

2.3.9.4. Event



If an external system can not communicate with an SAP System, it is possible to trigger an event in an SAP system via the operating system programs SAPEVT. With this start conditions, the appropriate event is specified with its corresponding parameter. The parameter is not absolutely required, but should be used - otherwise you have to define an event for each event.

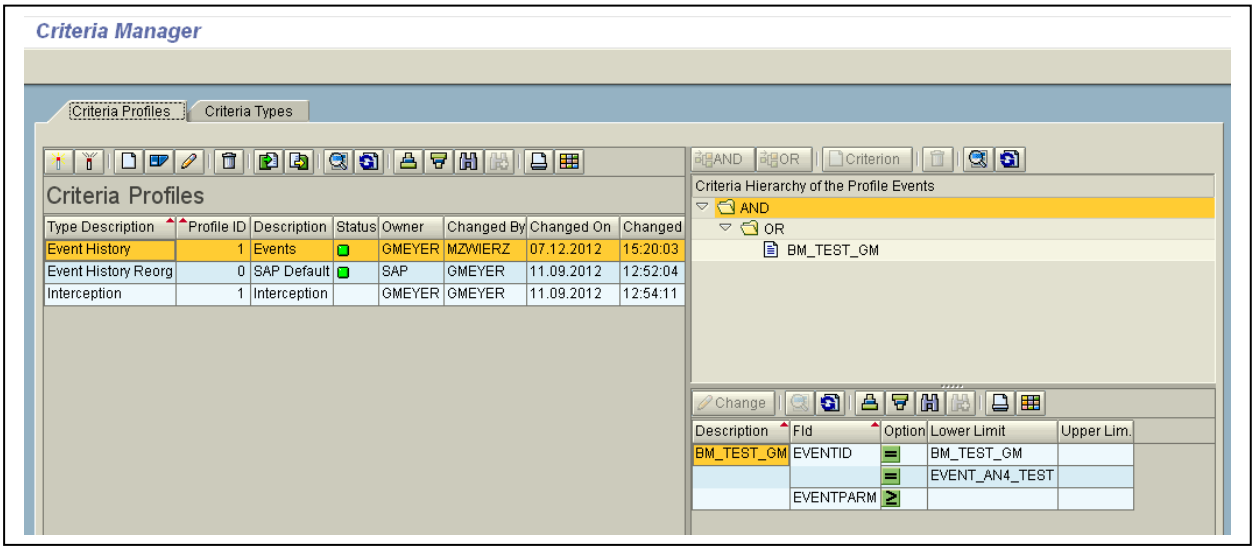
When an order is released, a physical SAP job is created for the SAP system defined by the **RFC destination**. The event must thus always be triggered in the SAP system.

If "**Read Event from SAP Event History**" is marked, the system reads the event with the parameter from the event history. No job is scheduled for this since the event history is a table in SAP.

Note: Events without SAP history are obsolete, please do not use them anymore!

In order to use the SAP event history, you have to set it up accordingly on the system that is to receive the event later (transaction SM64 or CRIT).

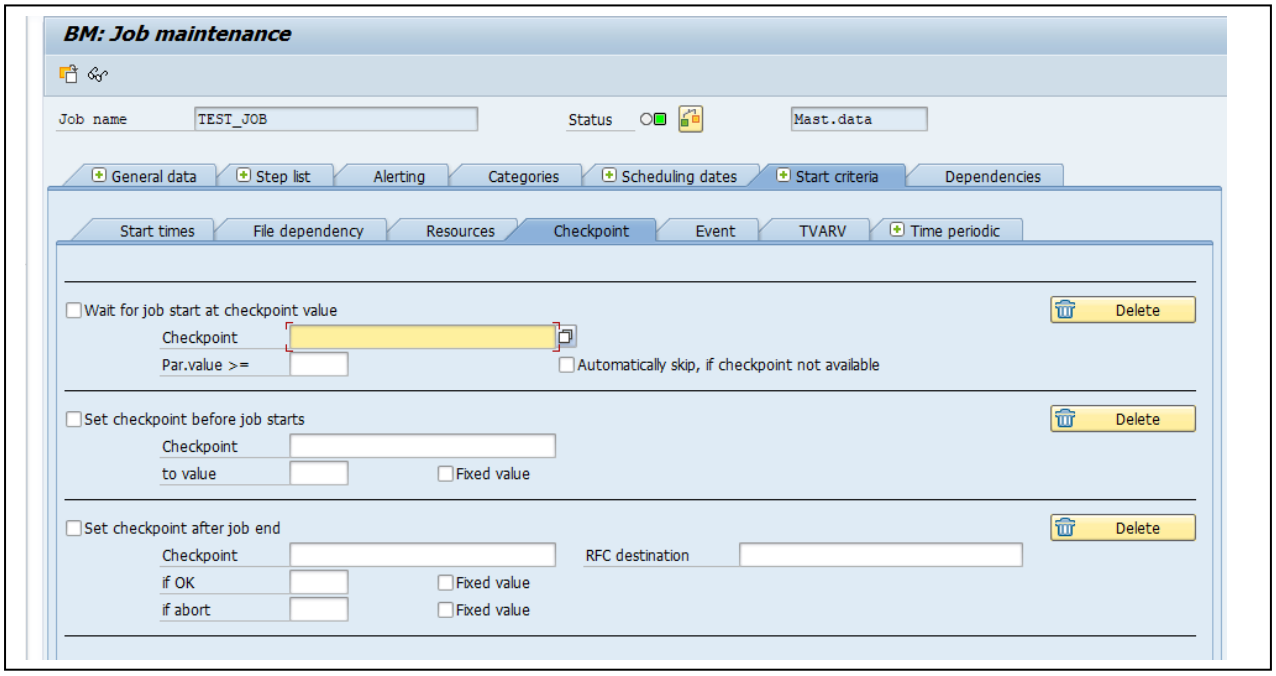
The criteria to which the SAP system should react are maintained here.



If the parameter of the event is needed for subsequent processing in SAP, you can save the parameter in a BatchMan table and transfer it to a job again within the processing (register event parameter in BatchMan event history).

The event to be provided (object start with event) must then be different than the causal event.

2.3.9.5. Checkpoints



A check point is a symbolic field, which is used by the job monitoring for determining start characteristics.

Checkpoints are mainly used to

- Trigger jobs network-wide (also system-wide)
- define certain states/conditions in the system (Usage as milestones with the alert possibility)

Wait for job start at checkpoint value: A job can be parameterized in such a way that it only **starts** when a checkpoint contains the correct or higher value.

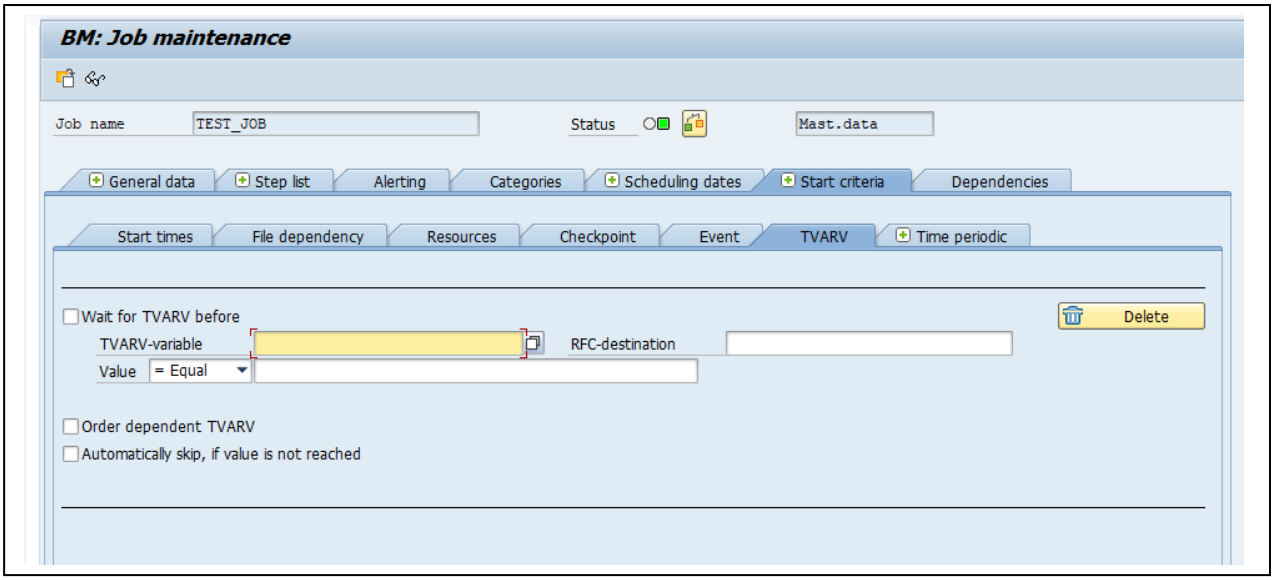
A job can change the value of checkpoints.

The flag '**Automatically Skip, if checkpoint not availabe**' causes the check if the checkpoint is due to be queried exactly once at the start time of the job/net - if the result is positive, the job/net will be started, if the result is negative, the job/net will be skipped.

Set checkpoint BEFORE job starts: Before scheduling or starting of the job, the checkpoint gets a new value (fixed value) or it is increased (negative value lowers the value of a checkpoint).

Set checkpoint AFTER job ends: After the job was finished, the option exists (depending on job success or failure) to assign a new value to a checkpoint (fixed value) or to increase (negative value lowers the value of a checkpoint)it.

2.3.9.6. TVARV



With the TVARV start criterion, the job will only start if the specified TVARVC variant has the corresponding value on the relevant system. This is checked by the BatchMan master job for each run.

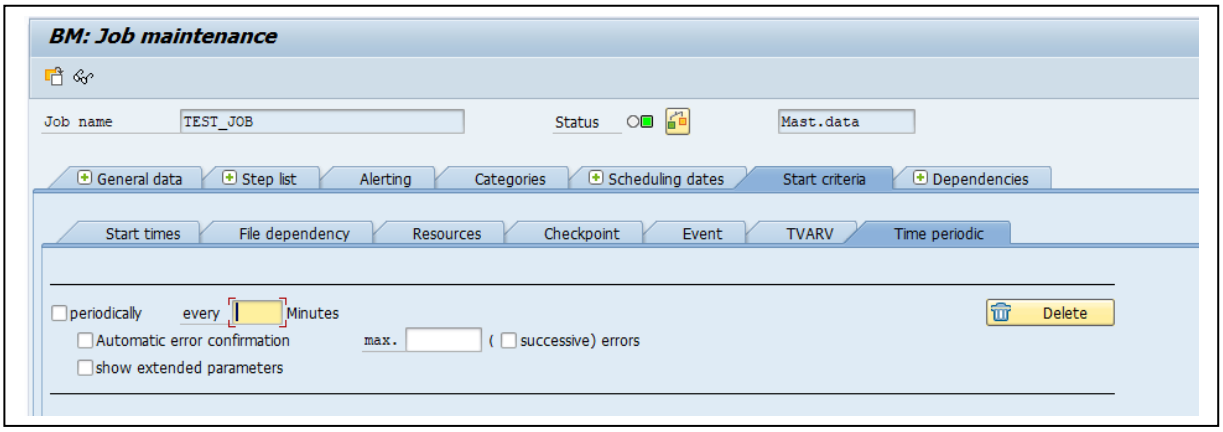
The following comparisons are possible:

=	TVARV-LOW_VALUE = [Value]
>	TVARV-LOW_VALUE > [Value] (Note: Alphanumeric comparison! „5“>“2000“)
<	TVARV-LOW_VALUE < [Value] (Note: Alphanumeric comparison! „1000“<“90“)
<>	TVARV-LOW_VALUE np [Value]
Regex	TVARV-LOW_VALUE matches pattern [Value]
IN	[Value] in TVARV

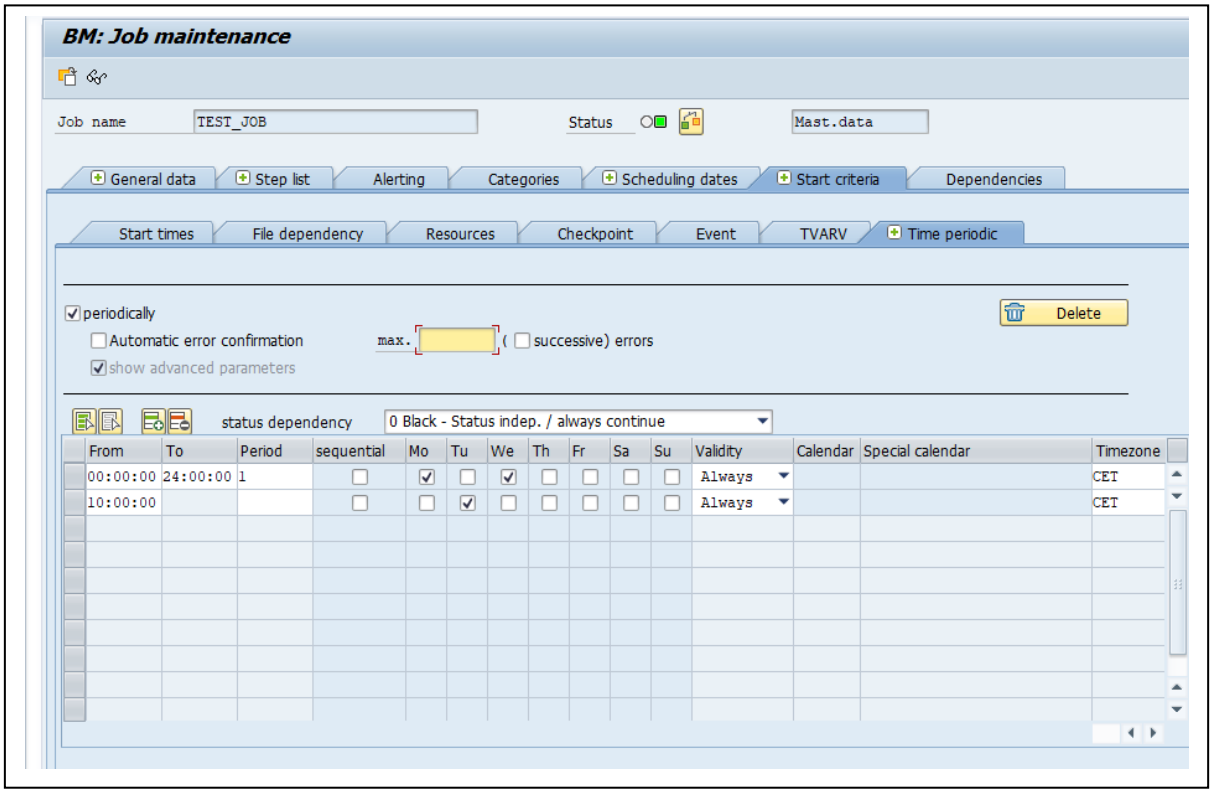
2.3.9.7. Time periodic

If a job/net is to run several times a day, the corresponding time period can be entered in the '**Time periodic**' tab (e.g. every 5 minutes).

By default, this type of period scheduling is similar to how a periodic SAP job works, which has been scheduled using the SAP standard SM36: If a job starts, its periodic successor is scheduled for the next due period. If this period time is reached, the successor starts regardless of the current status of the predecessor job. This type of execution is not protected against parallel execution, so a job can start even if the predecessor is still active.



The functionality of the periodic job scheduling can be further differentiated via the checkbox '**show extended parameters**'.



With the help of the extended parameters it is possible to define different periods at different execution times. A period can be explicitly marked as **sequential** here.

If a period is sequential, the successor does not start until the predecessor has ended, regardless of the period. By selecting the status dependency, you can also define which status the predecessor must have:

Black status independent; always on

Green status dependent; continue if successful

Red status dependent; continue if unsuccessful

If a job is only to start once at an exact time, the period is 0 and the To time field is deactivated.

The **automatic error confirmation** enables aborts to be acknowledged automatically. With the help of the various status dependencies, you can stop the job period here, for example, after 5 aborts (max. 5 errors + green line) or you do not stop the period but want an abort alarm after the 5th error (max. 5 errors + black line + alert for job abortion) etc.

Additional periodic dependencies can also be mapped in BM via time periods:

File Periodic = Time Periodic + File Dependency

Event periodic = time periodic + event

TVARVperiodic = timeperiodic + TVARV

The time period here corresponds to the period in which the start criteria are checked (e.g. check file trigger 1x per hour). For file and event periods, the time period is ignored as long as the last trigger was successful -> all files or events will always be processed.

Example: File periodic + time period 1h -> 1 file arrives at 01:30:00, at 02:50:00 2 files arrive:

00:00:00 -> no file found

01:00:00 -> no file found

02:00:00 -> 1 file found -> Start

02:00:00 -> no file found

03:00:00 -> 1 file found -> Start

03:00:00 -> 1 file found -> Start

Note: The From/To times of the time period are subordinate to the Start Times tab. A period begins at the earliest on the "earliest start time" and runs max. until the "latest start time".

Skipping always ends the periodicity! If a periodic object has successors, these only run when an explicit skip has taken place.

There are the following skip options:

- Latest start time
- File not found
- Resource not available
- Checkpoint not reached
- TVARV value not reached
- Manual skip via monitoring
- Yellow lines

Example: Earliest start time 8 a.m., latest start time 11 a.m. + automatic skip, period 30 min 00:00-24:00

08:00:00 -> Start

08:30:00 -> Start

09:00:00 -> Start

09:30:00 -> Start

10:00:00 -> Start

10:30:00 -> Start

11:00:00 -> Skip -> periods end

If a periodic object is not explicitly skipped, it runs endlessly according to its period.

To ensure that the periodic object only runs in the current STD order, the scheduling must always be transferred to the latest STD order. If an STD order is generated, the periodic execution in the 'old' STD order is stopped and started in the 'new' STD. If the type of period is set to sequential, the periodic object in the 'old' STD is also connected to the periodic object in the 'new' STD via a connector.

Example: Period 30min 23:00-01:00

STD_01012022

01.01.2022 00:00 -> Start

01.01.2022 00:30 -> Start

01.01.2022 01:00 -> Start

01.01.2022 23:00 -> Start

01.01.2022 23:30 -> Start

02.01.2022 00:00 -> Skip -> Period end (set by STD_02012022)

STD_02012022

02.01.2022 00:00 -> Start

02.01.2022 00:30 -> Start

02.01.2022 01:00 -> Start

02.01.2022 23:00 -> Start

02.01.2022 23:30 -> Start

03.01.2022 00:00 -> Skip -> Period end (set by STD_03012022)

When a STD is generated, latest start time = next period - 1 is automatically set for the periodic object in the STD of the previous day. This way, the period ends in the old STD and starts in the new STD.

The default time of the period transfer between STDs is therefore always the next period after the STD generation time. If a periodic object has defined its own latest start time, this defines the time of the period transfer.

In an INDIV order, the latest start time of a periodic object is set to 24h system time by default, unless a separate latest start time has been explicitly set.

2.3.10. Predecessor/ Successor

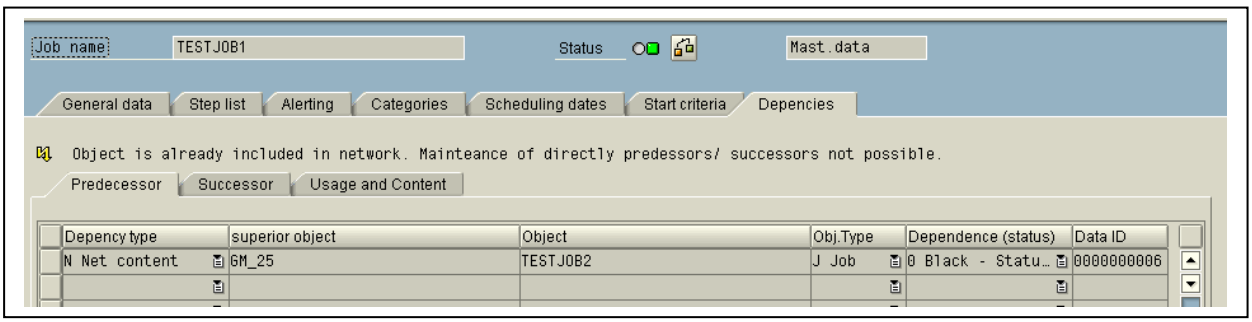
This file card is used for maintaining or displaying the predecessor and successor objects. Objects can be individual jobs or also networks.

Objects that have the same scheduling date as the current job are linked with the current job in the order for the automatic scheduling. This function only makes sense if the current job is scheduled as an individual job (flag "**automatic scheduling"** is active). Thus, it is clear that this function can only be used for the STD orders.

There is only one exception. During the scheduling of INDIV orders via the ‚Open world' interface, the data of this table is also processed.

Before this function is used, you should read those sections of the documentation that deal with dependencies within orders and networks, as well as the individual order types. If the job is already tied in higher ranking networks, predecessor and successor can only be deleted.

Visible through the notification above both tables.



The buttons for inserting predecessor and successor can be found in the appropriate table. The status of a relationship can be determined via the dependency list box.

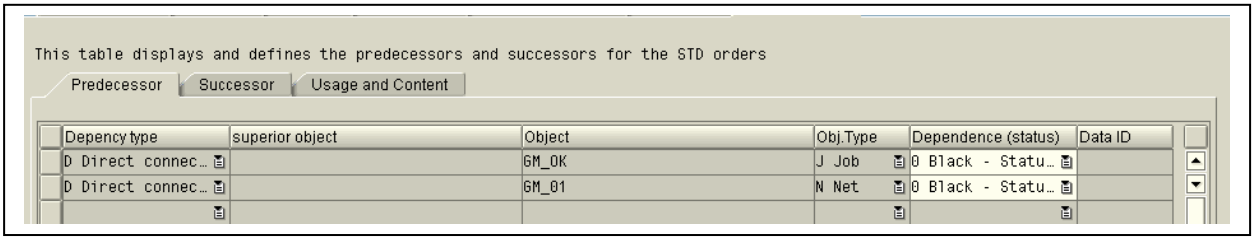
Black status-independent; continuing

Green status-dependent; continuing if successful

Red status-dependent; continuing if not successful

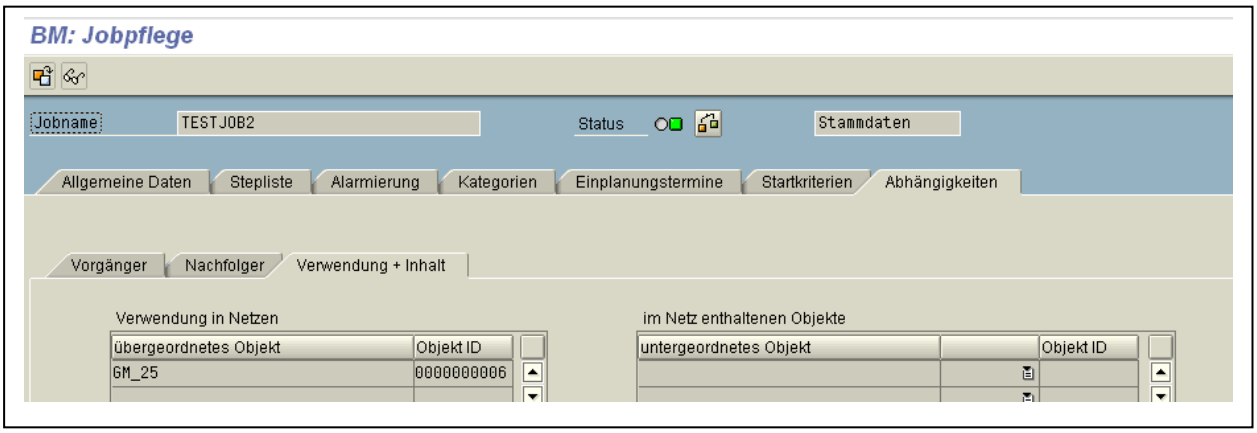
Yellow Reunion after branching

Blue Continue, ignore predecessors



The same object as predecessor and successor should never occur; this is a ring chain and should be avoided.

The system also displays the use in higher-level networks and the sub-objects included in the network (only here).



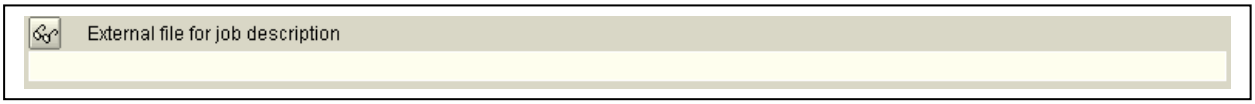
You can call and display an object by double-clicking it.

2.3.11. Documentation of a Job

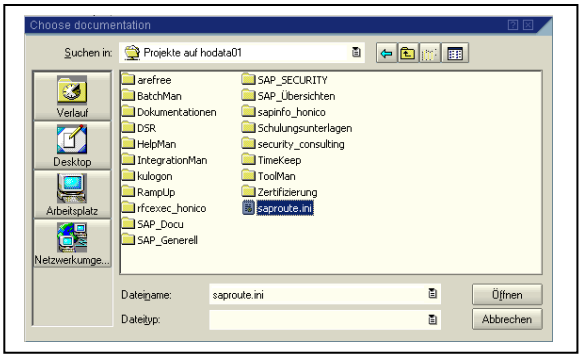
With this function, documentation can be defined with an extensive description for a job. This description is used for the operating and work preparation for the planning of orders and for the elimination of malfunctions.

2.3.11.1. GOTO → External Object Documentation

Using the external object documentation, it is possible to store links to external documents. For this, call the search help (or F4 key).

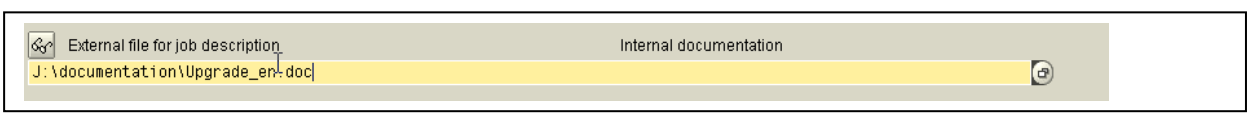


The system opens an input window where it searches for and marks the documentation in the corresponding directory. After you choose the **"Open"** pushbutton, the system saves the link in the job definition.



Which directory is used as the search directory is determined in the setup with the "**Standard directory for storage of the external object documentation"** parameter.

After the selection, it might look like this:



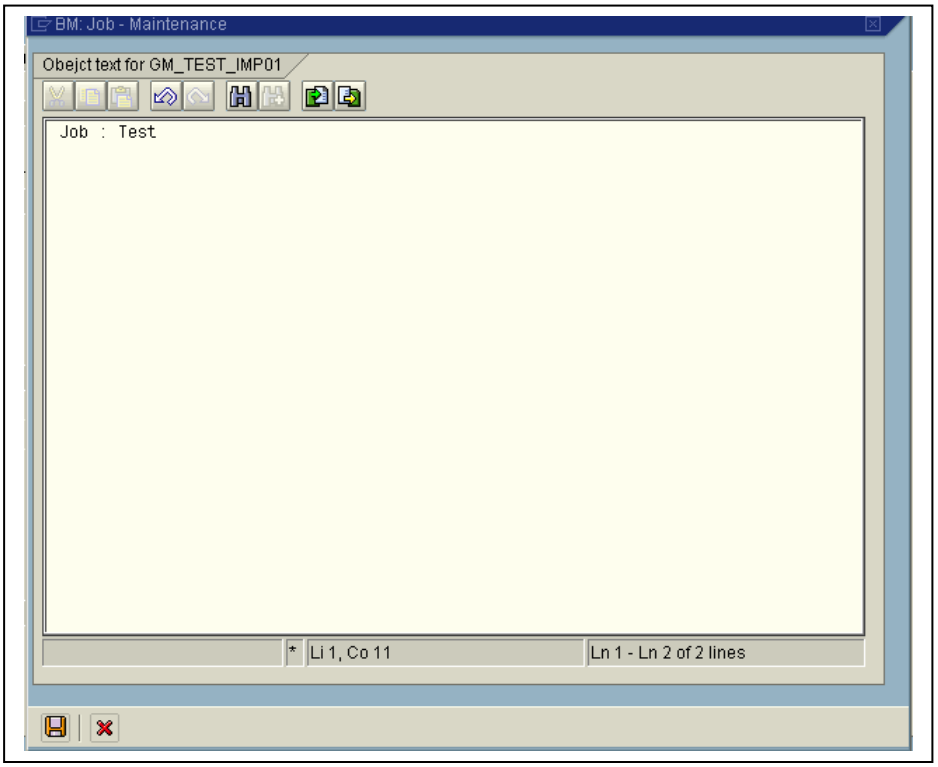
You can call the object by then choosing the menu path *Goto → Ext. object document* or the "**Display"** pushbutton.

The related links must be known to the SAP system. By default, Word and Excel files are already known.

2.3.11.2. GOTO → Object Documentation

In addition to the external documentation, there is also an internal documentation. Call via *Menu Goto → Object documentation* or the push button 

. In this case, the system calls an editor window where the documentation is entered.



It is possible in to specify the texts which are already displayed in the editor when the object documentation is called for the first time.

These text templates can be maintained language-dependent via *Function tree BatchMan → Customizing → Extended Customizing → Text Templates*

2.3.11.3. Jobdocumentation Solution Manager

As of BatchMan 4.3, it is officially possible to store documentation of the jobs and networks in a solution manager system.

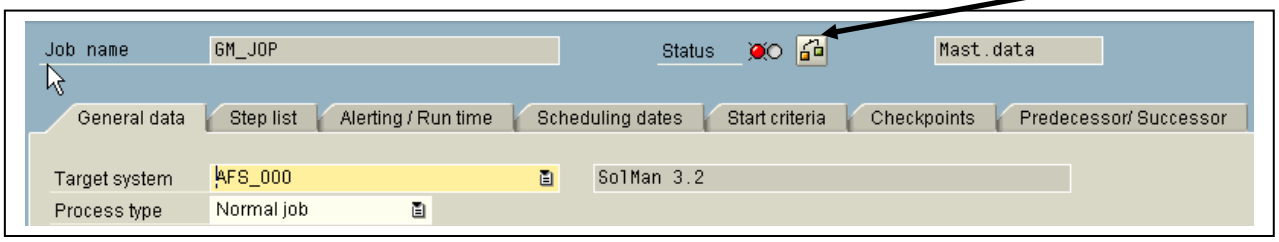
You have to maintain the related settings in the setup.

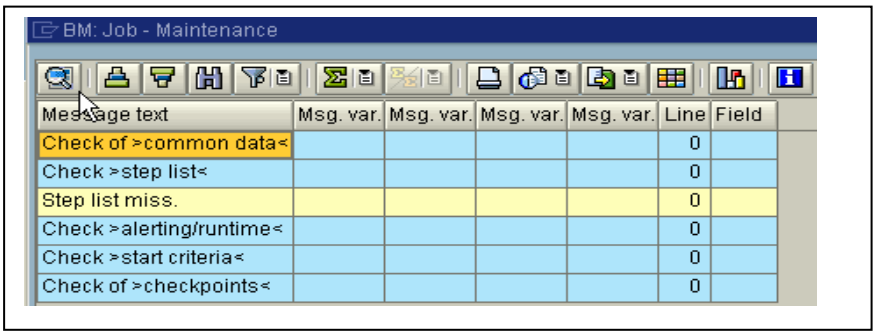
2.3.12. Additional Functions: Job Maintenance

2.3.12.1. Checking a Job

Each time you save, the entered values are compared/ checked with the SAP system (and therefore clients). If all entries are error-free/correct, the status is GREEN. In case of errors, the status is RED.

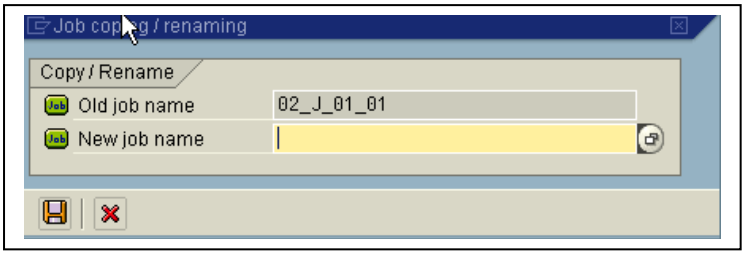
If the status is RED, then the error list can be displayed by pushing the button ,*Check status*.





2.3.12.2. Copy

Call the function to copy the job. The system opens a window to query the new job name.



2.3.12.3. Rename

Call the function to rename the job. The system opens a window to query the new job name (same as copying).

2.3.12.4. Delete

Call the function to delete the job. This can only be deleted if the job is no longer used in networks. After a confirmation prompt, the job is deleted.

2.3.12.5. Other object

Call the init mode of the job maintenance. All fields are deleted, the job name is ready for input.

2.3.12.6. Create transport request

Creates a SAP transport request, which contains the BM job definition as table content.

2.3.12.7. Send job definition

Sends the job definitions via RFC to another BM master system that can be selected via selection.

2.3.12.8. Compare job

Compares the job definitions of different BM master systems via RFC.

2.3.13. Menu Functions

Under the menu point **Job** you find the already described functions ‚*Other object'*, ‚*Create*, ‚*Change*, ‚*Display*, ‚*Delete* and ‚*Save*. The function ‚*Print'* calls an option for printing all job values. See also ‚Reporting about Jobs' below. Under **Edit** you can find the functions ‚*Copy* and ‚*Rename'*. (Already described above)

2.3.13.1. Edit → Transport Object

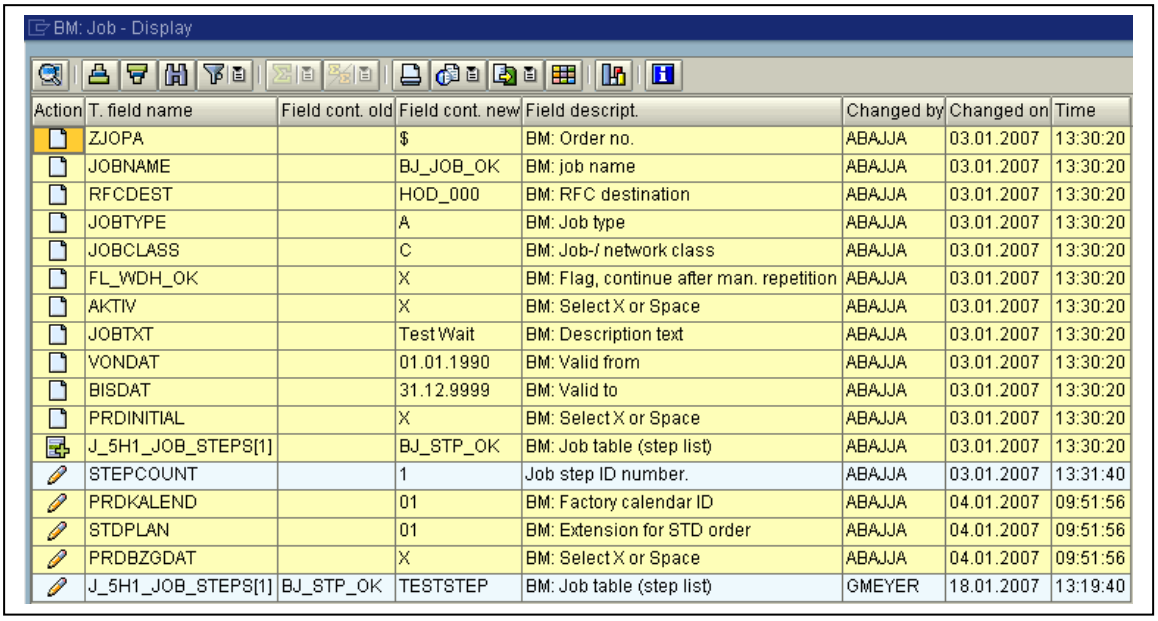
Via this menu point or the button with the truck (display mode), a job can be transported. Since this function is executed via the SAP transport mode, the appropriate authorization (S_TRANSPRT / S_CTS_ADMI) is needed.

After a confirmation prompt a window is displayed in which you select an already existing transport order; or you create a new order.

There is also the option to transport by writing (export) to a file and then uploading (import) into a second system. Also see 2.7 Transport.

2.3.13.2. Goto → Change history

Call the change history for a job. All changed fields are displayed.



Action	T. field name	Field cont. old	Field cont. new	Field descript.	Changed by	Changed on	Time
	ZJOPA		$	BM: Order no.	ABAJJA	03.01.2007	13:30:20
	JOBNAME		BJ_JOB_OK	BM: job name	ABAJJA	03.01.2007	13:30:20
	RFCDEST		HOD_000	BM: RFC destination	ABAJJA	03.01.2007	13:30:20
	JOBTYPE		A	BM: Job type	ABAJJA	03.01.2007	13:30:20
	JOBCLASS		C	BM: Job-/ network class	ABAJJA	03.01.2007	13:30:20
	FL_WDH_OK		X	BM: Flag, continue after man. repetition	ABAJJA	03.01.2007	13:30:20
	AKTIV		X	BM: Select X or Space	ABAJJA	03.01.2007	13:30:20
	JOBTXT		Test Wait	BM: Description text	ABAJJA	03.01.2007	13:30:20
	VONDAT		01.01.1990	BM: Valid from	ABAJJA	03.01.2007	13:30:20
	BISDAT		31.12.9999	BM: Valid to	ABAJJA	03.01.2007	13:30:20
	PRDINITIAL		X	BM: Select X or Space	ABAJJA	03.01.2007	13:30:20
	J_5H1_JOB_STEPS[1]		BJ_STP_OK	BM: Job table (step list)	ABAJJA	03.01.2007	13:30:20
	STEPCOUNT		1	Job step ID number.	ABAJJA	03.01.2007	13:31:40
	PRDKALEND		01	BM: Factory calendar ID	ABAJJA	04.01.2007	09:51:56
	STDPLAN		01	BM: Extension for STD order	ABAJJA	04.01.2007	09:51:56
	PRDBZGDAT		X	BM: Select X or Space	ABAJJA	04.01.2007	09:51:56
	J_5H1_JOB_STEPS[1]	BJ_STP_OK	TESTSTEP	BM: Job table (step list)	GMEYER	18.01.2007	13:19:40

The icons of the action correspond to the ones of the buttons.

2.3.13.3. Goto → Deleted Jobs

With this function, all jobs deleted in **BatchMan** can be displayed.

If a job is deleted, only one entry is created that the job was deleted. If all other change documents for this job will be deleted is dependent on the Setup parameter ‚History will be kept when if master data are deleted'. If this value is active, they can still be displayed via *Reporting → Change history → Change history jobs*.

2.3.13.4. Goto → Runtime-Histogram

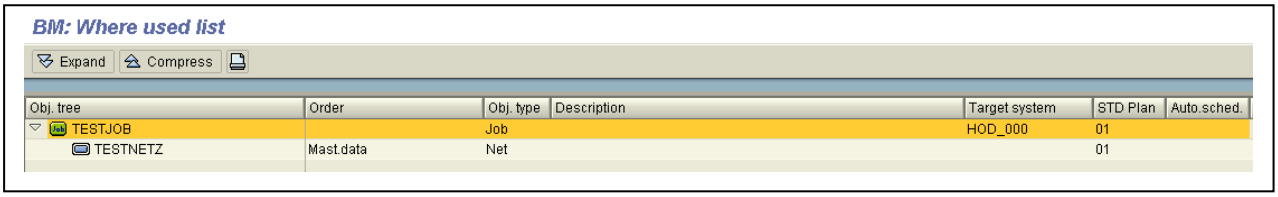
See Run time histogram (Number of runs) and also Run time histogram (Date)

2.3.13.5. Goto → Long Term History

Siehe See Number/Count table job scheduling (Long term history)

2.3.13.6. Environment → Usage only in Master Data

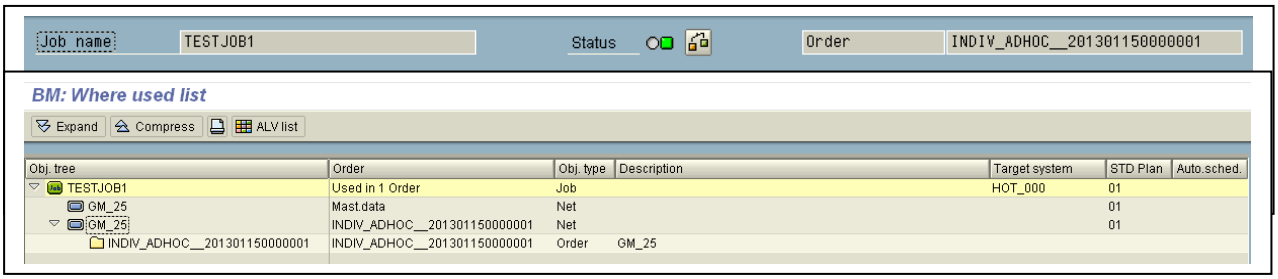
The function displays the usage of the current jobs in all networks and in upper-level networks.



2.3.13.7. Environment → Usage in Master- and Order Data

The function displays the usage of the current jobs in networks and upper-level networks, as well as the usage in the orders.

If you branch from the order maintenance to the job maintenance, the system displays the job copied for the order. This is evident when “Order” and the related order number are now displayed in the fields next to the status.



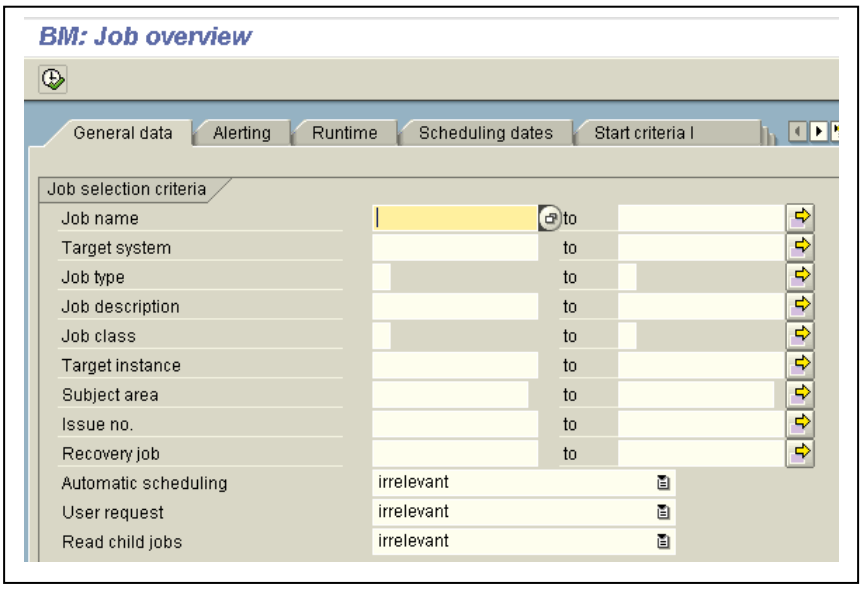
The where-used list then displays the orders.

2.3.14. Reporting about Jobs

Via jobs, the following evaluation options exist:

2.3.14.1. Job Overview

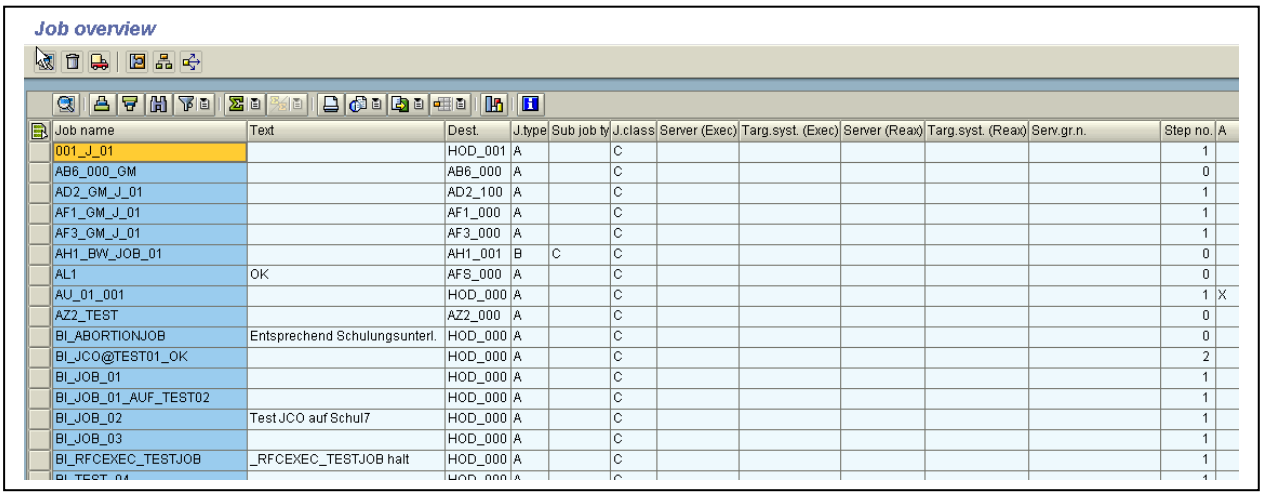
In this report, a job overview may be compiled by selective selection criteria. Here, it is possible



to choose by job type (process type).

The results list is output as an ALV list, and you can set it up as you like. You can change and save the settings using the layout management.

	Calling the job maintenance; analog to a double click on a line
	Detailed display of all job fields of a list (see Detailed list jobs); the jobs have to be marked
	Where-used document of a job; the jobs have to be marked

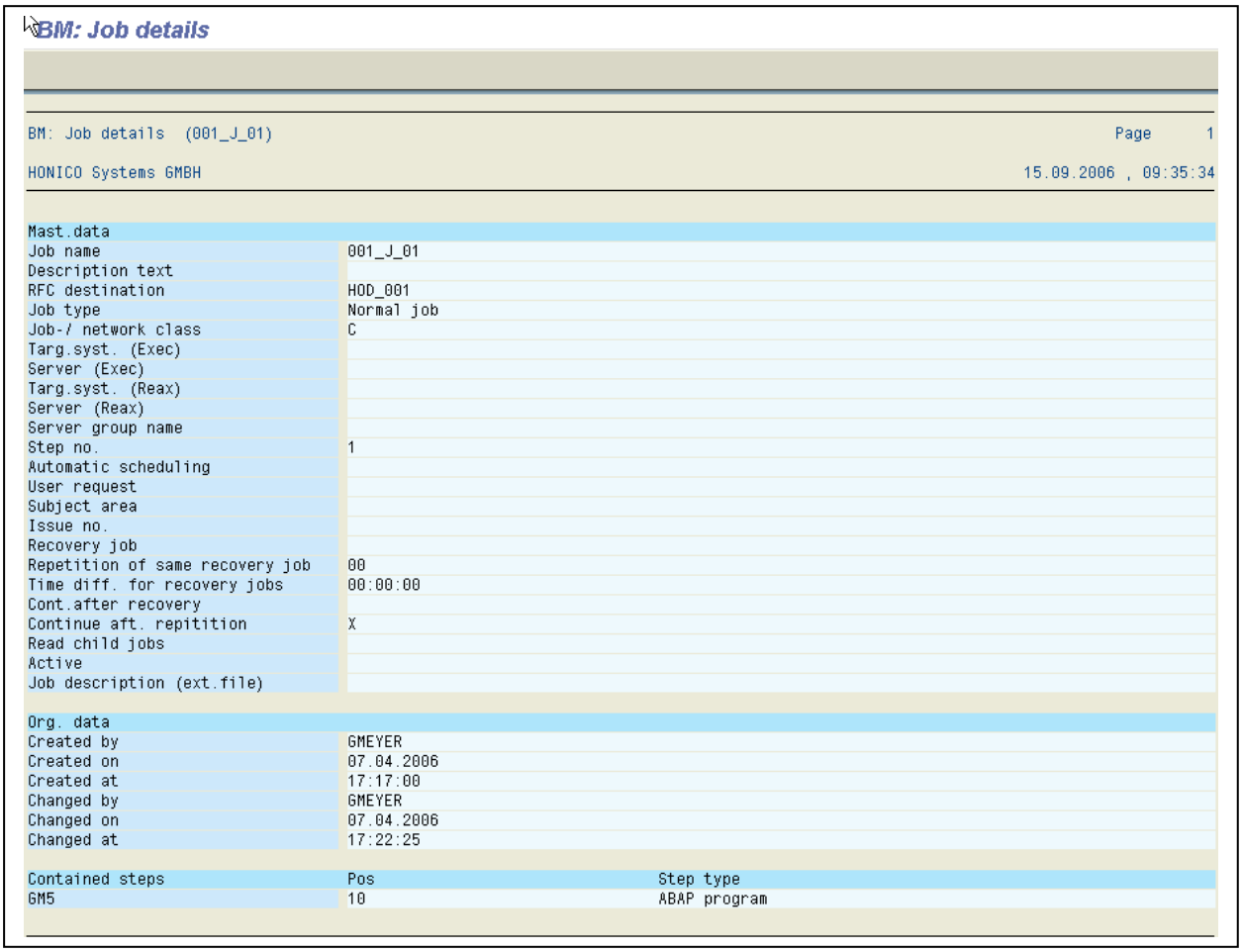


	Deleting the job (not only from the list). The usage of the job in networks and WDH orders is checked.
	Display of the marked jobs with their included steps (see Job List with Steps)
	Transport of the marked jobs via SAP transport order

2.3.14.2. Jobs: Detailed List

This function can be used for the complete printout of one or all jobs. All fields of a job are specified. Only those fields are displayed which, for example, are also relevant for the process type.

Excerpt:



BM: Job details

BM: Job details (001_J_01) Page 1

HONICO Systems GMBH 15.09.2006 , 09:35:34

Mast.data			
Job name	001_J_01		
Description text			
RFC destination	HOD_001		
Job type	Normal job		
Job-/ network class	C		
Targ.syst. (Exec)			
Server (Exec)			
Targ.syst. (Reax)			
Server (Reax)			
Server group name			
Step no.	1		
Automatic scheduling			
User request			
Subject area			
Issue no.			
Recovery job			
Repetition of same recovery job	00		
Time diff. for recovery jobs	00:00:00		
Cont.after recovery			
Continue aft. repitition	X		
Read child jobs			
Active			
Job description (ext.file)			
Org. data			
Created by	GMEYER		
Created on	07.04.2006		
Created at	17:17:00		
Changed by	GMEYER		
Changed on	07.04.2006		
Changed at	17:22:25		
Contained steps	Pos	Step type	
GM5	10	ABAP program	

BM: Job details (001_J_01) Page 2

HONICO Systems GMBH 15.09.2006 , 09:35:34

Alerting / Run time	
Pos. resp.group	
Pos. alert tool	
Neg. resp.group	
Neg. alert tool	
Rec. resp.group	
Rec. alert tool	
NoExec. resp.group	
NoEx. alert tool	
Min. runtime	00000
Abort min runtime	
Min. runtime resp. group	
Min. runtime alert tool	
Max. runtime	00000
Abort max. runtime	
Max. runtime resp. group	
Max. runtime alert tool	
Reexec. max.runtime resp. group	
Reexec. max.runtime alert tool	
Trigger alert again	
Alert new after x min.	00000
Sched.date	
Calendar	01
STD plan	01
Valid from	01.01.1990
To	31.12.9999
Periodicity	No scheduling
Restrictions	-
Other scheduling	No scheduling

Start criteria

No start criteria found

Checkpoints

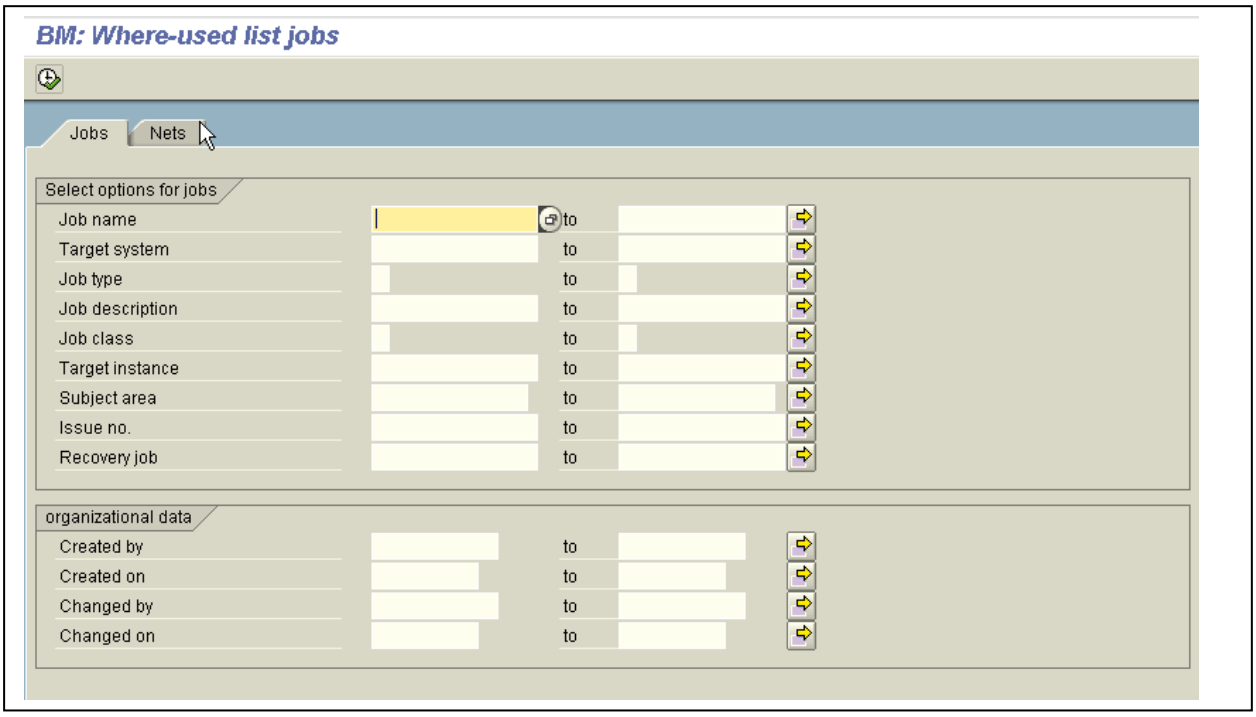
No checkpoints found

Parent

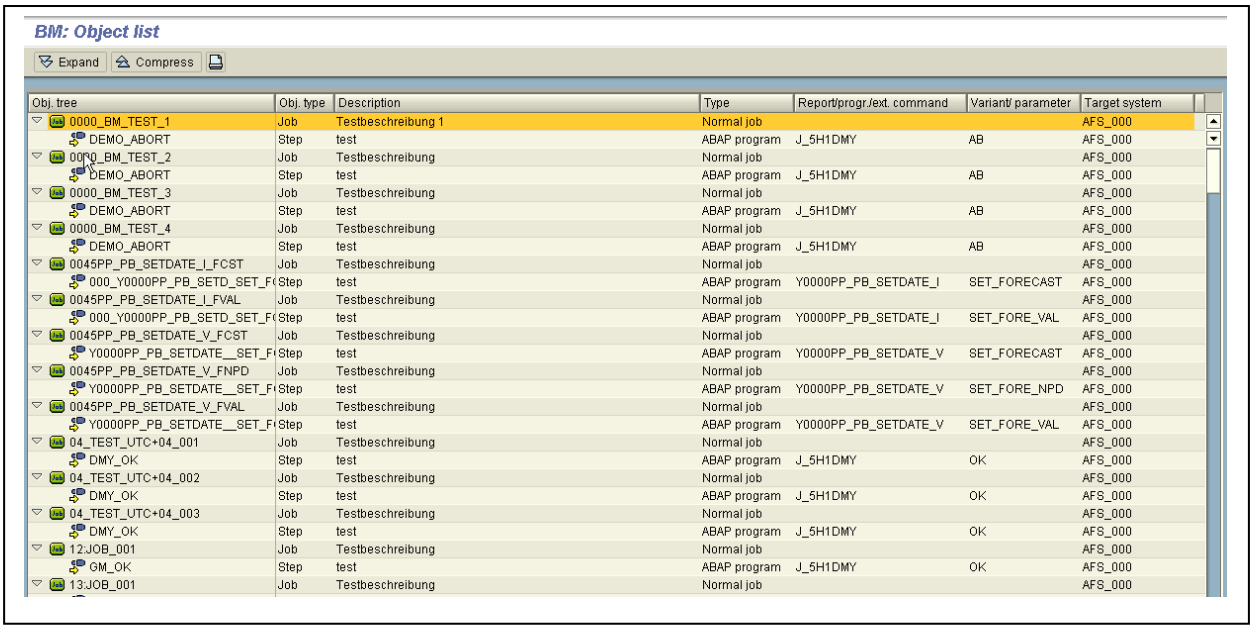
No predecessors found

2.3.14.3. Job List with Steps

Call a list for individual jobs or all jobs with their steps included (thus the step list). Processes have no step list, in contrast to “normal jobs.”



Result list:



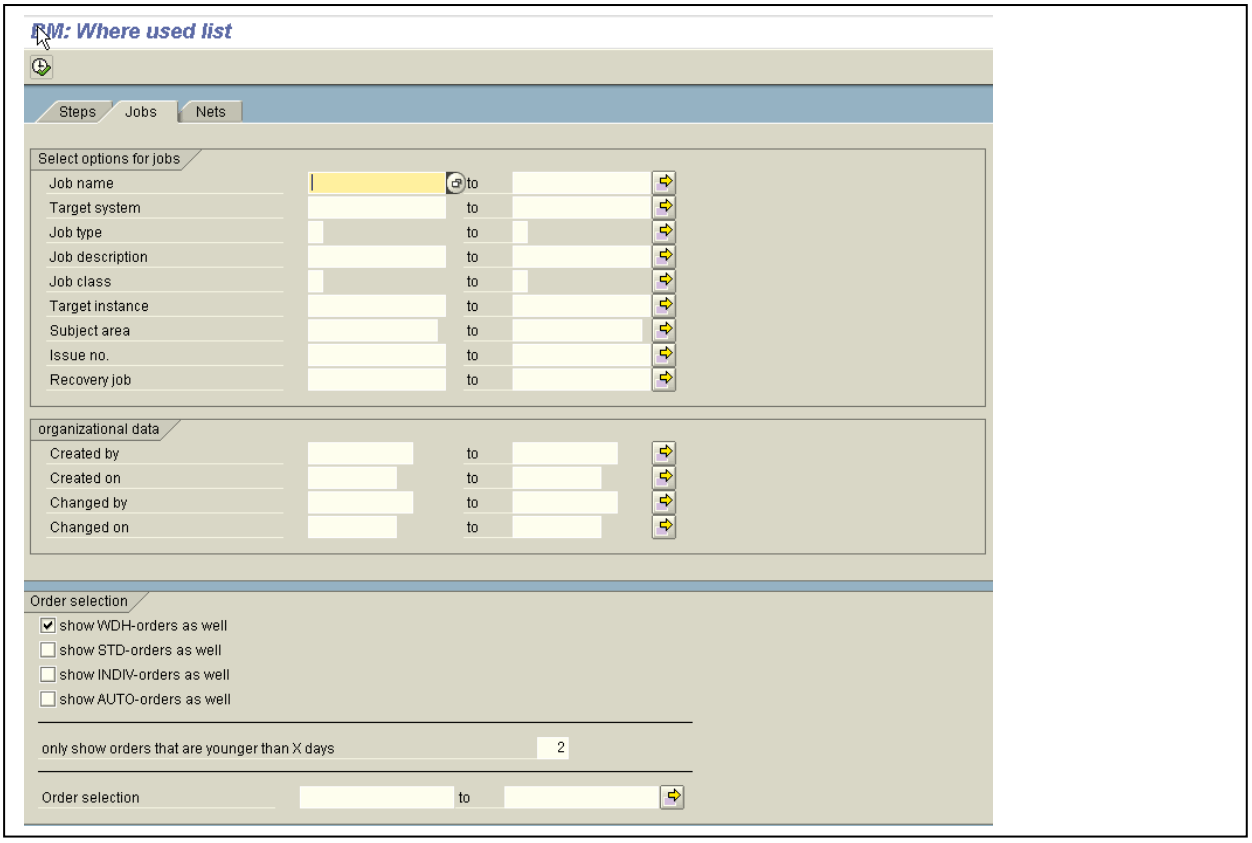
In all objects, you can jump directly to the display with a double click.

2.3.14.4. Where-Used List Jobs

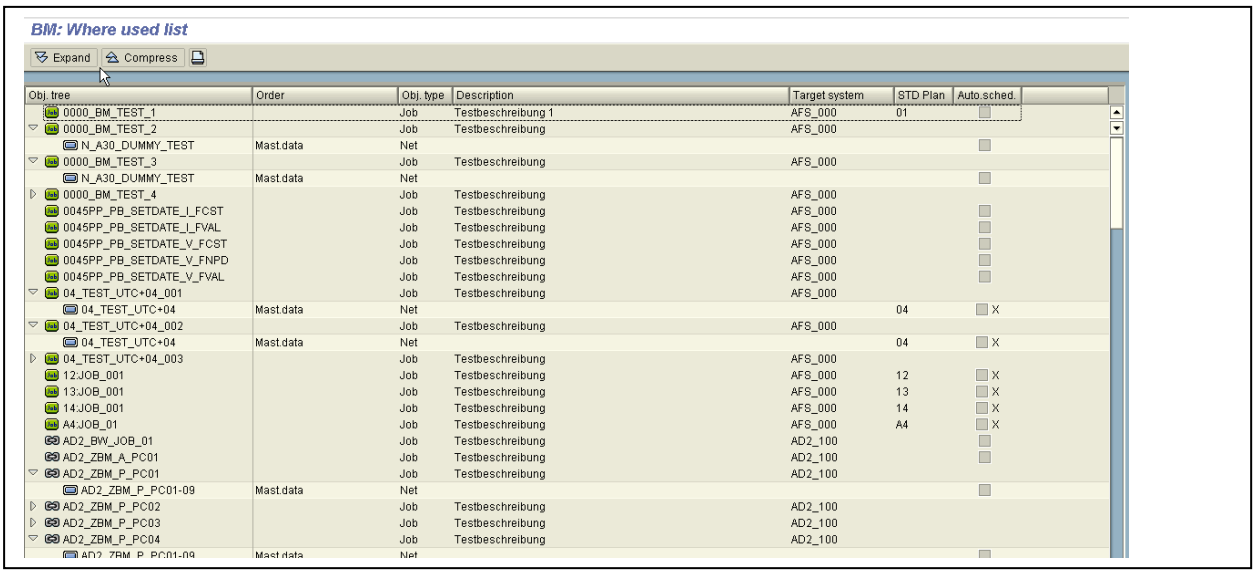
Call where-used list for individual jobs or all jobs. This is possible both for master data and orders. For this, set the "*also display orders"* flag.

Via the field for orders, you can restrict certain orders.

If only those jobs are to be displayed that go into STD orders, activate the flag "**only automatically scheduled jobs**."



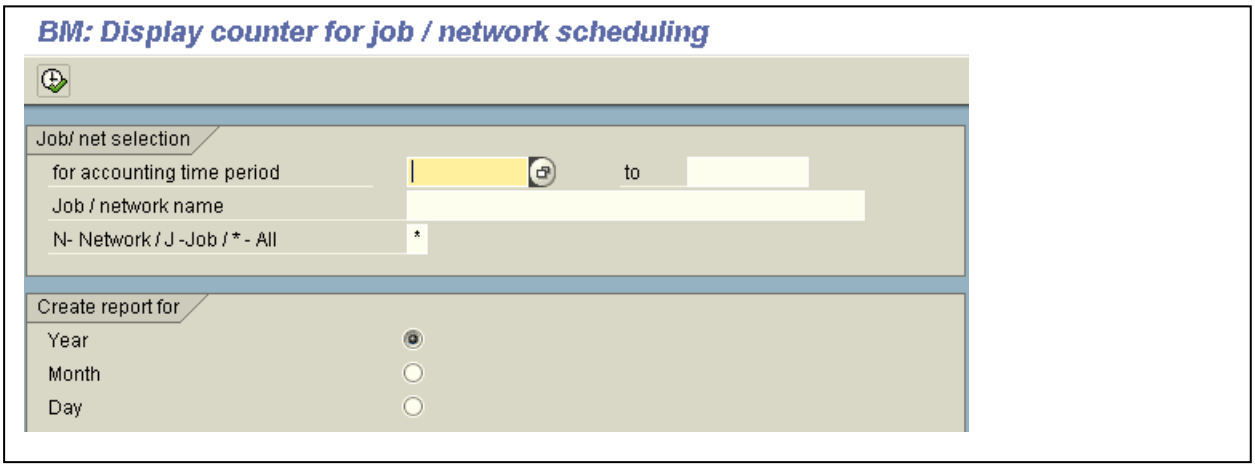
Result list:

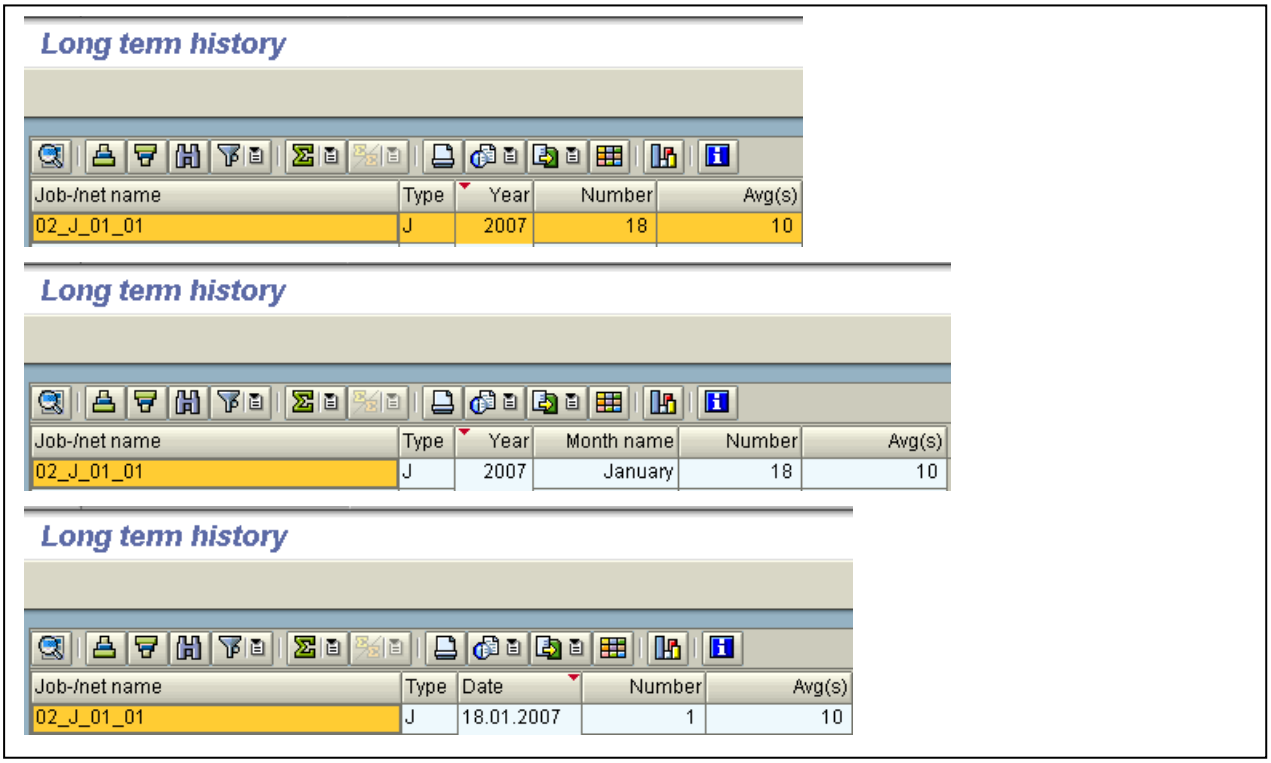


In all objects, you can jump directly to the display with a double click.

2.3.14.5. Counting Table Job Scheduling (Long term history)

This report (/BTCMAN/M_B22) shows for the selected objects (network or job), how often an object ran in the selected report period and which average run time results from it. View per year, month or day possible.



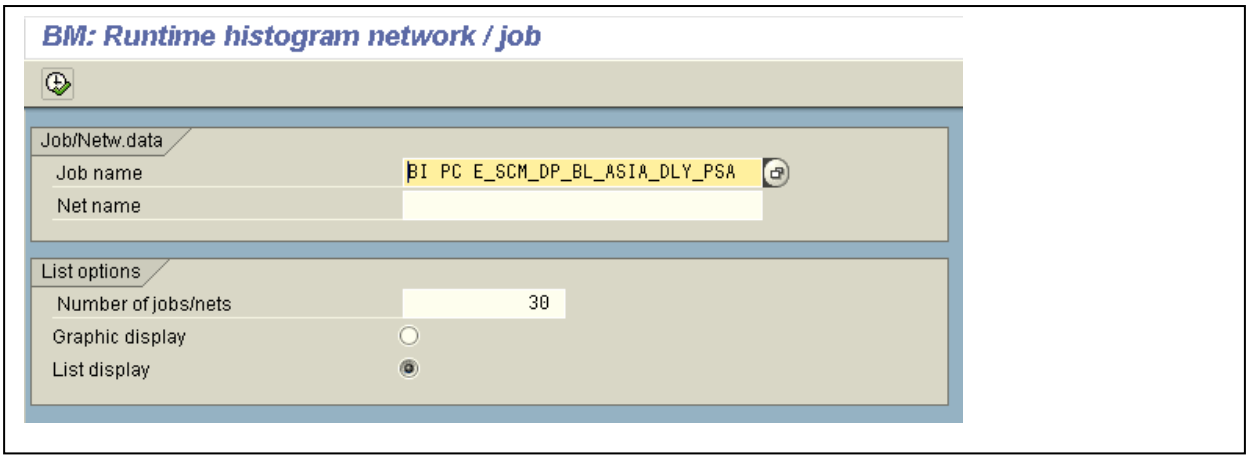


2.3.14.6. Run Time Histogram (Number of Runs)

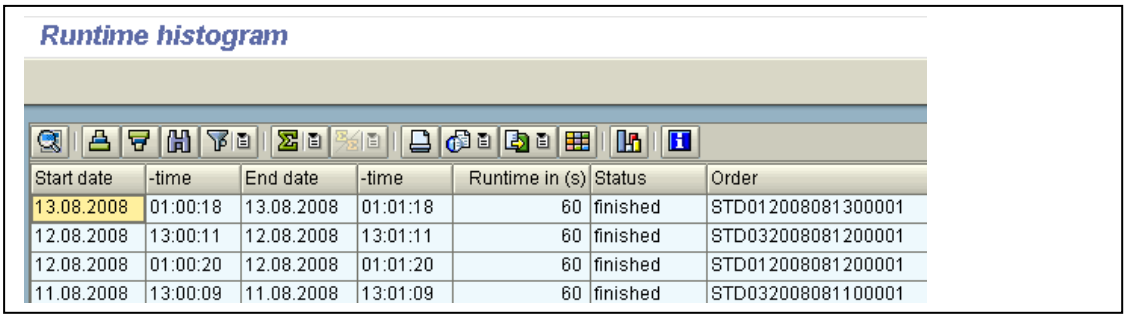
The run times of jobs can be analyzed with this report (/BTCMAN/M_LFZ).

Attention: This detailed information is only available if the orders are still contained in BatchMan; or if the option ‚*Hold runtime histogram*' was activated when BatchMan orders were archived.

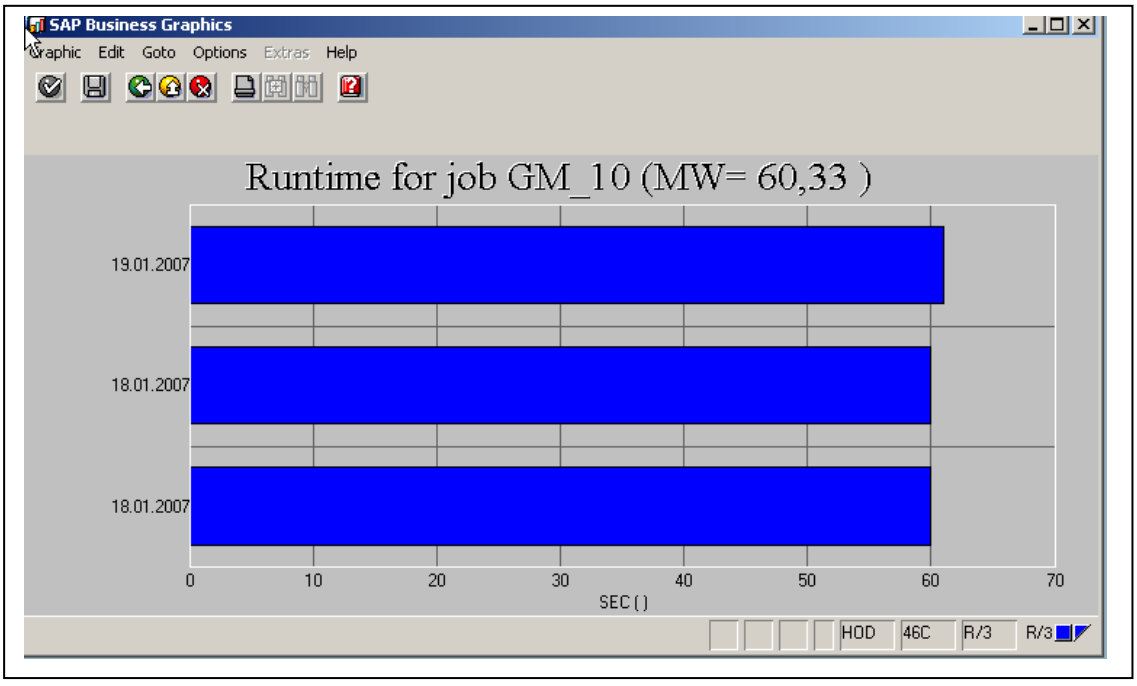
In this report, a statistical evaluation about individual run times of a selected job is done; output



can occur as list or as graphic display (the graphical display can not display more than 30 runs).

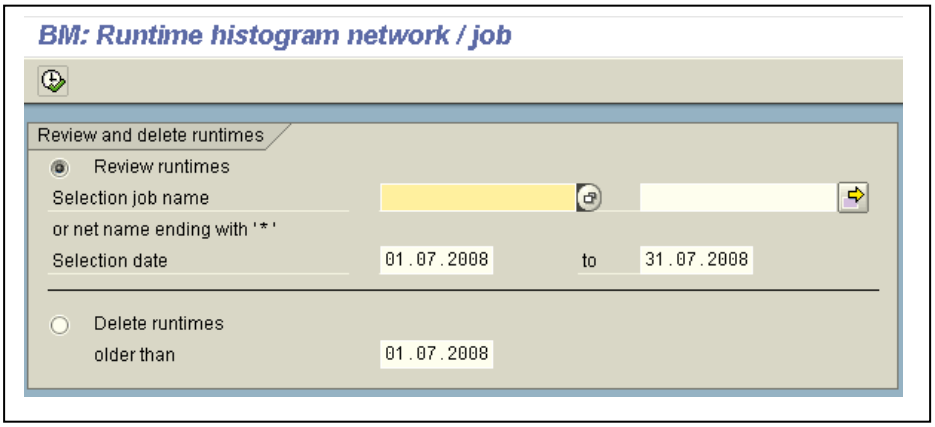


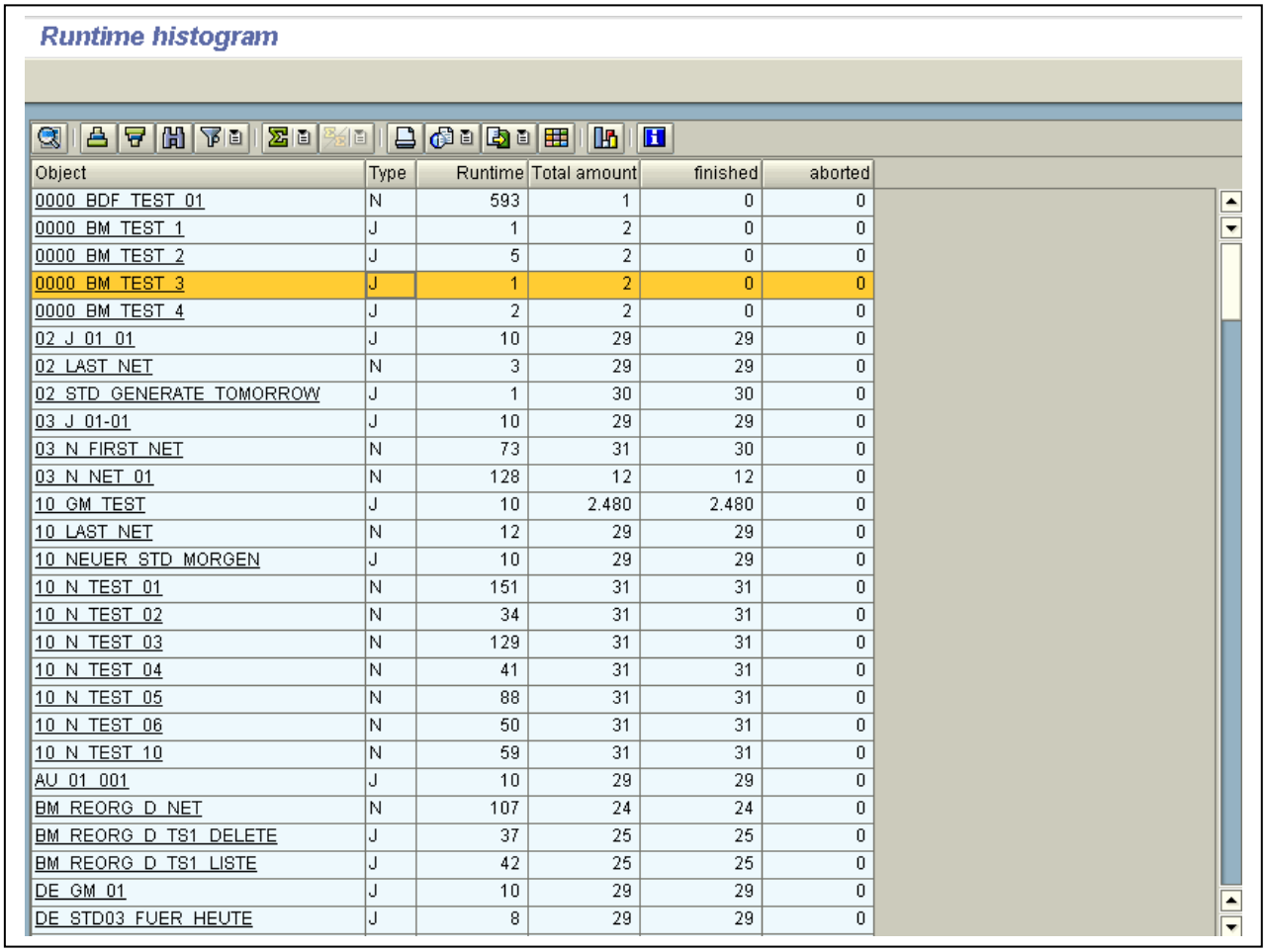
Start date	-time	End date	-time	Runtime in (s)	Status	Order
13.08.2008	01:00:18	13.08.2008	01:01:18	60	finished	STD012008081300001
12.08.2008	13:00:11	12.08.2008	13:01:11	60	finished	STD032008081200001
12.08.2008	01:00:20	12.08.2008	01:01:20	60	finished	STD012008081200001
11.08.2008	13:00:09	11.08.2008	13:01:09	60	finished	STD032008081100001



2.3.14.7. Run Time Histogram (Date)

In this report (/BTCMAN/M_LFG), a statistical evaluation about individual run times of a selected time period is done. In the output list, you can jump via double click in the detailed list (Run time histogram (Number of Runs)). The number of jobs determines the average run time in this case.





Object	Type	Runtime	Total amount	finished	aborted
0000_BDF_TEST_01	N	593	1	0	0
0000_BM_TEST_1	J	1	2	0	0
0000_BM_TEST_2	J	5	2	0	0
0000_BM_TEST_3	J	1	2	0	0
0000_BM_TEST_4	J	2	2	0	0
02_J_01_01	J	10	29	29	0
02_LAST_NET	N	3	29	29	0
02_STD_GENERATE_TOMORROW	J	1	30	30	0
03_J_01-01	J	10	29	29	0
03_N_FIRST_NET	N	73	31	30	0
03_N_NET_01	N	128	12	12	0
10_GM_TEST	J	10	2.480	2.480	0
10_LAST_NET	N	12	29	29	0
10_NEUER_STD_MORGEN	J	10	29	29	0
10_N_TEST_01	N	151	31	31	0
10_N_TEST_02	N	34	31	31	0
10_N_TEST_03	N	129	31	31	0
10_N_TEST_04	N	41	31	31	0
10_N_TEST_05	N	88	31	31	0
10_N_TEST_06	N	50	31	31	0
10_N_TEST_10	N	59	31	31	0
AU_01_001	J	10	29	29	0
BM_REORG_D_NET	N	107	24	24	0
BM_REORG_D_TS1_DELETE	J	37	25	25	0
BM_REORG_D_TS1_LISTE	J	42	25	25	0
DE_GM_01	J	10	29	29	0
DE_STD03_FUER_HEUTE	J	8	29	29	0

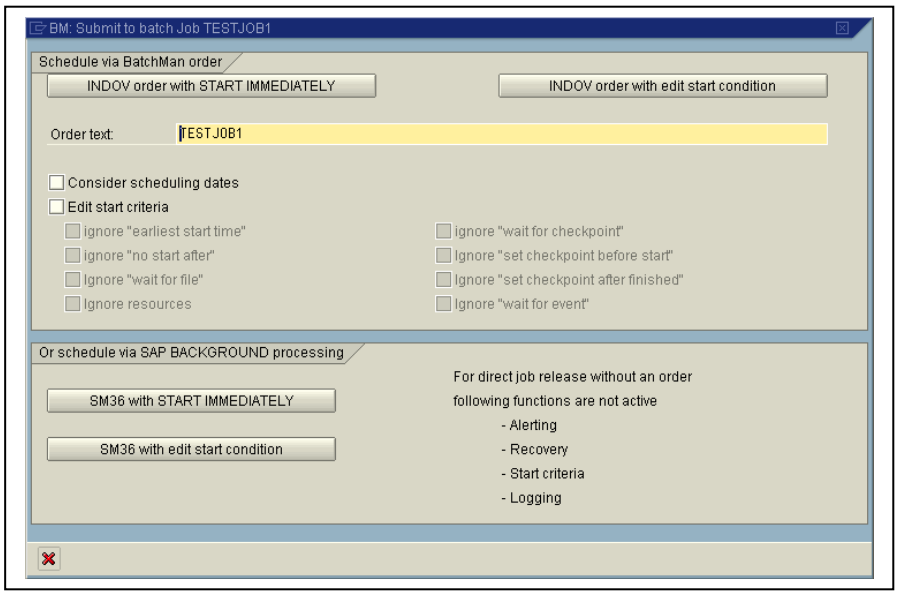
With this report you can also delete the run time histograms if they were not already deleted during archiving.

2.3.15. Scheduling of Ad hoc Job Processing

Via the button **‚Submit to Batch'** you have the possibility to schedule jobs quickly in the batch system. Here you have two options:

- The direct scheduling of jobs in the SAP- Batch-System via „Submit to Batch" on job-level with the alternatives **Scheduling** and **Release.** The scheduling option can be modified.
- Scheduling via orders. In this case an INDIV order is created specifically for this network. The text of the order can be changed. The scheduling option can also be modified.

Attention: The function ‚Submit to Batch' is only active in the **Display-Mode** of the job maintenance. After pushing ‚Submit to Batch', a Dynpro appears to request further selection.



Scheduling of jobs in the SAP-Batch-System can be done in two ways:

1. **Direct scheduling of jobs in the batch system via the function Submit to Batch with the two alternatives:**

The job can be released with **Start immediately o**r **Start time edit**. BatchMan functions like recovery, alerting etc. are not active in this case.

In the selection **Start time edit**, a Pop-Up appears, in which a start date must be entered. If you enter a valid start date, the processing (INDIV order or SAP job) is transferred to the batch system and receives the status *Released*.

2. **Scheduling of jobs via integration in operator orders (Type INDIV)**

In selection INDIV-order, the text of the operator order can be entered as well. The default is **ADHOC-JOB** and the **Job name**.

The operator order can be released via **Start immediately o**r **Start time edit**. A complete description of the operator orders follows in section 3.

If the “Consider Scheduling Dates” flag is activated, the scheduling dates of the object are taken into account.

For the scheduling, you may deactivate individual start criteria for this scheduling via the flag “*Edit start criteria”*.

2.4. Administration of job nets

A network consists of a sequence of network-type-linked jobs. „Network-type-linked“ means the mere sequential form, as well as the parallel or mixed form of both object links. The objects can be jobs or networks. The short name is NET.

The networks are managed the BatchMan library. The network maintenance can be called in two ways:

» Directly with the transaction J5HAN

» Via the function tree: *Master data / Library → Network maintenance*

The following options exist for the scheduling of networks in the batch system:

a) Scheduling of networks through integration in STD orders with the alternative „Automatic scheduling “ (Field „Automatic scheduling“ marked) in periods of at least one day (≥ 1Day)

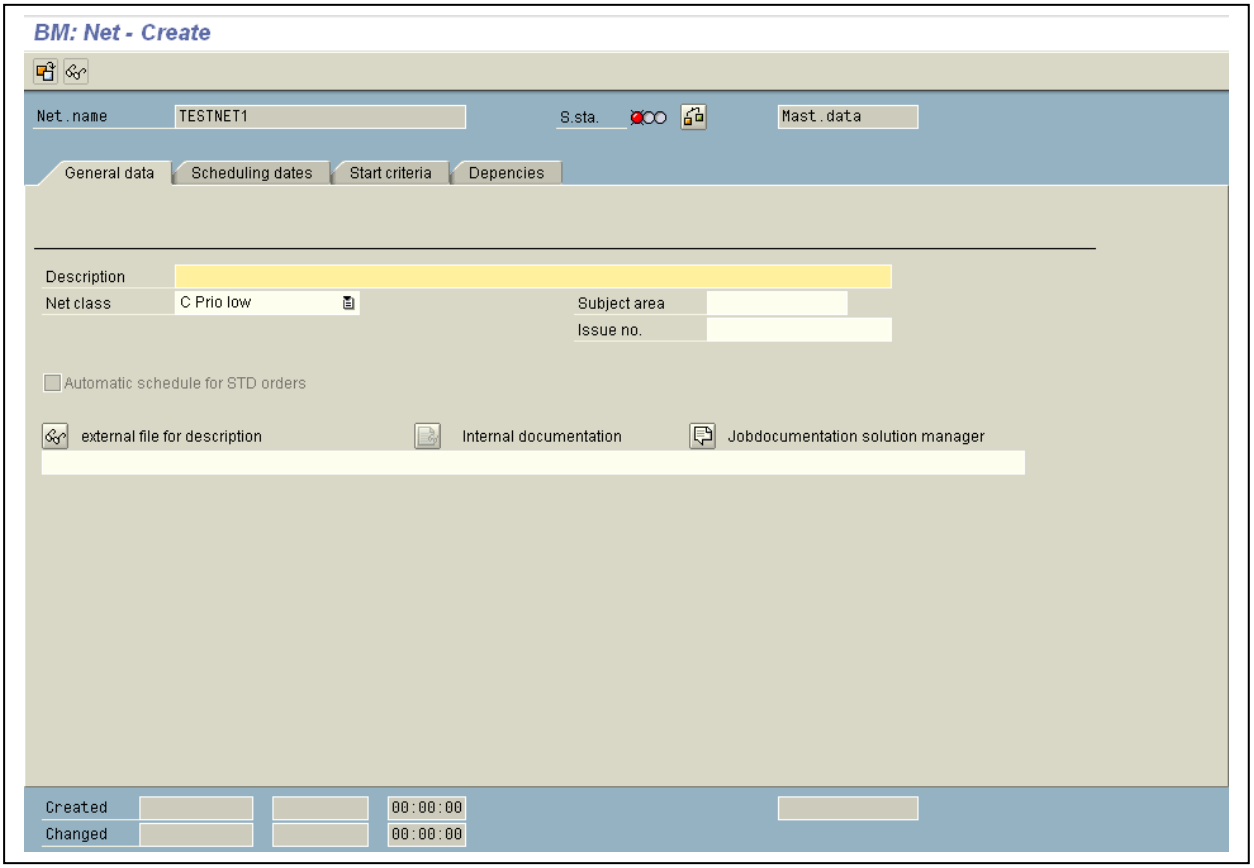
b) Scheduling of networks through integration in WDH orders in less than one day repetition (field ‘Automatic scheduling“ not marked) (< 1 day)

c) Manual scheduling of networks in INDIV orders (field „Automatic scheduling’ not marked)

2.4.1. Introduction Job networks – General Data

After entering the **network name**, the desired function can be selected:

	Create a new net
	Edit an existing net
	Display an existing net (also with the ENTER key)


BM: Net - Create
Net.name
TESTNET1
S.sta.
Mast.data
General data
Scheduling dates
Start criteria
Depencies
Description
Net class
C Prio low
Subject area
Issue no.
Automatic schedule for STD orders
external file for description
Internal documentation
Jobdocumentation solution manager
Created
Changed
00:00:00
00:00:00

Description: Here you can enter a long text for a job description.

Subject area: With the help of subject areas, you can group or classify jobs and network. In reporting, an evaluation/analysis can be executed on the basis of the subject areas.

Target instance: In this field you can specify a certain target instance (= application server), where all jobs of the network should run. Starting with SAP Release 6.10, you can also specify the target server group using this field.

If the SAP-System consists of several instances and no target system is indicated, then the scheduler decides on which host the job should run. All batch-capable target instances are displayed through a match code function.

ISSUE Number: The ISSUE number is a blank/empty field. Here sort criteria of other software can be deposited, e.g. from HelpDesk Systems.

‚**Automatic scheduling for STD orders':** The network for automatic scheduling of STD orders is activated via this flag. When an operator order of the type STD is created, only those networks which have the field 'Automatic scheduling' marked are automatically suggested for scheduling (see also order administration).

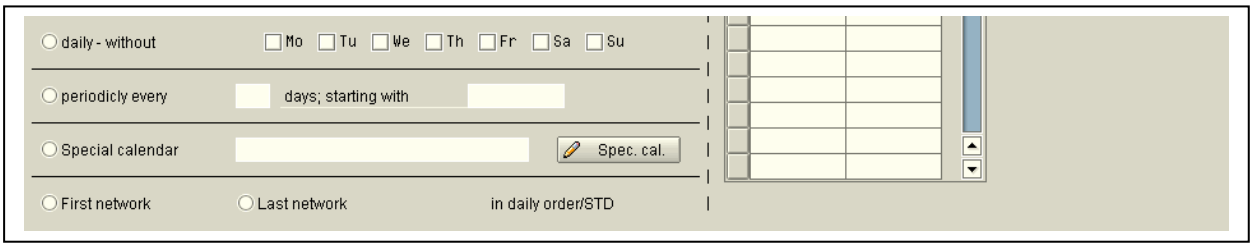
‚**External file for network description':** An external file for object documentation can be deposited in this field (see also menu function: Goto → Object documentation).

2.4.2. Scheduling Date of a Job Net

After you save for new networks, you can maintain using the "**Scheduling dates"** tab page. The scheduling date of a network results from the combination of its reference date and its periodicity.

For the description of the functionality **„Scheduling date**" see the respective section of this user manual. In addition, there are two flags in the scheduling dates for networks:

- Net is **first network** in the corresponding standard order or
- Net is **last Network** in the corresponding standard order



Networks that are scheduled with this flag are mainly there to establish or query certain system states before or after the day processing.

- Reset checkpoints
- Set checkpoints to certain values
- Set TVARV variables for the day processing
- Query whether the previous day order was ended

2.4.3. Start criteria for a net

The start criteria of a network are almost identical to the start criteria of jobs. For a description of the functionality „Start Criteria“, please refer to Section ‚Start Criteria of a Job‘.

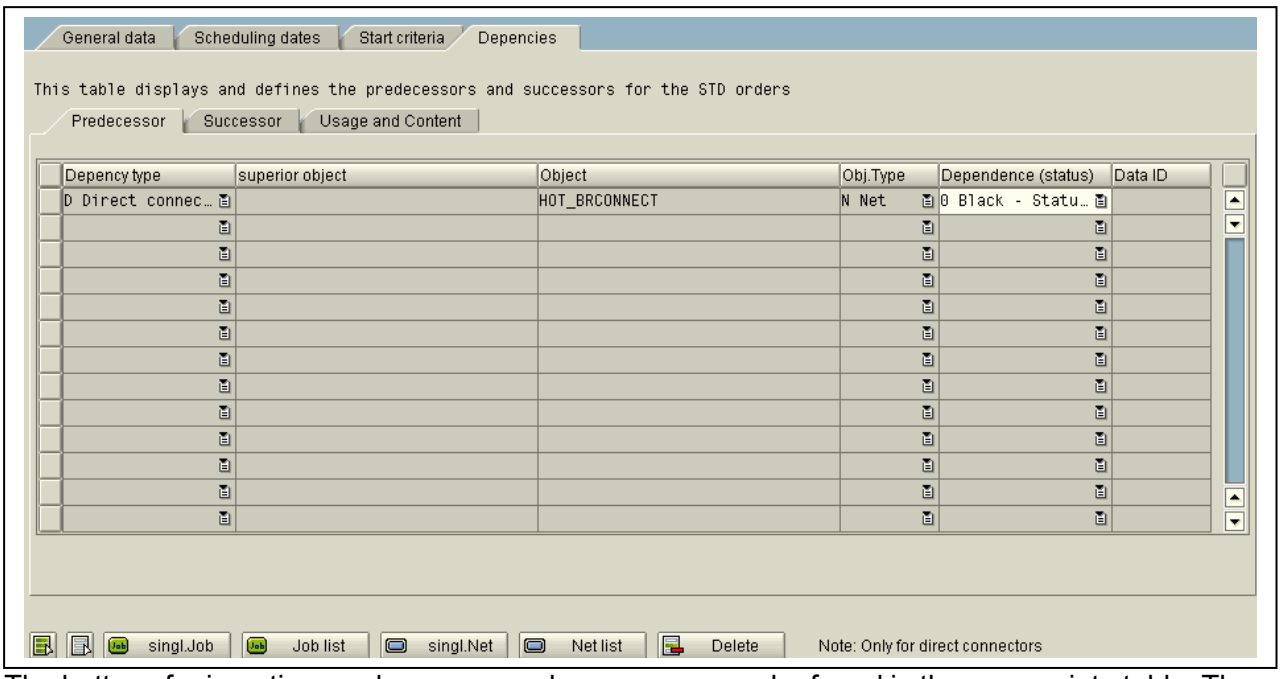
Exceptions are resources. In networks, the resource is occupied when the network is started - and only at network end the resource is released again. In the case that jobs are aborted in the network which have status-dependent successor jobs, the resource remains blocked. Should the resource be released nevertheless, it could be executed, for example, as recovery job with the report /BTCMAN/M_RE1.

2.4.4. Checkpoints

The checkpoint maintenance of a network is identical to the job maintenance. For the description of the functionality „Checkpoint“ see section ‘Start Criteria of a Job’.

2.4.5. Predecessor/ Successor

This file card is used for the maintenance or display of the predecessor- and successor objects. The objects can be individual jobs or networks.
Objects, that have the same scheduling dates like the current network, are linked with the current job in the order during automatic scheduling. This function only makes sense if the current network is scheduled as network on the highest level (flag ‚autom. scheduling‘ is active). This function can likewise only be used in STD orders.
Here, there is only one exception: When scheduling INDIV orders using “OpenWorld” interface, the data of this table is also processed.
We strongly recommend to read the appropriate documentation („Dependencies within Orders and Networks“ and „Individual Order Types“) before this function is used.
If the network is already linked with higher ranking networks, then predecessors and successor objects can only be deleted analog/according to the description of predecessors / succcessors of jobs (see respective secion of this handbook).
Jobs or networks, which are included in networks as objects, should not be entered analog as predecessor or successor.



The buttons for inserting predecessors and successors can be found in the appropriate table. The status of a link can be determined through the dependency.

Black	status-independent; continuing
Green	status-dependent; continuing if successful
Red	status-dependent; continuing if not successful
Yellow	Reunion after branching
Blue	Continue, ignore predecessors

An object may never be predecessor and successor. This is a **loop concatenation** and must strictly avoided.

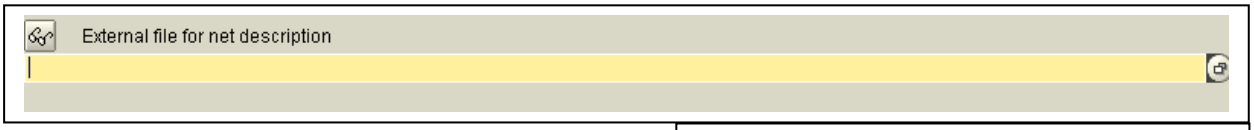
2.4.6. Documentation of a Net

With this function you can deposit documentations in a network. They allow a detailed description for the network.

This description is needed for operating procedures and preparation for order scheduling, as well as handling/ removal of malfunctions.

2.4.6.1. GOTO → External Object Documentation

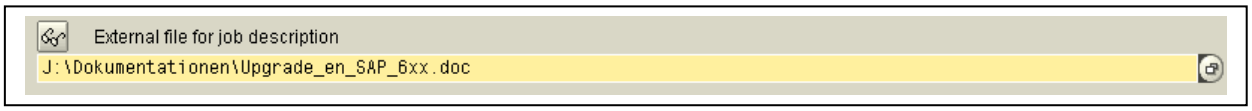
Using the external object documentation, you may store links to external documents. For this, call the search help (or F4 key).



An entry window is opened in which the documentation can be selected. After pushing the button ‚Open', the link is saved in the network definition

Which directory is used as search directory is specified in the setup with the parameter Standard directory for object documentation folders'.

After the selection, the path for the external document can be seen in the appropriate field:



If you now choose the menu path *Goto → ext. Objekt documentation,* then the system opens the object specified here. The related links must be known to the SAP system. By default, Word and Excel files are already known.

2.4.6.2. Goto → Object Documentation

An internal documentation also exists (in addition to the external documentation). In this case an editor window is called, in which the documentation is entered (see 'Goto Object documentation' in jobs).

It is possible to specify the texts which are displayed in the editor when the object documentation is called for the first time. These texts - templates can be maintained language-dependent:

Function tree BatchMan → Customizing → Extended Customizing → Text Templates

2.4.6.3. Job documentation Solution Manager

As of BatchMan 4.3, it is officially possible to store documentation of the jobs and networks in a solution manager system.

You have to maintain the related settings in the setup.

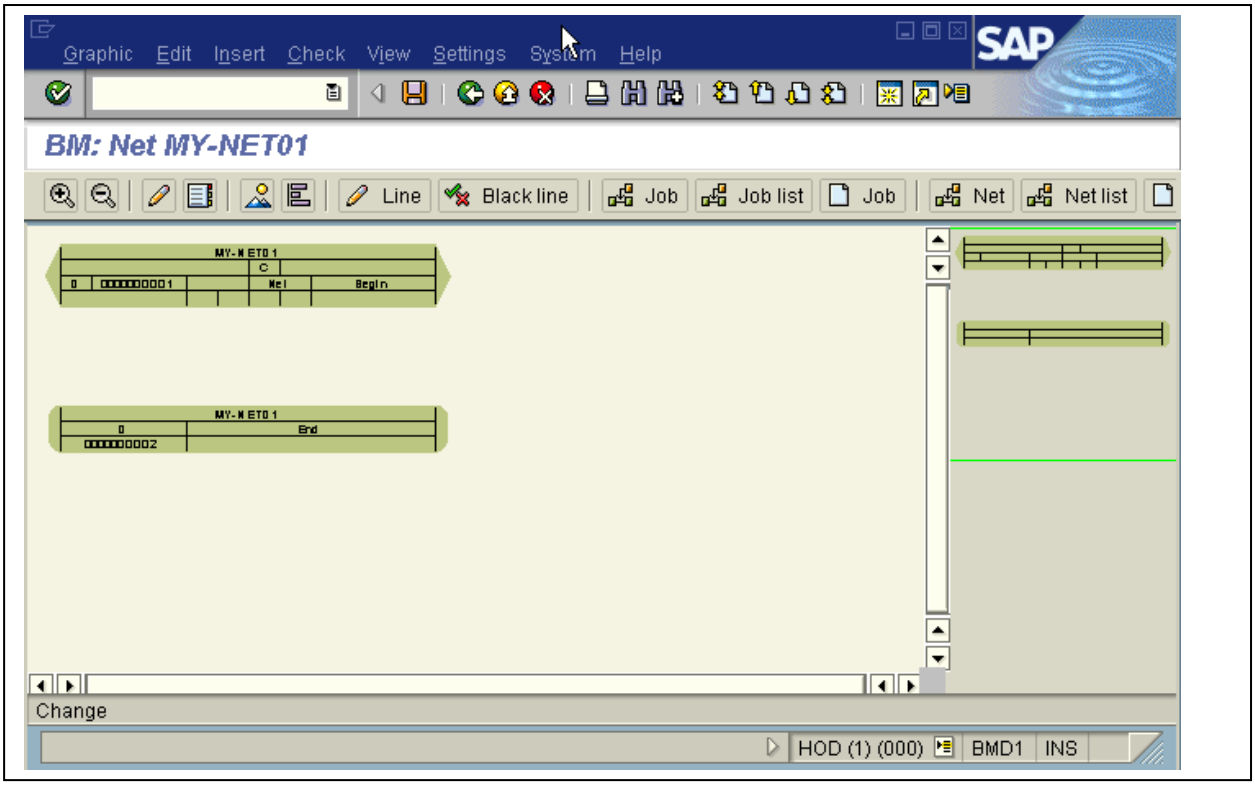
2.4.7. Net Plan Graphics of a Network

In order to maintain the positions (contents) of a network, the graphics function is called with the button ‚**Graphics'** or via the menu *Goto → Positions Graphics (F9)*.

In the network plan graphics of a network, the to-be used networks and jobs (objects) are assembled. In this graphics, objects are inserted, moved, removed in a network and the dependencies between these objects are maintained. The network can be shown in full screen view or partial screen view.

The objects within the network are displayed as objects in the graphics with their node numbers and therefore also called nodes. The connections between the nodes are called lines.

When you call a network for the first time, two nodes are already created. These are the beginning/start node (node number 1) and the end node (node number 2). Between these two nodes, all other nodes are positioned and connected.



Note: The graphic has a separate note or status bar. This can be seen at the very bottom.

On the right is the navigational area. The green frame represents the border of the display area, which is displayed in more detail on the left. If the mouse pointer is in the navigational area, it changes depending on the position. With this, it is possible to change the size and position of the navigational window; and thus also the left area.

Some buttons for the display size of the display area

	Complete view of the display area
	Maximize the nodes in the display area; the display area becomes smaller
	Minimize the nodes in the display area; the display area becomes larger

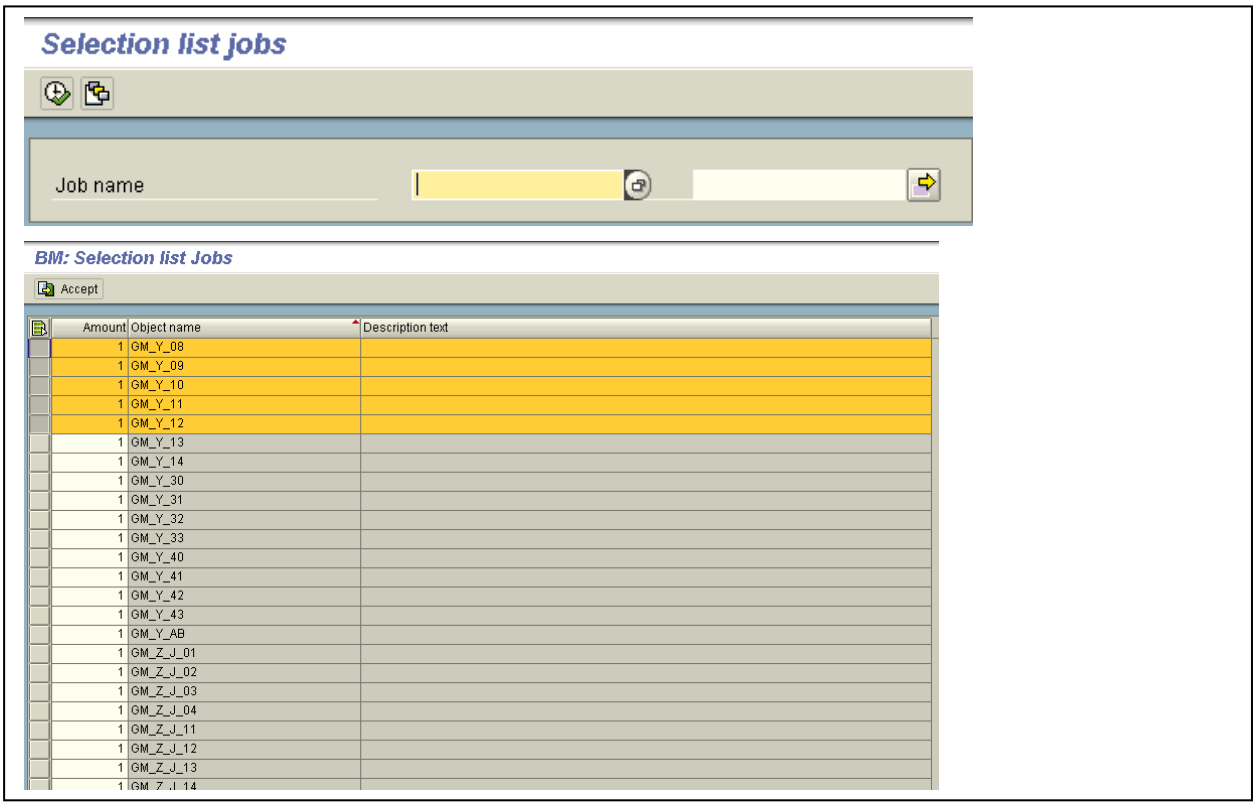
2.4.7.1. Insert Nodes

Inserting nodes is possible via the following buttons:

	Inserting an individual job. Depending on the number of the jobs defined in the master data and the personal set ups in the help settings, a selection screen is called. The jobs are selected according to the selection during executing. Due to the restriction to a certain number of objects (200-500), it is possible that not

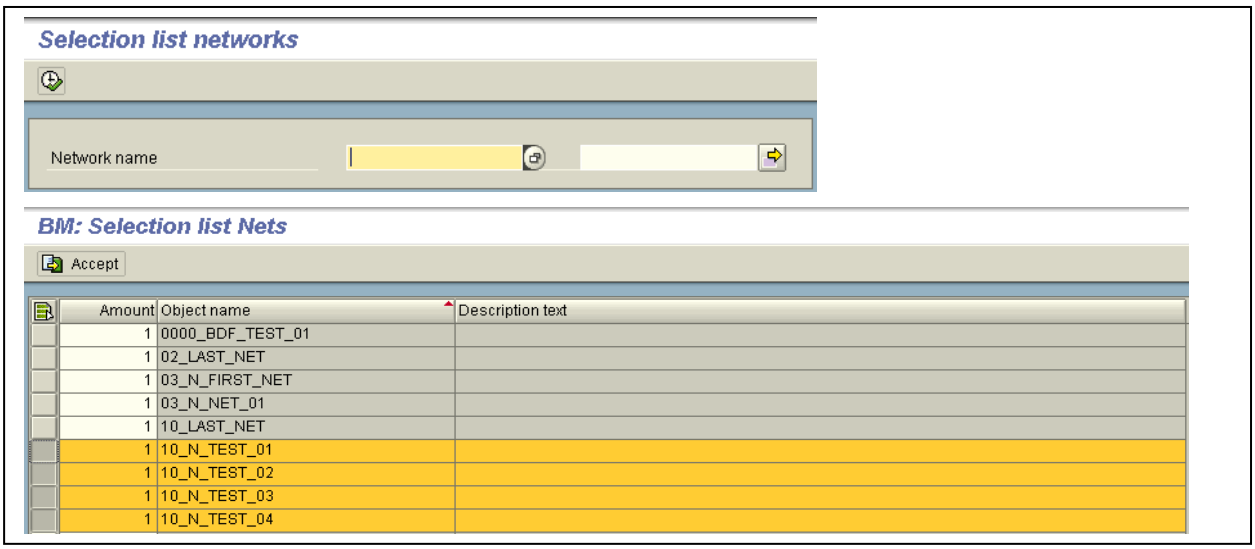
	all jobs are displayed.If fewer jobs are defined than indicated in the restriction, then all available jobs are shown at once.
Job list	Calling a selection screen to select jobs. After executing, a list of jobs is displayed, in which all jobs must be marked that should be entered in the graphics.
Net	Insert an individual network. Depending on the number of networks defined in the master data and the personal settings for the input help, only one selection screen is called at first. Upon execution, the networks are chosen by the selection. Due to the restriction to a certain number of objects (200-500), not all networks may be displayed. If fewer networks are defined than are specified in the restriction, the system displays at once all the networks available.
Net list	Calling a selection screen to select networks. After executing, a list of networks is displayed, in which all networks must be marked that should be entered in the graphics.

Job list



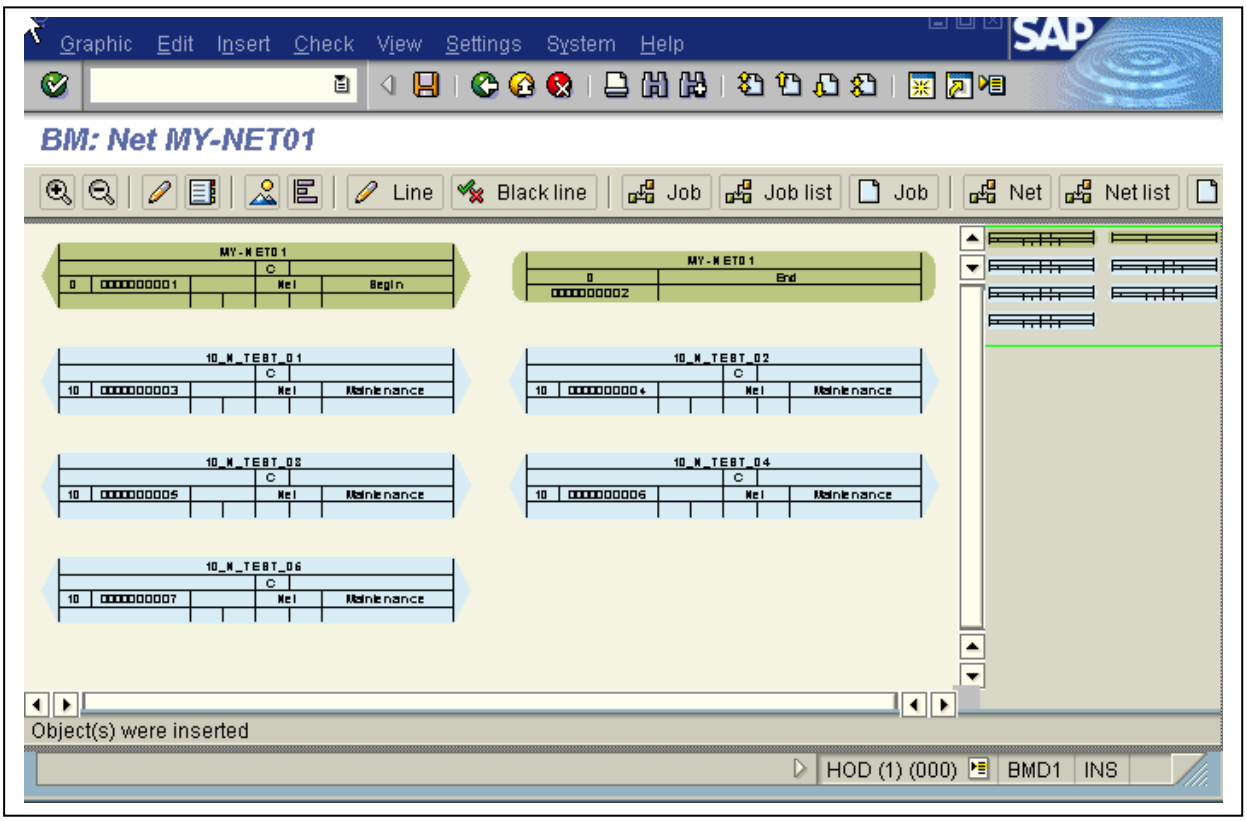
The jobs that should be entered in the graphics must be marked. By changing the numbers (count), a job can be entered several times. After pushing the **Accept** button, the selected jobs are inserted in the graphics. Status ‚Object(s) was (were) inserted' appears.

Net list



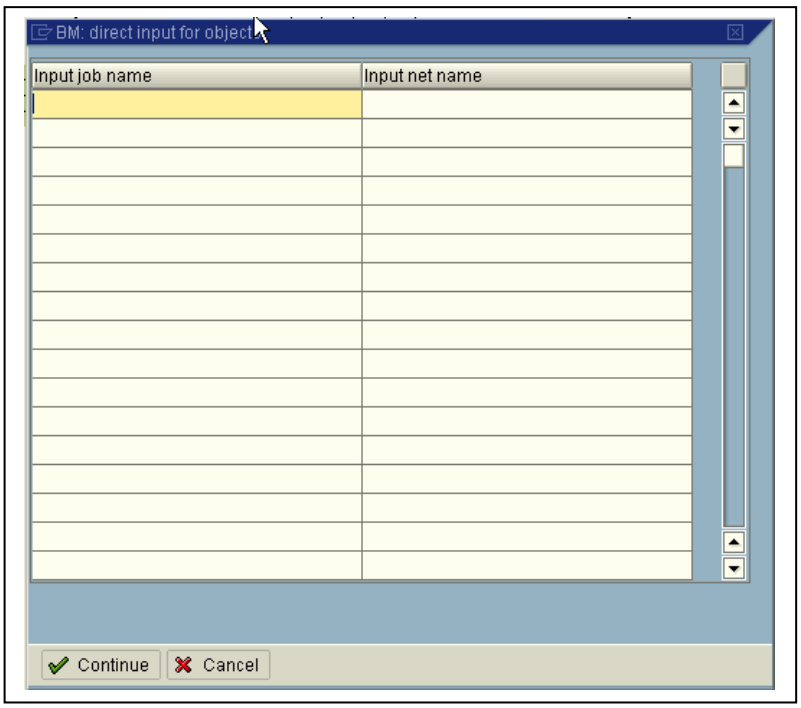
The networks that should be inserted in the graphics should be marked. By changing the numbers (count), a network can be entered several times.

After pushing the **Accept** button, the selected networks are inserted in the graphics. Status ‚Object(s) was (were) inserted' appears.



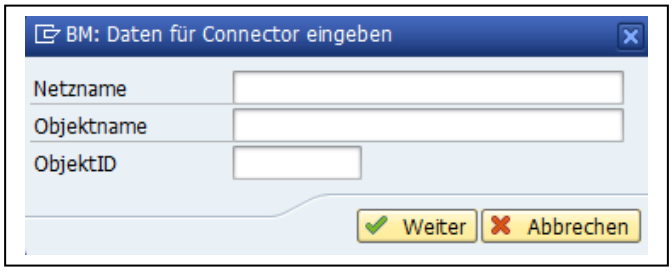
Inserting many nodes in one or two selections can restrict the display clarity. It is therefore recommended to connect and align the nodes in an intermediate step.

Using this pushbutton, you can call a sub-window, where you can enter the objects (jobs and networks) directly.



Conn.

With the help of connectors, predecessor and successor dependencies between objects within two different networks can be displayed graphically.



The connectors are created when the order is generated. If the connectors refer to networks in 2 different orders, these connectors are only taken into account when both orders have been generated.

2.4.7.2. Move Nodes

You may move nodes in the display area by holding down the mouse button.

2.4.7.3. Create nodes

Job	Branch to the job maintenance; the job may be created.
Net	Branch to the network maintenance; the network may be created.

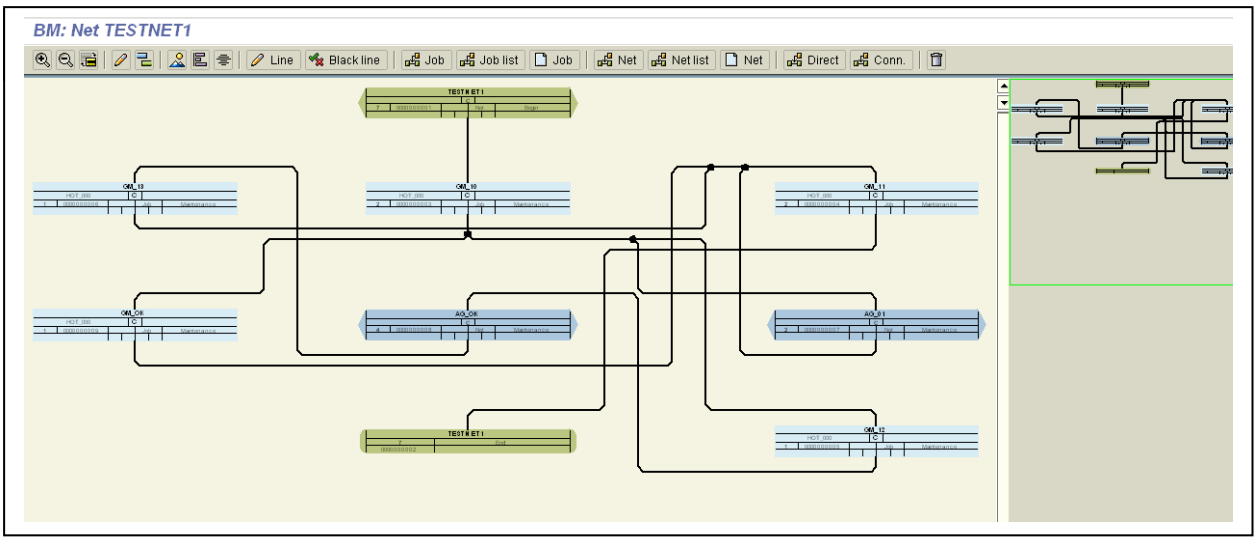
Afterwards the nodes can be inserted in the graphics via the inserting functions.

2.4.7.4. Connect nodes

The nodes can be connected with each other via the button ‚Line'.

In is drawn (here: black), is specified through the appropriate 

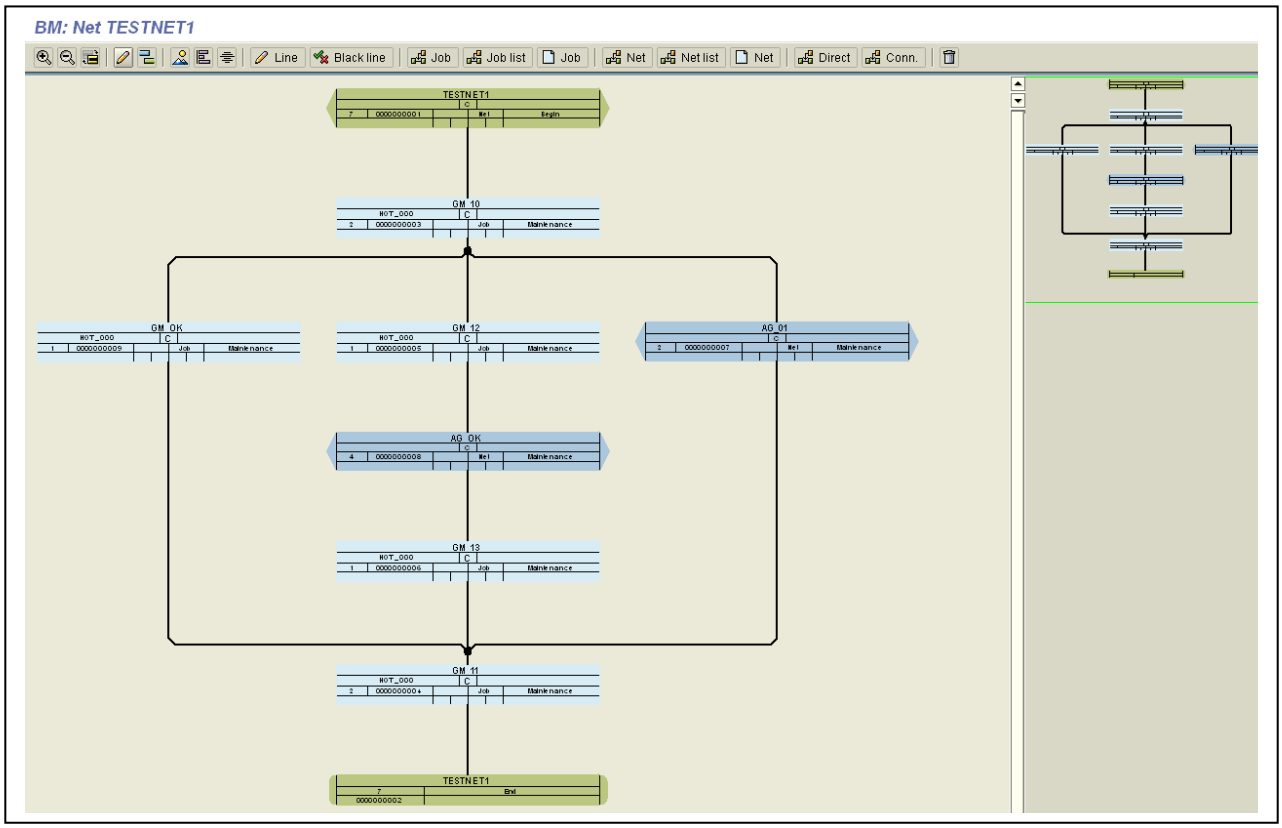
button (see also „Line color")



In comprehensive links, it is difficult to recognize which node represents the predecessor and which node represent the successor. The function „Align“ improves the graphics’ clarity.

2.4.7.5. Aligning the Graphics

Via the button ‚Align/ Adjust’ the graphics is adjusted so that the relationship between the nodes becomes clear.

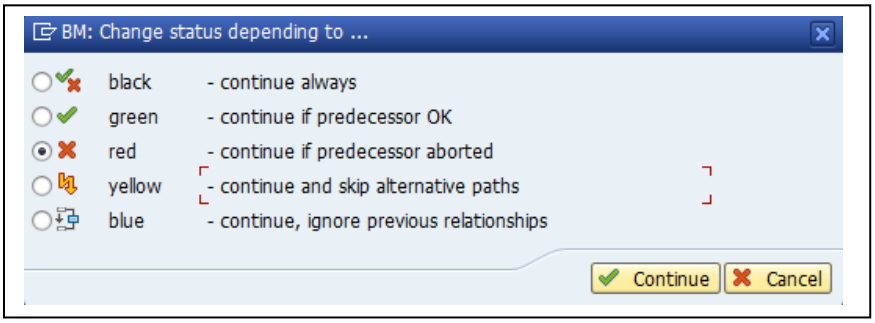


2.4.7.6. Line colors – Status Dependency

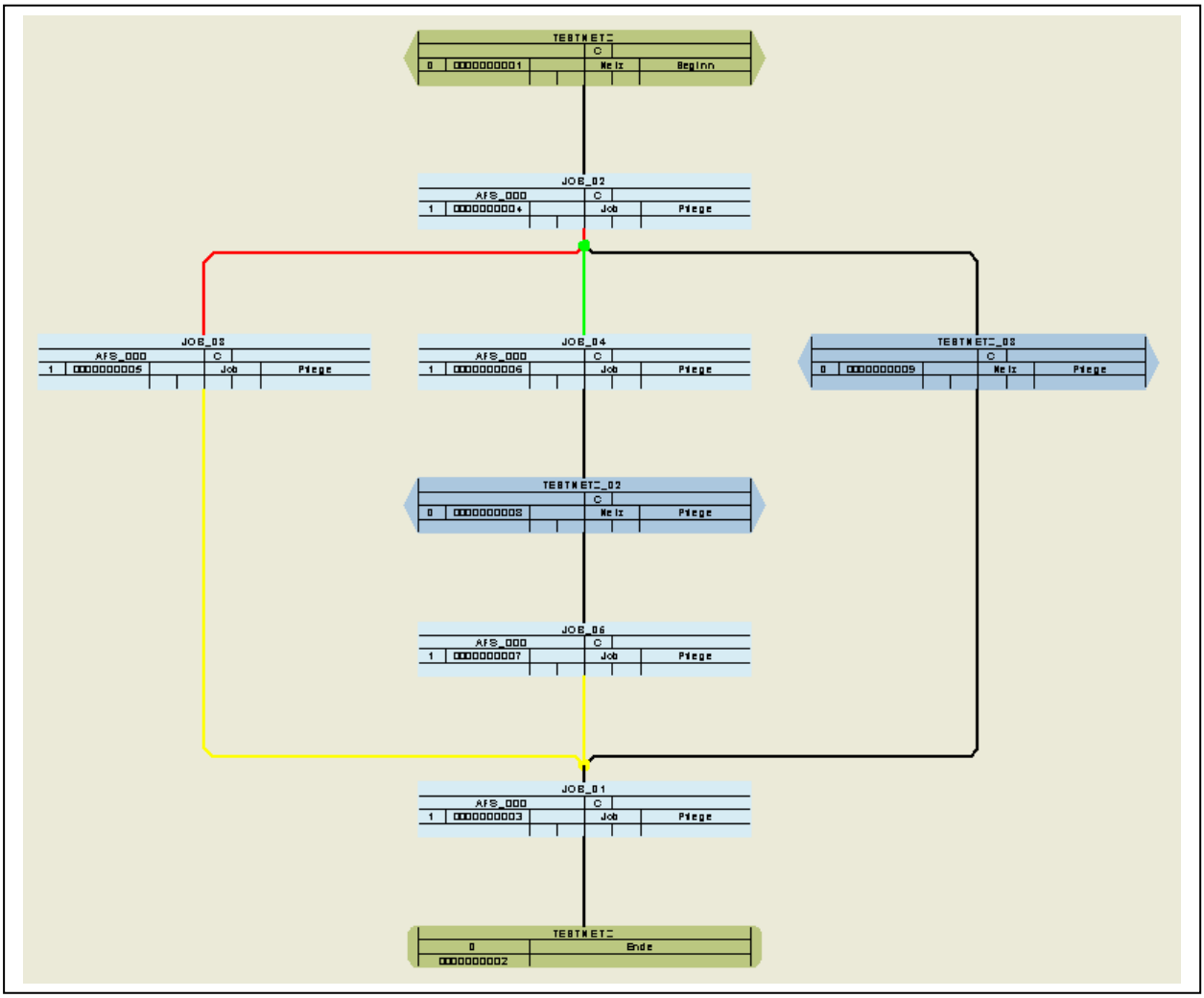
The color of the line influences the processing flow. The color can be changed before or during the drawing of the lines via the button ‚Line color'.

Black line	The connection is status-independent; the successor node ALWAYS starts, independent of the predecessor node status.
Green line	The connection is status-dependent; the successor node starts if the status of the predecessor nodes is SUCCESSFUL.
Red line	The connection is status-dependent; the successor node starts when the status of the predecessor node is ABORTED. The red line is an alternative to the green line.
Yellow line	This connection is the junction of an alternative path of green and red line.
Blue line	A blue line starts the successor regardless of whether its other predecessors have already been finished (race condition). If a blue line is parallel to a colored line, the colored line has priority – the alternative route that is not used is skipped.

The line color can also be changed by double-clicking on the line. In a PopUp window, the line color can be changed as desired.



In the following example, the line colors and the corresponding dependencies are explained again:



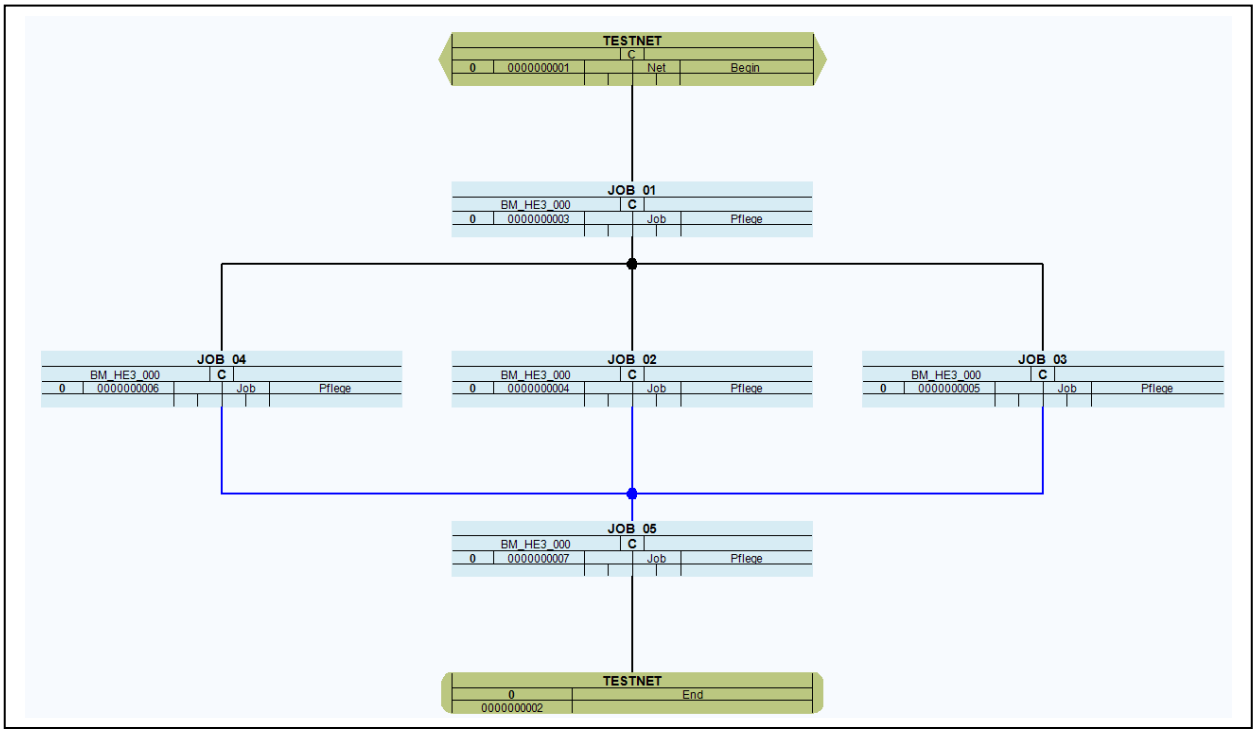
After the begin/start node, JOB_02 is started. Depending on the status of the execution of this job, further processing is continued.

JOB_02 has run successfully, Status ‚finished':

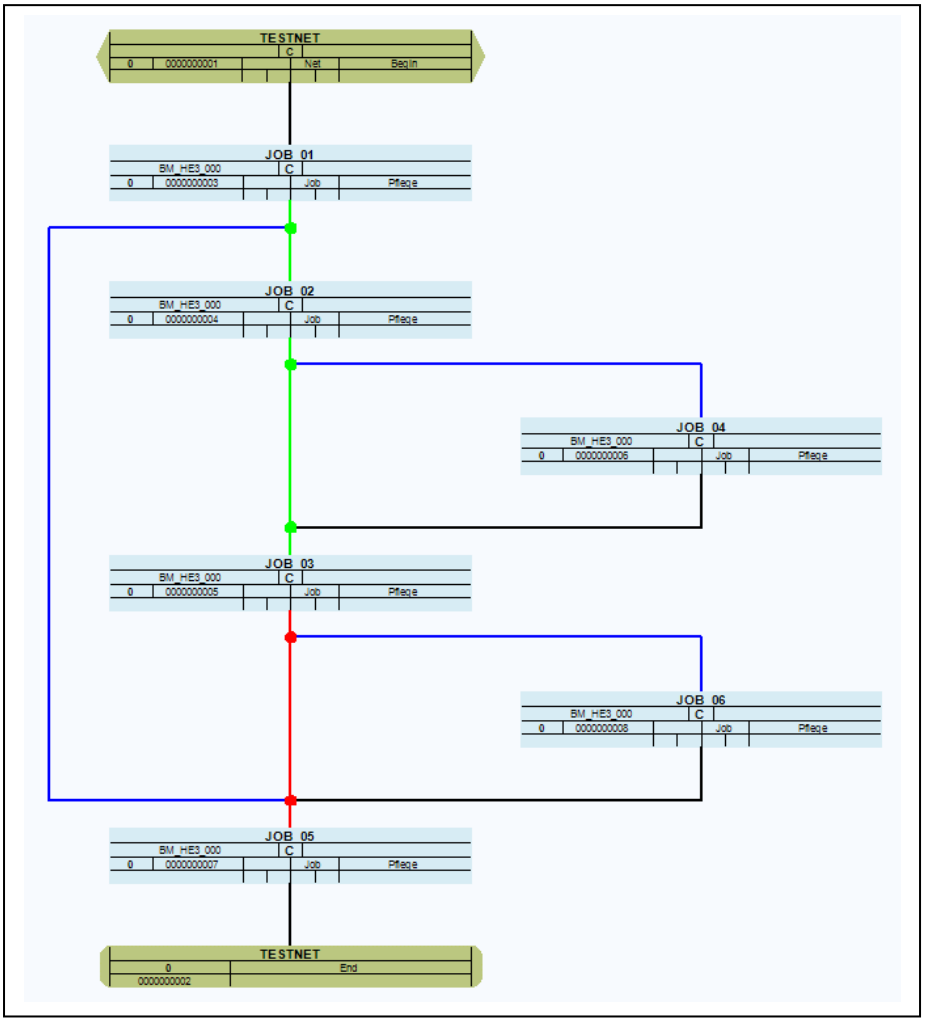
- TESTNETZ_03 starts running, since status-independent
- JOB_04 starts running, afterwards TESTNETZ_02 and then JOB_05
- JOB_05 finds the yellow line for its successor JOB_01 and looks for all parallel yellow lines; all nodes before the lines are skipped.
- When TESTNETZ_02 is also finished, JOB_01 will start running

JOB_02 did not run successfully, status ‚aborted':

- TESTNETZ_03 starts running, since status-independent
- JOB_03 starts running
- JOB_03 finds the yellow line for its successor JOB_01 and looks for all parallel yellow lines; all nodes ahead of the lines (JOB_05, TESTNETZ_02 and JOB_04) are skipped.
- If TESTNETZ_02 is also finished, then JOB_01 will start running.



After the begin/start node, JOB_01 is started. JOB_03, JOB_02 and JOB_04 are then started in parallel. As soon as one of these 3 jobs is finished, JOB_05 is started immediately - due to the blue lines, not all predecessors of JOB_05 must be finished (race condition).



After the Begin node, JOB_01 is started. JOB_01 has a blue line parallel to a colored line – the colored line has priority, the alternative route is skipped:

- If JOB_01 is aborted, the blue line is followed and JOB_02, JOB_04, JOB_03, JOB_06 are skipped as the alternative route, processing continues with JOB_05.
- If JOB_01 has run successfully, the green line is followed (with priority as a colored line). The blue line has a direct intersection with the path started by the green line (before JOB_05) with no intermediate jobs to be skipped, processing continues with JOB_05.

JOB_02 has a blue line parallel to a colored line – the colored line has priority, the alternative route is skipped:

- If JOB_02 is aborted, the blue line is followed. The green line has a direct intersection with the path started by the blue line (before JOB_03) with no intermediate jobs to be skipped, processing continues with JOB_03
- If JOB_02 has run successfully, the green line is followed (with priority as a colored line) and JOB_04 is skipped as an alternative route, processing continues with JOB_03.

JOB_03 has a blue line parallel to a colored line – the colored line has priority, the alternative route is skipped:

- If JOB_03 is aborted, the red line is followed (with priority as a colored line) and JOB_06 is skipped as an alternative route, processing continues with JOB_05.

If JOB_03 has run successfully, the blue line is followed. The red line has a direct intersection with the path started by the blue line (before JOB_05) with no intermediate jobs to be skipped, processing continues with JOB_05.

2.4.7.7. Delete Node

Mark nodes and lines and delete with the button **Trashcan**. You can mark several objects simultaneously with pushed ‚*Shift'* – key.

2.4.7.8. Edit Node

Mark desired node (black mark on the corners) and click on the button ‚**Edit'**. The object (Network or job) can be edited.

2.4.7.9. Display Node

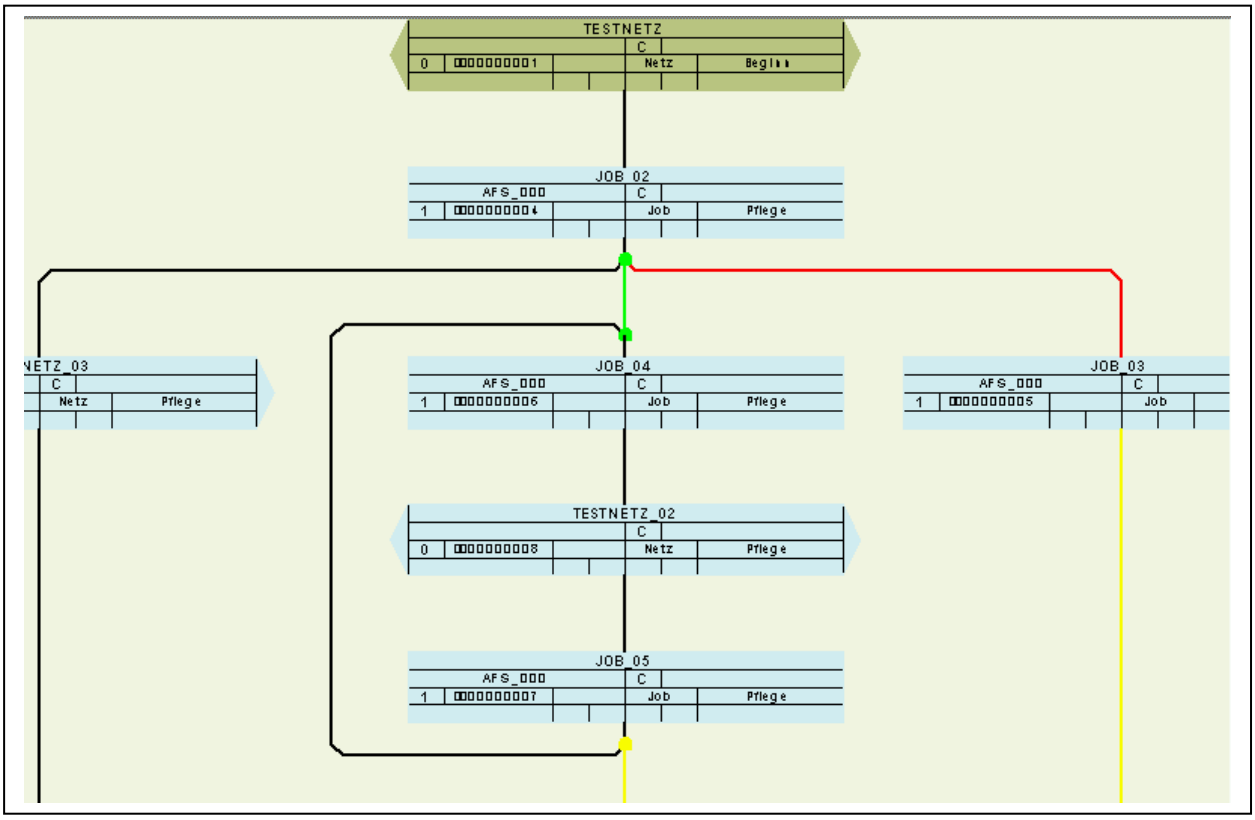
Mark the desired node (black mark on the corners) and click on the button ‚**Display'.** The object (network or job) will be displayed.

2.4.7.10. Print Graphics

Via *Menu → Graphics → Print* you can print the graphics. Several options for printer setup are available.

2.4.7.11. Loop concatenation

Ring chaining are connections that move in circles.



Here loop concatenation is used between the nodes JOB_04, TESTNETZ_02 and JOB_05. When calling the function in *Menu → Check → Test loop concatenation* these three nodes are displayed as error. loop concatenation must be corrected by the user himself.

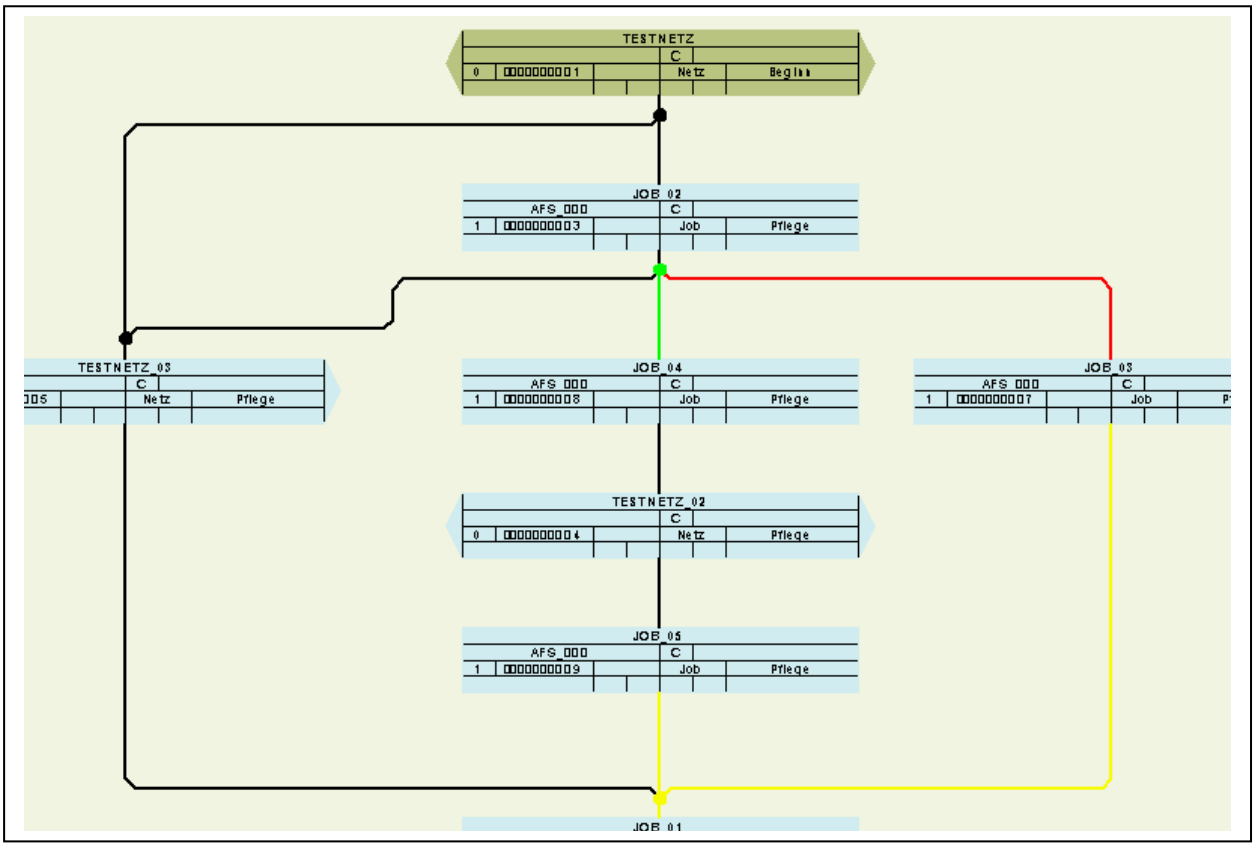
This function is automatically called during saving.

2.4.7.12. Nodes without Connections

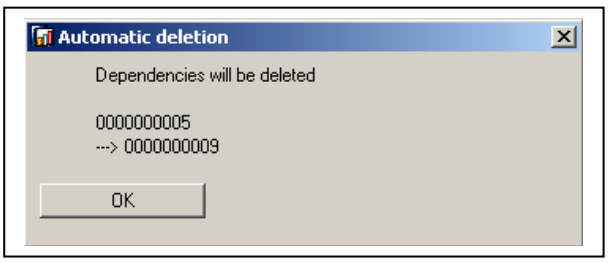
Nodes without connections to a predecessor and / or without connections to a successor are checked via *Menu → Check → Test Nodes without connection*. PopUp windows appear with suggestions for node connections.

Nodes without predecessor connection are linked to the beginning/start node. Nodes without successor connection are linked with the end node. This function is automatically called during saving.

While the node without connection is tested, an additional check is done if redundant lines exist. These will be deleted.



The connection between the beginning/start node and node TESTNETZ_03 is redundant because the node is always only started when JOB_02 was executed. The path runs always via the node JOB_02.

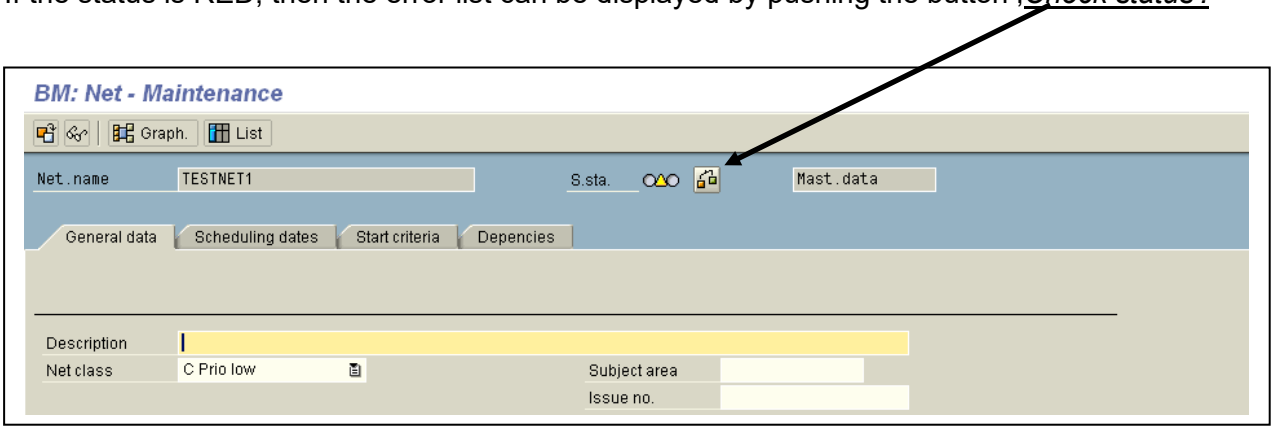


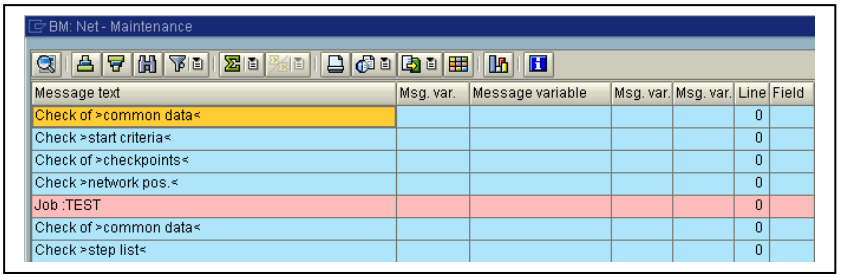
2.4.8. Additional Functions for Network Maintenance

2.4.8.1. Checking a Network

The entered values are always checked against the SAP system every time you save. If all entries are correct (no errors), the status will be GREEN. If errors occur, the status will be RED.

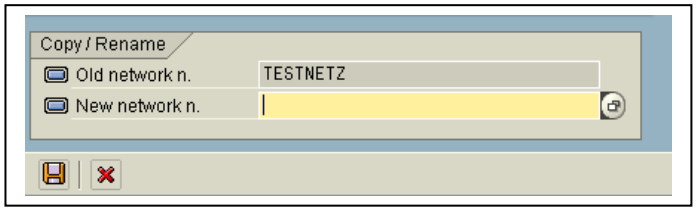
If the status is RED, then the error list can be displayed by pushing the button *‚Check status'.*





2.4.8.2. Copy

Call the function to copy the net. The system opens a window to query the new net name.



2.4.8.3. Rename

Call the function to rename the net. The system opens a window to query the new net name (same as copying).

2.4.8.4. Delete

Call the function to delete the net. This can only be deleted if the net is no longer used in other networks. After a confirmation prompt, the net is deleted.

2.4.8.5. Other object

Call the init mode of the net maintenance. All fields are deleted, the net name is ready for input.

2.4.8.6. Create transport request

Creates a SAP transport request, which contains the BM net definition as table content.

2.4.8.7. Send net definition

Sends the net definitions via RFC to another BM master system that can be selected via selection.

2.4.8.8. Compare net

Compares the net definitions of different BM master systems via RFC.

2.4.9. Menu Functions

Under the menu **Network,** the already described functions *Other Object, Create*, *Change*, *Display*, *Delete* and *Save* can be found. Via the function ‚*Print* all nets can be printed. See also ‚Reporting about Networks' below.

Under **Edit** you can find the functions ‚C*opy* and ‚*Rename* (see description ‚Additional Functions for the Network Maintenance' above).

2.4.9.1. Edit → Transport Object

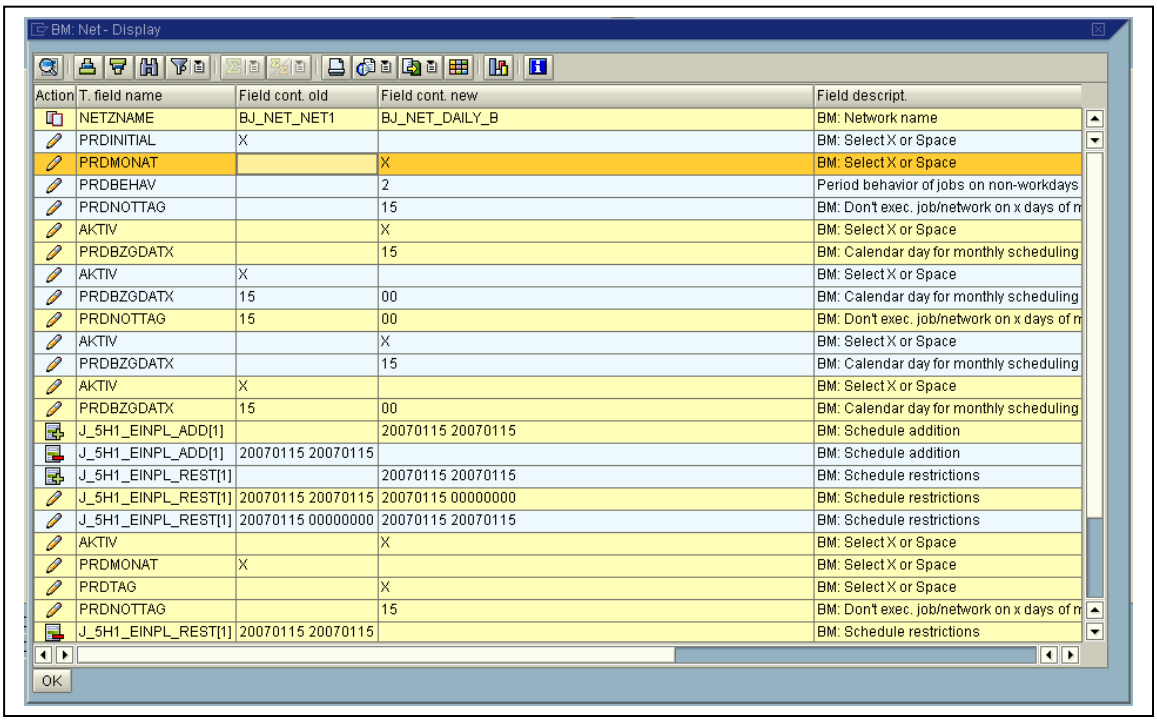
A network can be transported via this menu point or the button with the truck (in display mode). Since this function is executed via SAP transportation, you need the appropriated authorization (S_TRANSPRT / S_CTS_ADMI).

After a confirmation prompt, a window opens in which you select an already existing transport order. There is also the option to create a new order.

There is also the option to transport by writing (export) to a file and then uploading (import) into a second system. You can find more information in the "Transport" section.

2.4.9.2. Goto → Change History

Calling the change history for a network. All changed fields are shown.



Action	T. field name	Field cont. old	Field cont. new	Field descript.
	NETZNAME	BJ_NET_NET1	BJ_NET_DAILY_B	BM: Network name
	PRDINITIAL	X		BM: Select X or Space
	PRDMONAT		X	BM: Select X or Space
	PRDBEHAV		2	Period behavior of jobs on non-workdays
	PRDNOTTAG		15	BM: Don't exec. job/network on x days of n
	AKTIV		X	BM: Select X or Space
	PRDBZGDATX		15	BM: Calendar day for monthly scheduling
	AKTIV	X		BM: Select X or Space
	PRDBZGDATX	15	00	BM: Calendar day for monthly scheduling
	PRDNOTTAG	15	00	BM: Don't exec. job/network on x days of n
	AKTIV		X	BM: Select X or Space
	PRDBZGDATX		15	BM: Calendar day for monthly scheduling
	AKTIV	X		BM: Select X or Space
	PRDBZGDATX	15	00	BM: Calendar day for monthly scheduling
	J_5H1_EINPL_ADD[1]		20070115 20070115	BM: Schedule addition
	J_5H1_EINPL_ADD[1]	20070115 20070115		BM: Schedule addition
	J_5H1_EINPL_REST[1]		20070115 20070115	BM: Schedule restrictions
	J_5H1_EINPL_REST[1]	20070115 20070115	20070115 00000000	BM: Schedule restrictions
	J_5H1_EINPL_REST[1]	20070115 00000000	20070115 20070115	BM: Schedule restrictions
	AKTIV		X	BM: Select X or Space
	PRDMONAT	X		BM: Select X or Space
	PRDTAG		X	BM: Select X or Space
	PRDNOTTAG		15	BM: Don't exec. job/network on x days of n
	J_5H1_EINPL_REST[1]	20070115 20070115		BM: Schedule restrictions

Icons of the action correspond to the ones of the buttons.

2.4.9.3. Goto → Deleted Networks

With this function, you may display deleted networks in **BatchMan**. When a network is deleted, an entry is created indicating that the network was deleted.

If all other change documents for this networks are deleted depends on the Setup parameter ‚History will be kept if master data are deleted'. If this value is active, then these networks can still be displayed via *Reporting → Change history → Change history networks*.

2.4.9.4. Goto → Run Time Histogram

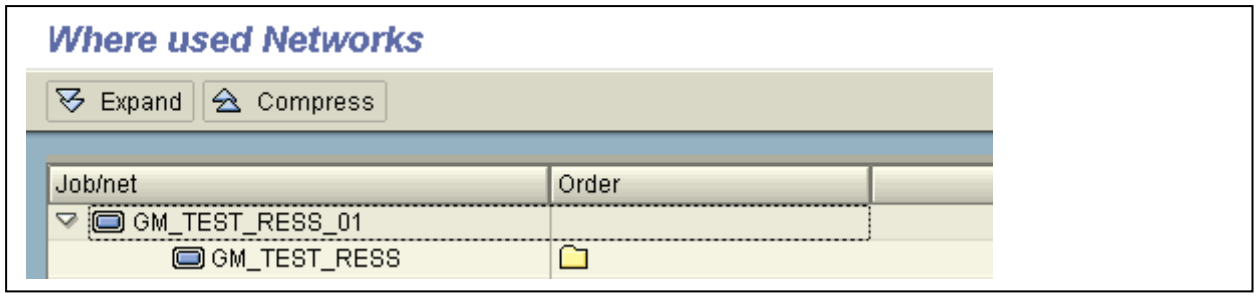
See ‘Run Time Histogram (runs)’ and also ‘Run Time Histogram (Date)’ under Reporting Networks.

2.4.9.5. Springen → Langzeit-Historie

See ‘Counter network scheduling’ under Reporting networks.

2.4.9.6. Environment → Usage only in Master Data

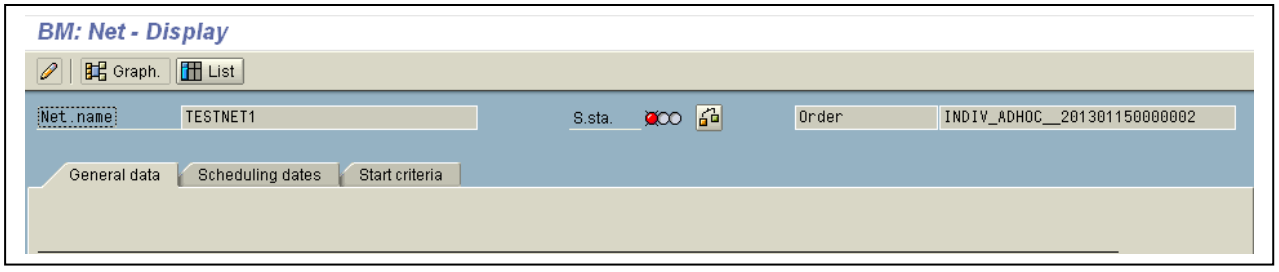
The function displays the usage of the current network in all upper-level networks (also across several levels).



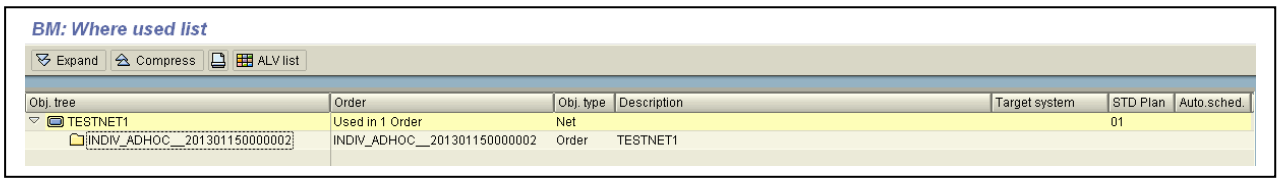
2.4.9.7. Environment → Usage in Master Data and Order Data

The function displays the usage of the current network in all upper-level networks (also across several levels) as well as the usage in the orders.

If you switch from the order maintenance to the network maintenance, the system displays the network copied for the order. This is evident when “Order” and the related order numbers are now displayed in the fields next to the status.



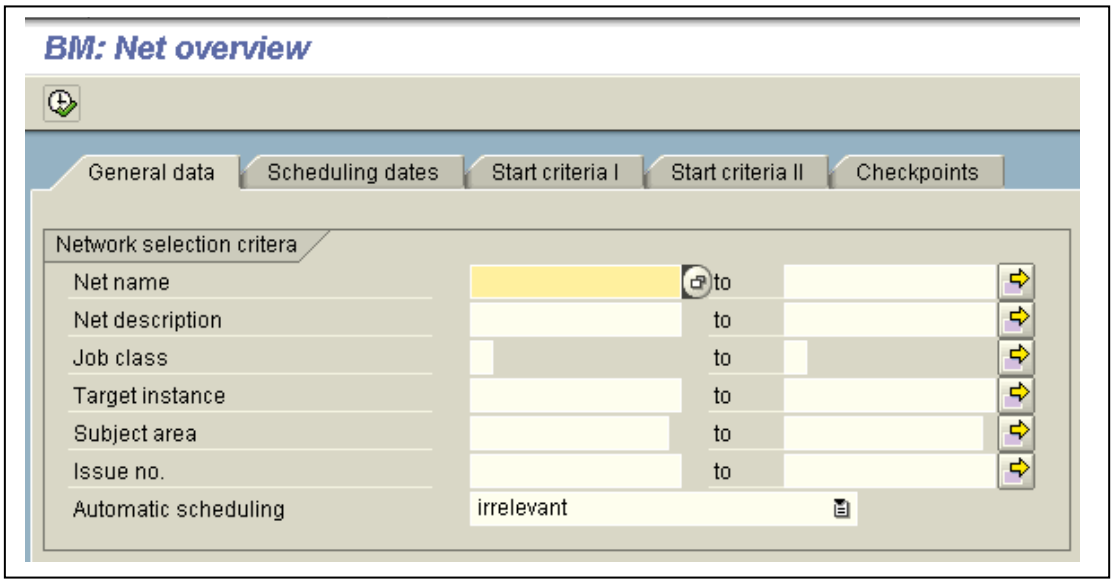
The Where used networks shows the orders.



2.4.10. Reporting about Networks

The following evaluation options exist about networks:

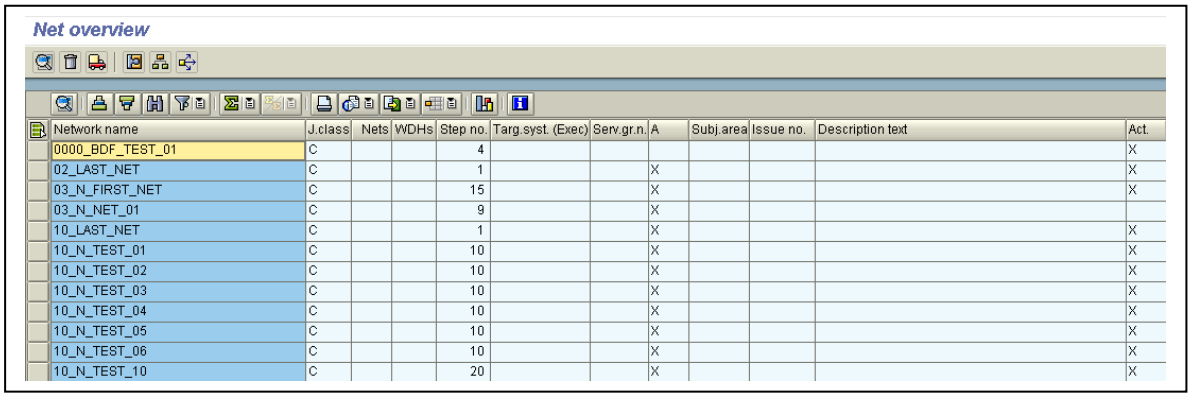
2.4.10.1. Net overview



In this report, a network overview may be compiled by selective selection criteria.

The result list is produced as ALV List and can be created as you wish. The lists can be changed and saved via the layout administration.

	Calling the job maintenance; analog to a double click on a line
	Detailed display of all job fields of a list (see Detailed list jobs); the jobs have to be marked
	Where-used document of a job; the jobs have to be marked
	Deleting the job (not only from the list). The usage of the job in networks and WDH orders is checked.
	Display of the marked jobs with their included steps (see Job List with Steps)
	Transport of the marked jobs via SAP transport order

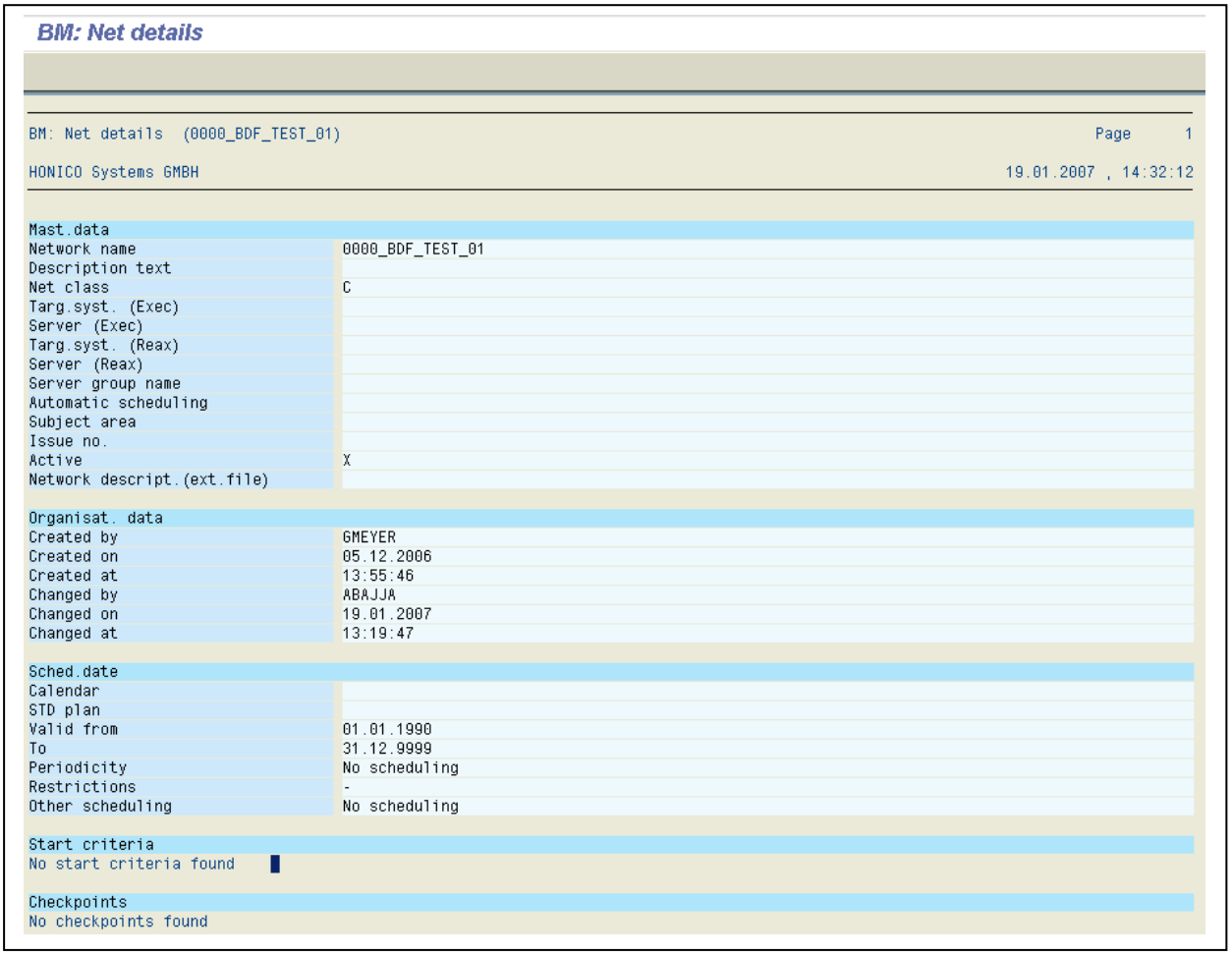


Network name	J.class	Nets	WDHs	Step no.	Targ.syst. (Exec)	Serv.gr.n.	A	Subj.area	Issue no.	Description text	Act.
0000_BDF_TEST_01	C			4							X
02_LAST_NET	C			1			X				X
03_N_FIRST_NET	C			15			X				X
03_N_NET_01	C			9			X				
10_LAST_NET	C			1			X				X
10_N_TEST_01	C			10			X				X
10_N_TEST_02	C			10			X				X
10_N_TEST_03	C			10			X				X
10_N_TEST_04	C			10			X				X
10_N_TEST_05	C			10			X				X
10_N_TEST_06	C			10			X				X
10_N_TEST_10	C			20			X				X

2.4.10.2. Net: Detailed List

This function can be used for a complete printout of one or all networks. All fields of a network are included.

Excerpt:



BM: Net details

BM: Net details (0000_BDF_TEST_01) Page 1

HONICO Systems GMBH 19.01.2007 , 14:32:12

Mast.data	
Network name	0000_BDF_TEST_01
Description text	
Net class	C
Targ.syst. (Exec)	
Server (Exec)	
Targ.syst. (Reax)	
Server (Reax)	
Server group name	
Automatic scheduling	
Subject area	
Issue no.	
Active	X
Network descript.(ext.file)	

Organisat. data	
Created by	GMEYER
Created on	05.12.2006
Created at	13:55:46
Changed by	ABAJJA
Changed on	19.01.2007
Changed at	13:19:47

Sched.date	
Calendar	
STD plan	
Valid from	01.01.1990
To	31.12.9999
Periodicity	No scheduling
Restrictions	-
Other scheduling	No scheduling

Start criteria

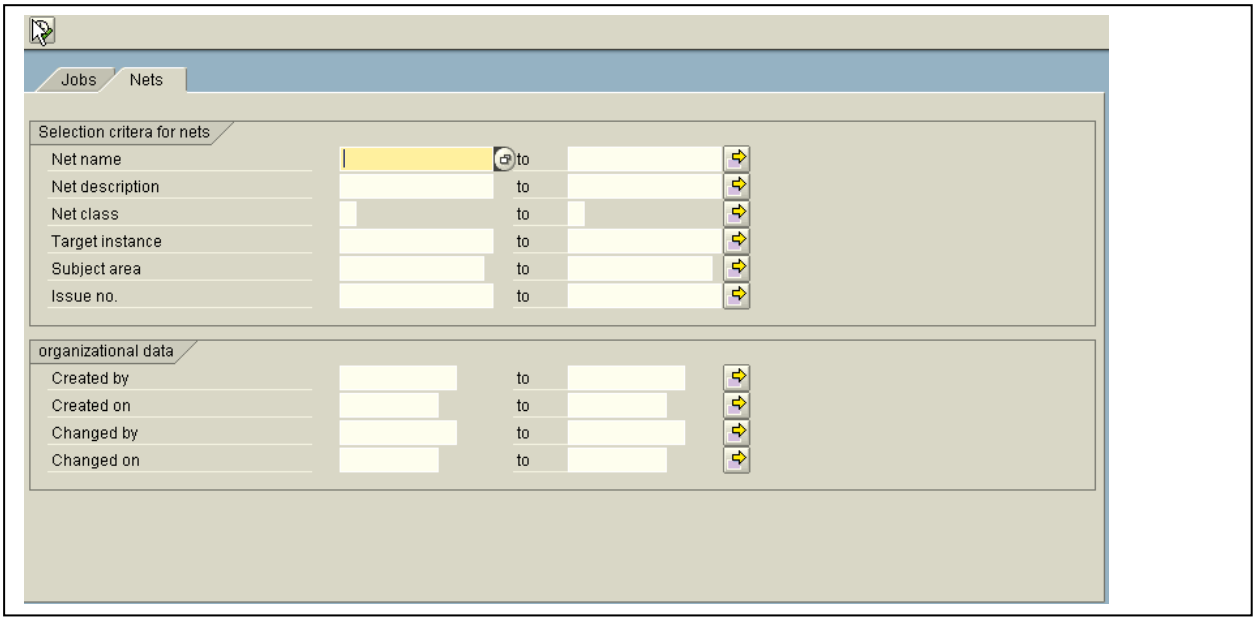
No start criteria found

Checkpoints

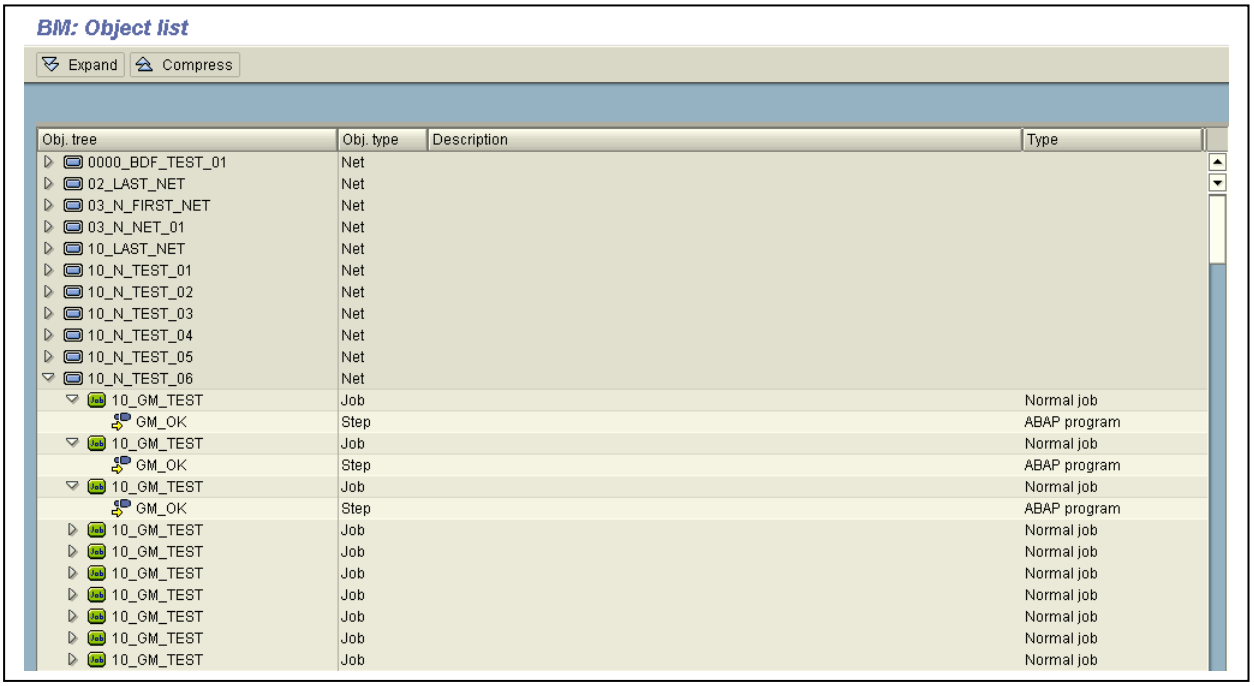
No checkpoints found

2.4.10.3. Net List with Jobs and Steps

Call a list for individual networks or all networks with their included objects and Sub-objects.



Result list:

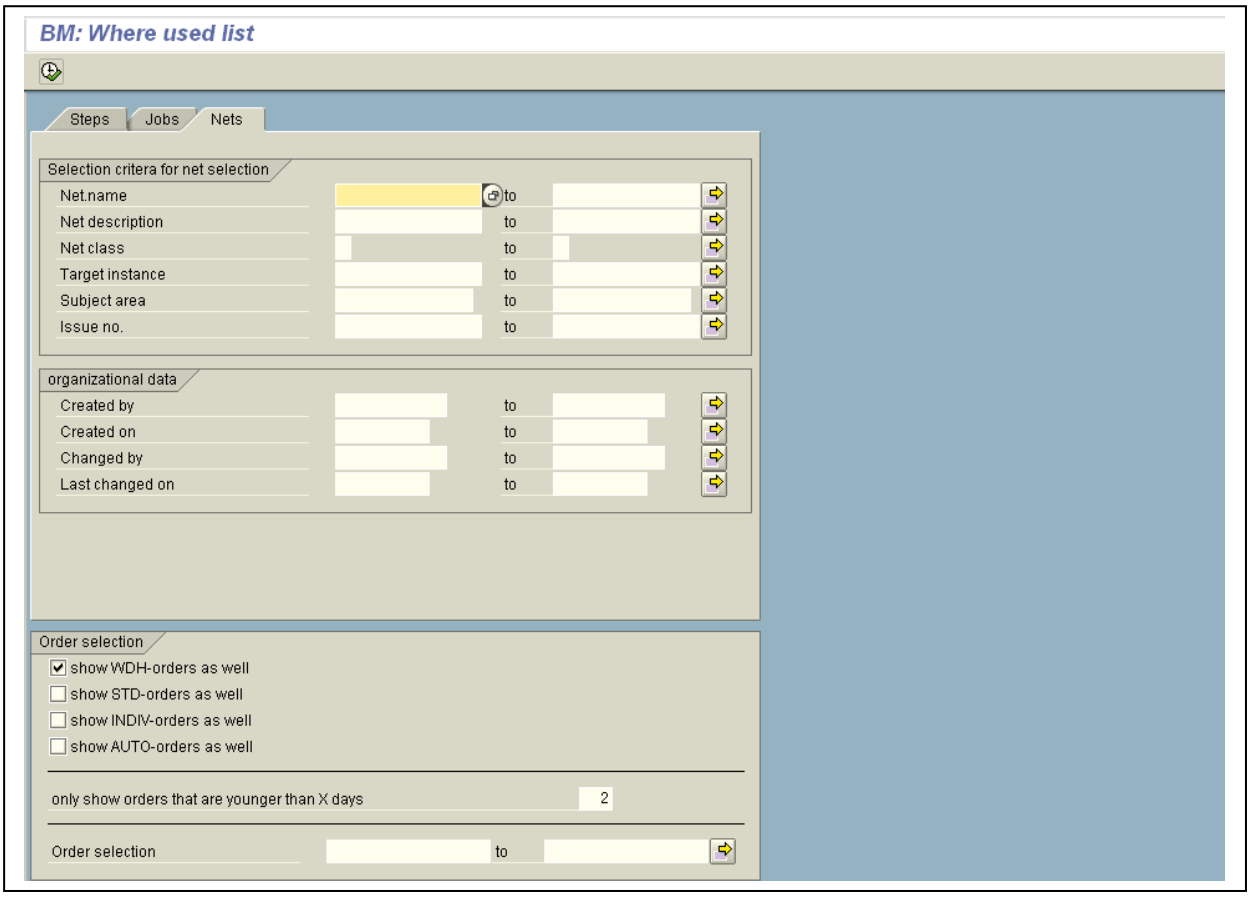


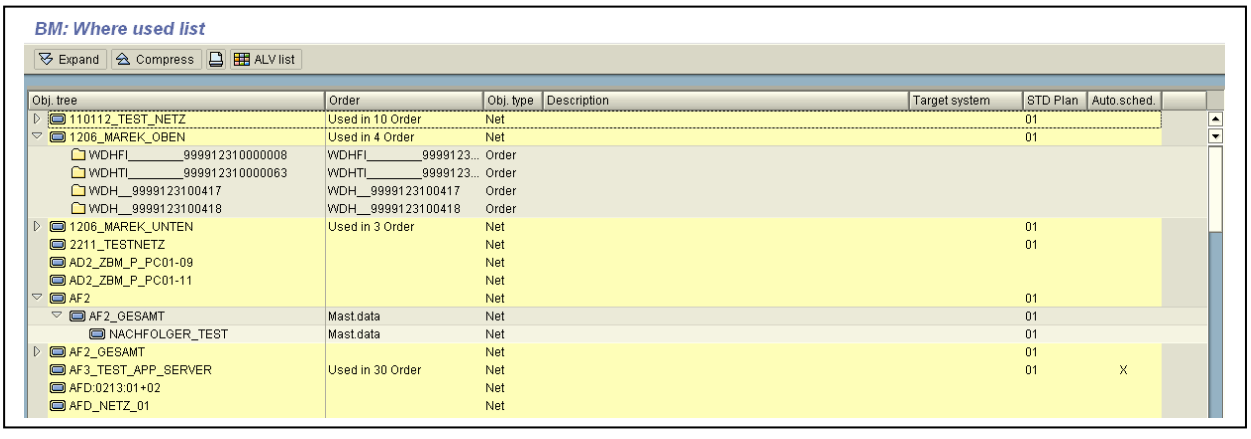
All objects can be displayed in detail by double-clicking on the name.

2.4.10.4. Where-used List for Networks

Calling the Where-used list for individual networks or all networks. This can be done for master data as well as for orders. In this case, the flag ‚Also show operator orders' has to be marked.

Via the field for operator orders, a restriction can be specified for certain orders. If only the networks should be displayed that enter STD orders, then the flag ‚Only automatically scheduled networks' has to be activated.



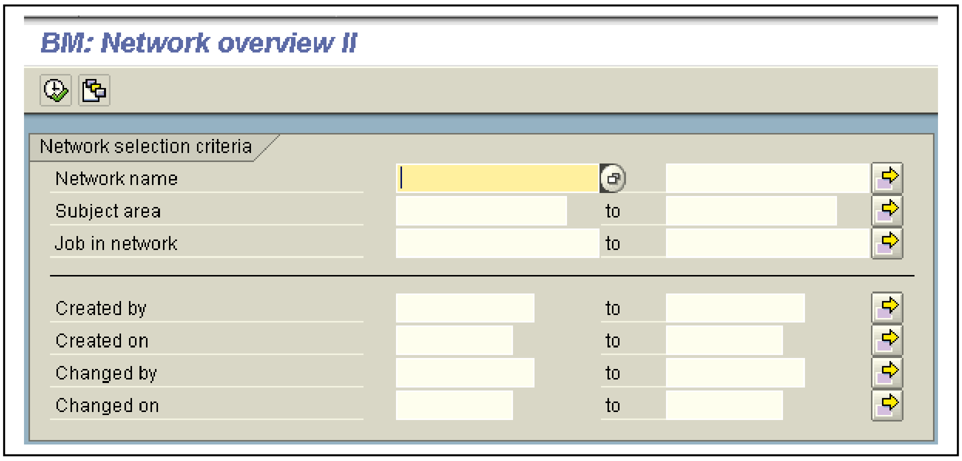


Result list:

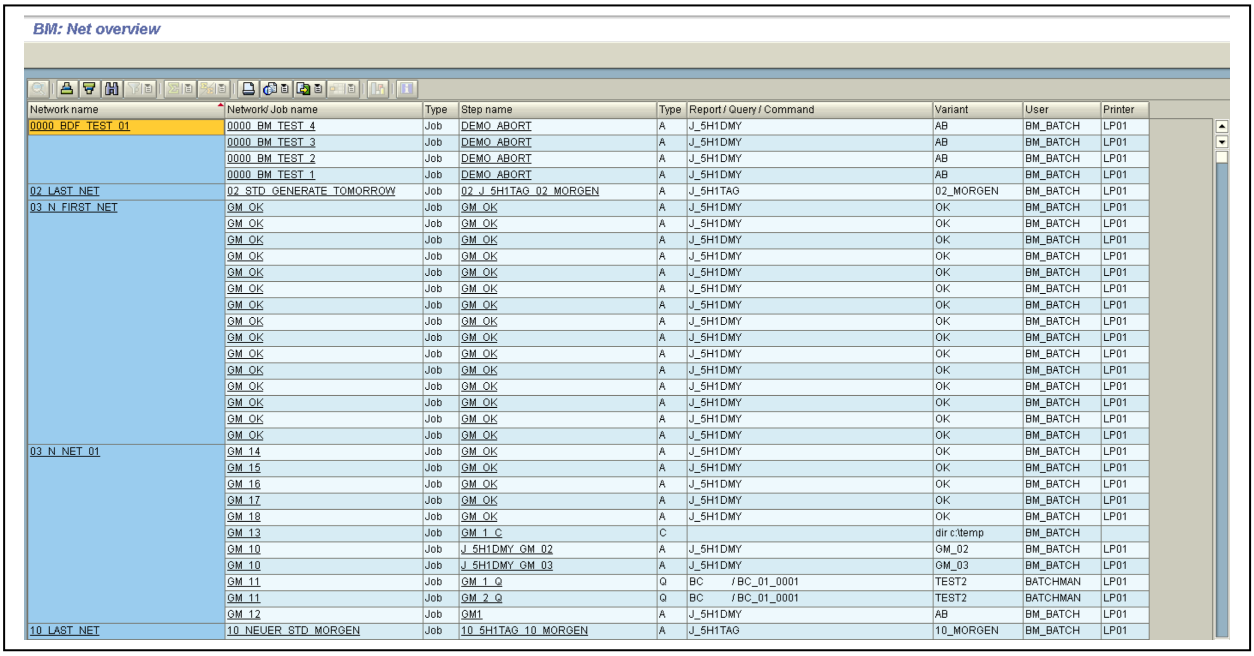
All objects can be displayed in detail by double-clicking on the name.

2.4.10.5. Net Overview II

This list offers an additional option to display networks with all included objects. If the Sub-object is a „normal job“, then the steps are displayed as well.



Result list:



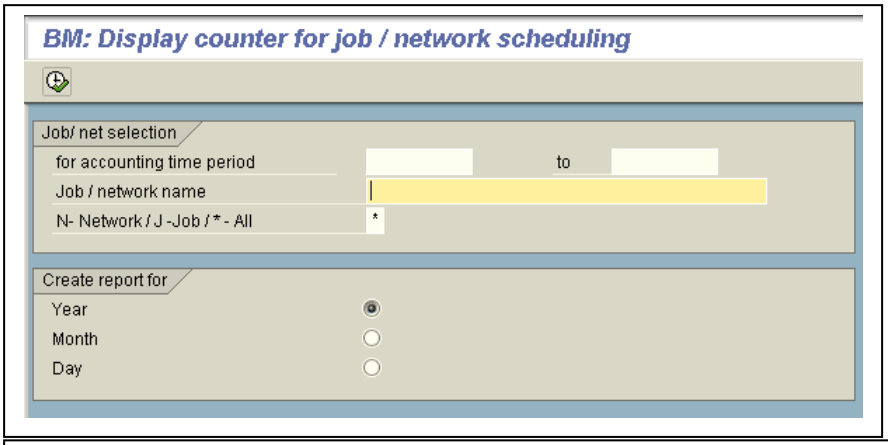
BM: Net overview

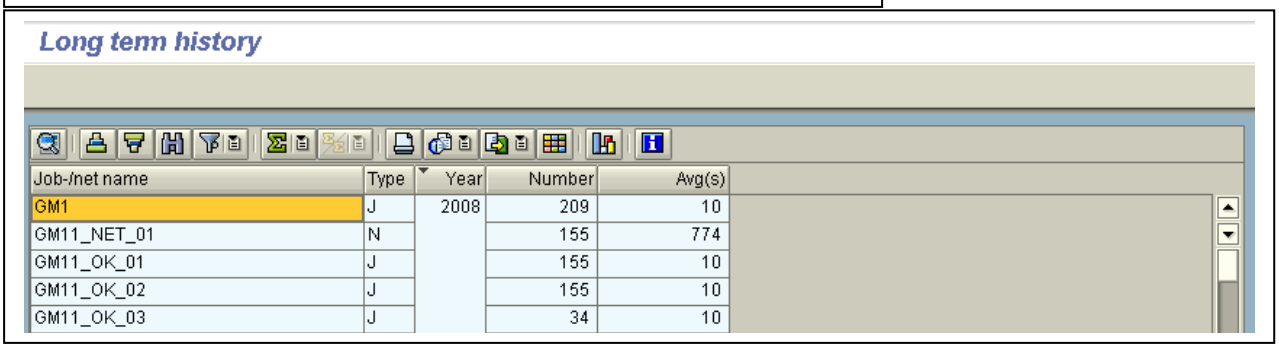
Network name	Network/ Job name	Type	Step name	Type	Report / Query / Command	Variant	User	Printer
0000_BDF_TEST_01	0000_BM_TEST_4	Job	DEMO_ABORT	A	J_5H1DMY	AB	BM_BATCH	LP01
	0000_BM_TEST_3	Job	DEMO_ABORT	A	J_5H1DMY	AB	BM_BATCH	LP01
	0000_BM_TEST_2	Job	DEMO_ABORT	A	J_5H1DMY	AB	BM_BATCH	LP01
	0000_BM_TEST_1	Job	DEMO_ABORT	A	J_5H1DMY	AB	BM_BATCH	LP01
02_LAST_NET	02_STD_GENERATE_TOMORROW	Job	02_J_5H1TAG_02_MORGEN	A	J_5H1TAG	02_MORGEN	BM_BATCH	LP01
03_N_FIRST_NET	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_OK	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
03_N_NET_01	GM_14	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_15	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_16	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_17	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_18	Job	GM_OK	A	J_5H1DMY	OK	BM_BATCH	LP01
	GM_13	Job	GM_1_C	C		dir c:\temp	BM_BATCH	
	GM_10	Job	J_5H1DMY_GM_02	A	J_5H1DMY	GM_02	BM_BATCH	LP01
	GM_10	Job	J_5H1DMY_GM_03	A	J_5H1DMY	GM_03	BM_BATCH	LP01
	GM_11	Job	GM_1_Q	Q	BC /BC_01_0001	TEST2	BATCHMAN	LP01
	GM_11	Job	GM_2_Q	Q	BC /BC_01_0001	TEST2	BATCHMAN	LP01
	GM_12	Job	GM1	A	J_5H1DMY	AB	BM_BATCH	LP01
10_LAST_NET	10_NEUER_STD_MORGEN	Job	10_5H1TAG_10_MORGEN	A	J_5H1TAG	10_MORGEN	BM_BATCH	LP01

All objects can be displayed in detail by double-clicking on the name.

2.4.10.6. Counting Table Network Scheduling

This report (/BTCMAN/M_B22) shows how of the selected objects (network or job) ran in a report period and which average run time results from it. Analysis per year, month or day is possible.



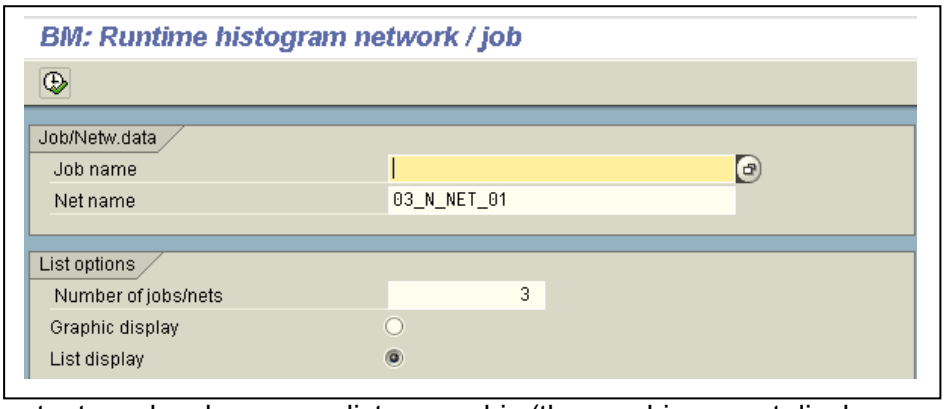


2.4.10.7. Laufzeithistogramm (Anzahl Läufe)

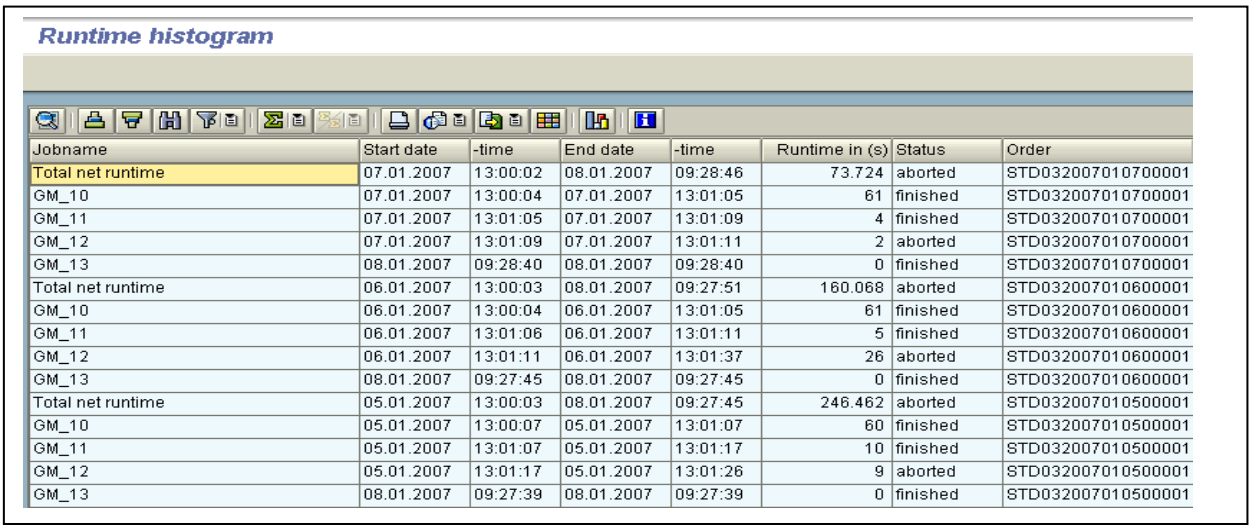
The run times of jobs can be analyzed with this report (/BTCMAN/M_LFZ).

Attention: This detailed information is only available if the orders are still contained in BatchMan; or if the option , Hold runtime histogram' was activated when BatchMan orders were archived.

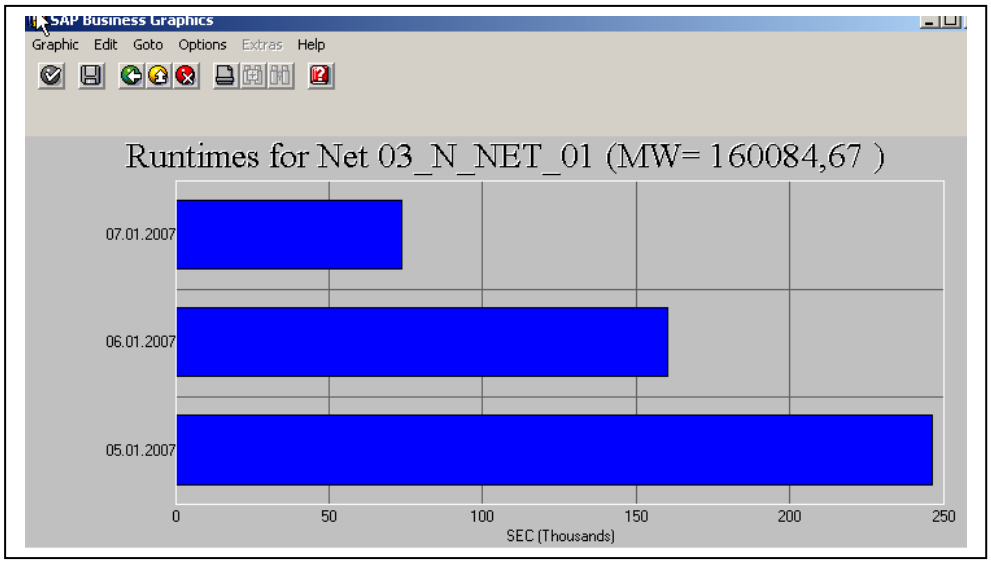
In this report, a statistical evaluation is carried out over individual runtimes of a selected job; the



output can be shown as a list or graphic (the graphic cannot display more than 30 runs).



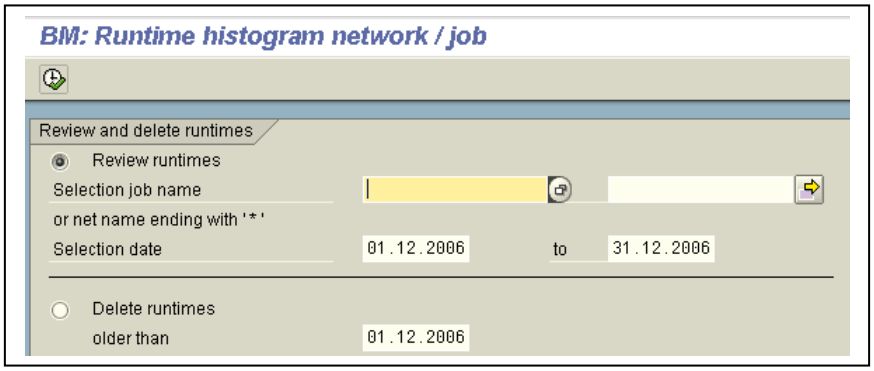
Jobname	Start date	-time	End date	-time	Runtime in (s)	Status	Order
Total net runtime	07.01.2007	13:00:02	08.01.2007	09:28:46	73.724	aborted	STD032007010700001
GM_10	07.01.2007	13:00:04	07.01.2007	13:01:05	61	finished	STD032007010700001
GM_11	07.01.2007	13:01:05	07.01.2007	13:01:09	4	finished	STD032007010700001
GM_12	07.01.2007	13:01:09	07.01.2007	13:01:11	2	aborted	STD032007010700001
GM_13	08.01.2007	09:28:40	08.01.2007	09:28:40	0	finished	STD032007010700001
Total net runtime	06.01.2007	13:00:03	08.01.2007	09:27:51	160.068	aborted	STD032007010600001
GM_10	06.01.2007	13:00:04	06.01.2007	13:01:05	61	finished	STD032007010600001
GM_11	06.01.2007	13:01:06	06.01.2007	13:01:11	5	finished	STD032007010600001
GM_12	06.01.2007	13:01:11	06.01.2007	13:01:37	26	aborted	STD032007010600001
GM_13	08.01.2007	09:27:45	08.01.2007	09:27:45	0	finished	STD032007010600001
Total net runtime	05.01.2007	13:00:03	08.01.2007	09:27:45	246.462	aborted	STD032007010500001
GM_10	05.01.2007	13:00:07	05.01.2007	13:01:07	60	finished	STD032007010500001
GM_11	05.01.2007	13:01:07	05.01.2007	13:01:17	10	finished	STD032007010500001
GM_12	05.01.2007	13:01:17	05.01.2007	13:01:26	9	aborted	STD032007010500001
GM_13	08.01.2007	09:27:39	08.01.2007	09:27:39	0	finished	STD032007010500001

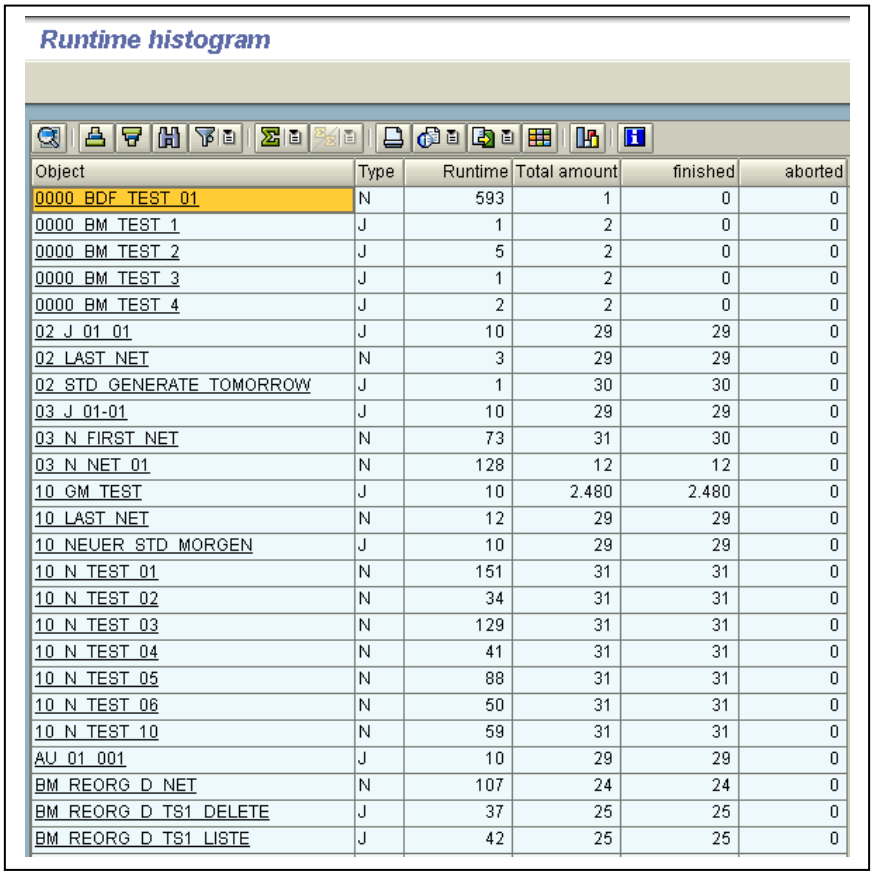


The number of networks determines the average mean value.

2.4.10.8. Run Time Histogram (Date)

In this report (/BTCMAN/M_LFG), a statistical evaluation about individual run times of a selected time period is done. In the output list, you can jump via double click in the detailed list (Run time histogram (number of runs). The number of jobs determines the average run time in this case.





Runtime histogram

Object	Type	Runtime	Total amount	finished	aborted
0000_BDF_TEST_01	N	593	1	0	0
0000_BM_TEST_1	J	1	2	0	0
0000_BM_TEST_2	J	5	2	0	0
0000_BM_TEST_3	J	1	2	0	0
0000_BM_TEST_4	J	2	2	0	0
02_J_01_01	J	10	29	29	0
02_LAST_NET	N	3	29	29	0
02_STD_GENERATE_TOMORROW	J	1	30	30	0
03_J_01-01	J	10	29	29	0
03_N_FIRST_NET	N	73	31	30	0
03_N_NET_01	N	128	12	12	0
10_GM_TEST	J	10	2.480	2.480	0
10_LAST_NET	N	12	29	29	0
10_NEUER_STD_MORGEN	J	10	29	29	0
10_N_TEST_01	N	151	31	31	0
10_N_TEST_02	N	34	31	31	0
10_N_TEST_03	N	129	31	31	0
10_N_TEST_04	N	41	31	31	0
10_N_TEST_05	N	88	31	31	0
10_N_TEST_06	N	50	31	31	0
10_N_TEST_10	N	59	31	31	0
AU_01_001	J	10	29	29	0
BM_REORG_D_NET	N	107	24	24	0
BM_REORG_D_TS1_DELETE	J	37	25	25	0
BM_REORG_D_TS1_LISTE	J	42	25	25	0

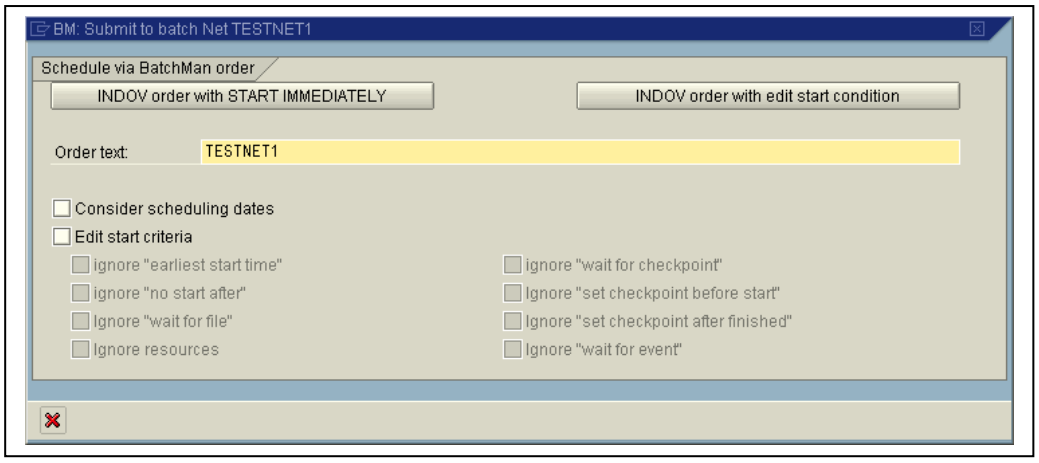
With this report, the runtime histograms are also deleted if they were not deleted during archiving.

2.4.11. Scheduling of Ad Hoc Processing: Networks

BatchMan offers the option for quick scheduling of networks into the batch system via the button ‚Submit to Batch'. In case of networks, scheduling can only occur via **BatchMan** orders.

In this case, an INDIV order is crated fort his particular network. The text of the order can be changed. The scheduling option can also be modified.

Note: The function ‚Submit to Batch' is only active in the **display mode** of the network maintenance. After pushing ‚Submit to Batch', a screen appears to request further selections.



If the "Consider Scheduling Dates" flag is activated, the scheduling dates of the object are taken into account.

For the scheduling, you may deactivate individual start criteria for this scheduling via the flag "*Edit start criteria"*.

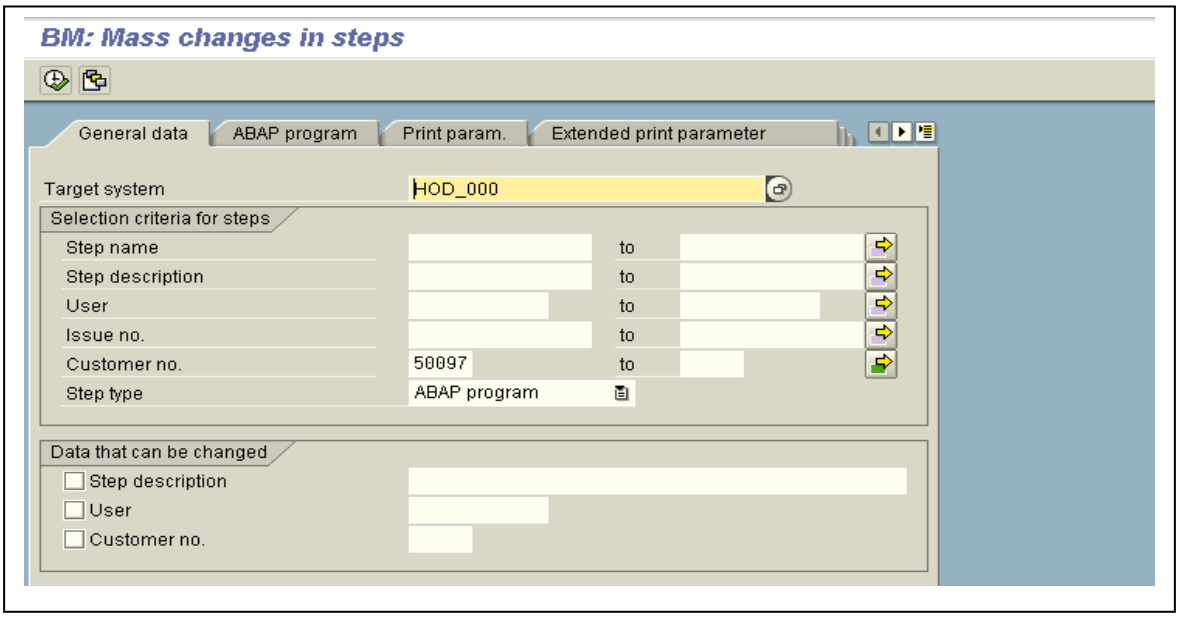
2.5. Mass Changes

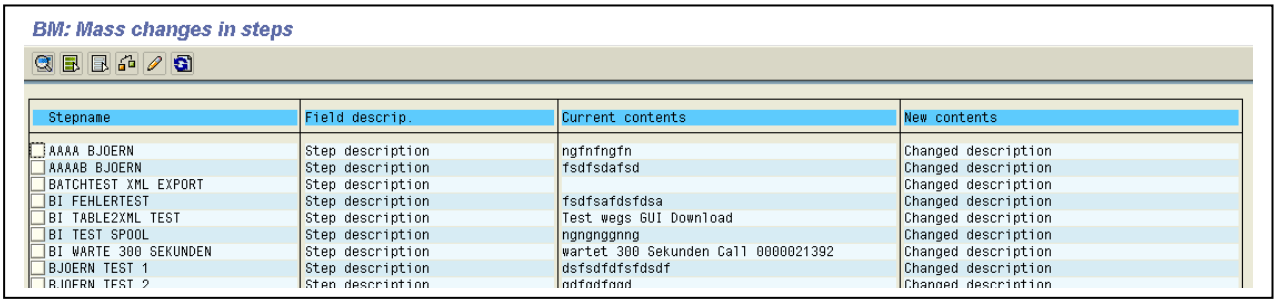
The mass changes in the master data are pure dialog programs (exception batch program /BTCMAN/M_B24). If these functions are called to carry out mass changes, the check functions must be called before the actual changes.

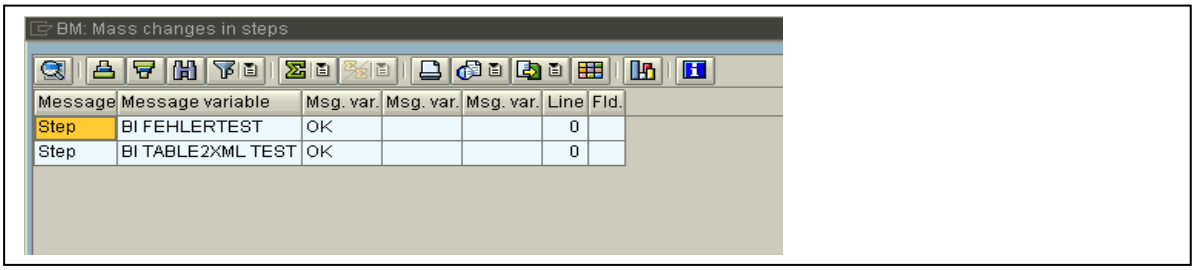
2.5.1. Steps: Mass Changes

This program change (among other things) program name, variant, language and user for the selected steps. It can be uses for post processing after a step transport. Specification of the ,**target system** (RFC destination) is required because the values are checked for validity.

The selected ,**Step type'** determines which file card can be selected.



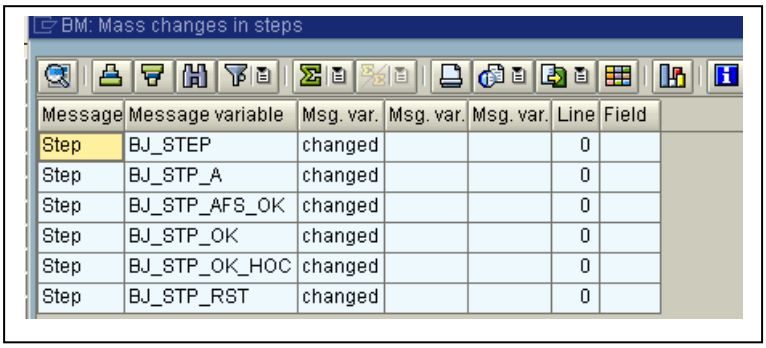




Example: Changing the field description produces the following list

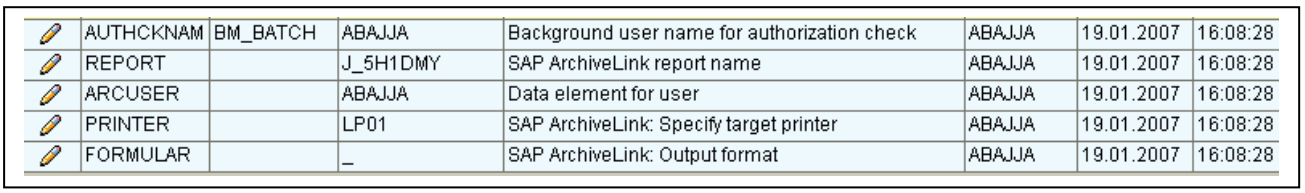
Marking all steps and clicking on the icon "check" produces:

All checks are “OK.” Prior to the actual change, the system generates an additional query (popup window) asking whether you really want to make changes. After you confirm with “Yes,” the changes are executed. Result:



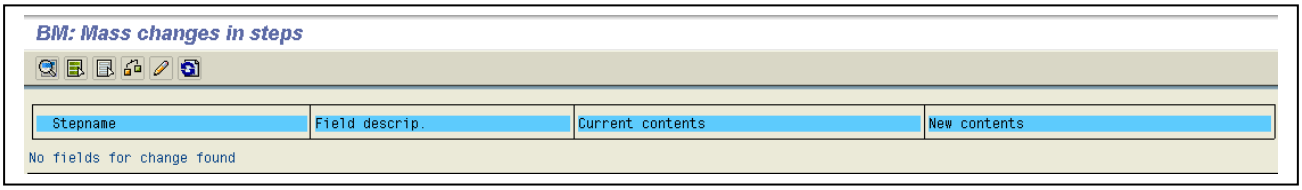
Message	Message variable	Msg. var.	Msg. var.	Msg. var.	Line	Field
Step	BJ_STEP	changed			0	
Step	BJ_STP_A	changed			0	
Step	BJ_STP_AFS_OK	changed			0	
Step	BJ_STP_OK	changed			0	
Step	BJ_STP_OK_HOC	changed			0	
Step	BJ_STP_RST	changed			0	

The changes are registered in the change history, and the fields of the last changes (who, date time) are updated.



	AUTHCKNAM	BM_BATCH	ABAJJA	Background user name for authorization check	ABAJJA	19.01.2007	16:08:28
	REPORT		J_5H1DMY	SAP ArchiveLink report name	ABAJJA	19.01.2007	16:08:28
	ARCUSER		ABAJJA	Data element for user	ABAJJA	19.01.2007	16:08:28
	PRINTER		LP01	SAP ArchiveLink: Specify target printer	ABAJJA	19.01.2007	16:08:28
	FORMULAR		_	SAP ArchiveLink: Output format	ABAJJA	19.01.2007	16:08:28

The list of the selected steps is empty if all steps were changed.



2.5.2. Jobs: Mass Changes

You may carry out extensive changes regarding job class, job text, recovery job, target system, etc. The jobs displayed in the list may be restricted on the selection screen. In the list, those jobs that are to be changed can be selected again by flagging them.

Otherwise, the handling is as described in “Mass change: Steps.“

2.5.3. Network: Mass Changes

For networks, the contents of the fields may be changed. The networks displayed in the list may be restricted on the selection screen. In the list, those networks that are to be changed can be selected again by flagging them.

Otherwise, the handling is as described in “Mass change: Steps.”

2.5.4. Jobs/Networks: Mass Changes (Batch)

Since the above mass changes are not batch-capable because screens are used, you can change the target instance in a batch run with the program /BTCMAN/M_B24 (e.g. after transport of BatchMan master data from a quality system to the productive system.)

Using the “**Network”** and “**Job”** pushbuttons, you may switch the view to the master data objects. For the network, fewer fields may be changed.

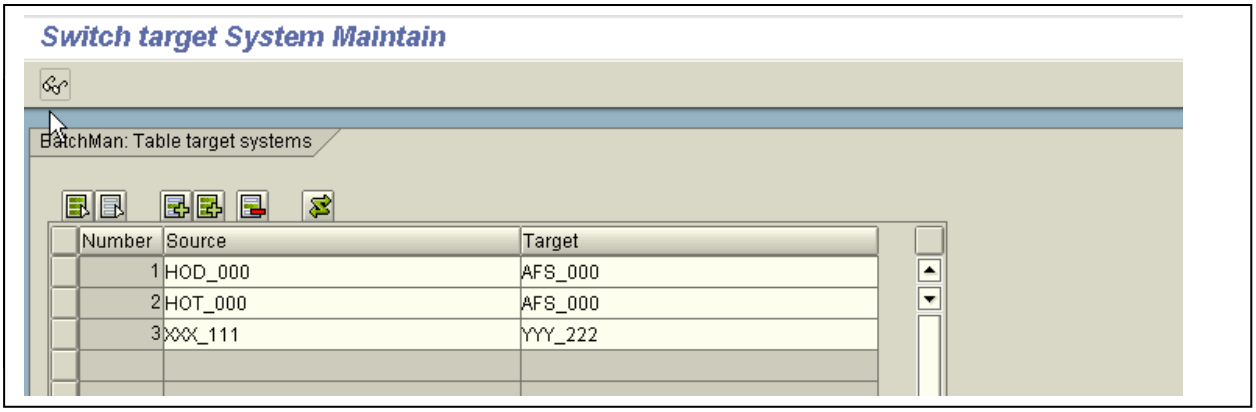
2.5.5. Switching target systems (repeating orders / networks / jobs)

With this function, it is possible to switch target systems and RFC connections in the jobs, networks and repeating orders.

The function is used with

- System copies
- Transport of master data from development system or quality system to the production system

Prerequisite is a migration table maintained for target systems.



The program switches all connections in one run, i.e. jobs, networks and repeating orders.

In addition to switching target systems, empty connections may be filled with the default connection.

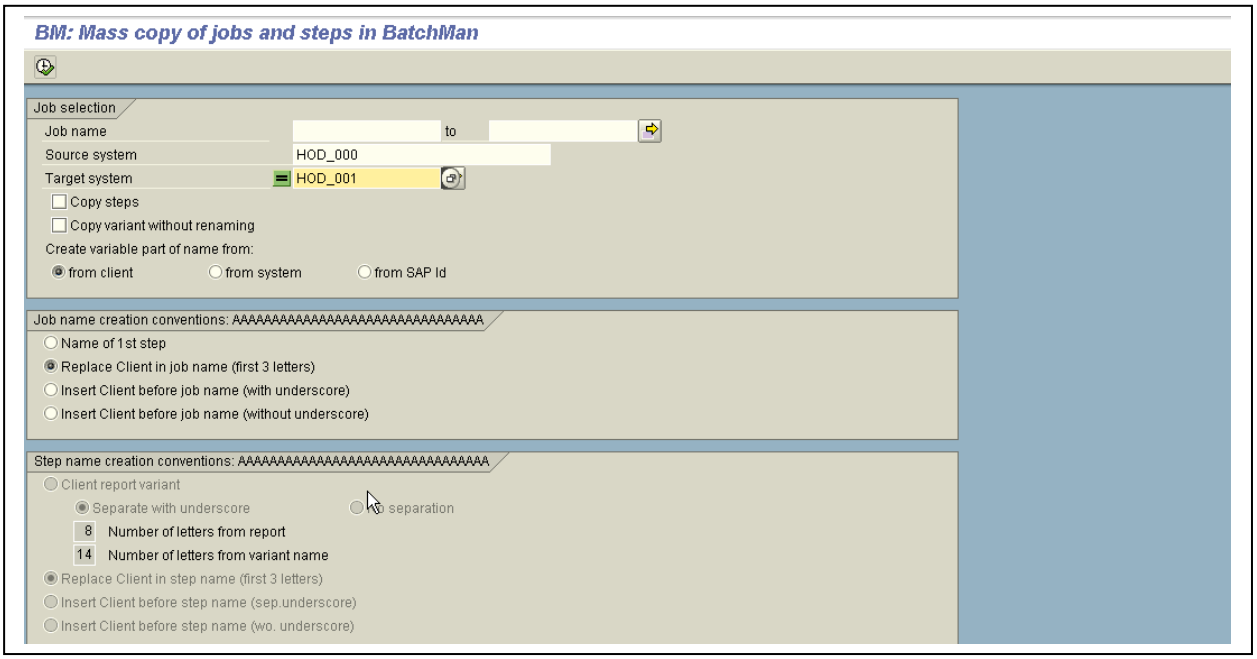
Also, you may switch the RFC connection in the definition of the alert tool (type script). This is only required if an alert is to call the copy on another server than for the original. In general, this is not required.

2.6. Copying -/ Renaming Functions

2.6.1. Copying Jobs (with Steps + Variants)

In some SAP Systems the same runs/processing flows are scheduled and executed for different clients. Each customer has his own client in this case.

With this program, it is possible to copy jobs and steps within the **BatchMan** master data from one client to another (based on the target systems). Variants may also be copied. If the contents of the variants are identical, system variants should be used in the customer namespace (CUS&...), however.



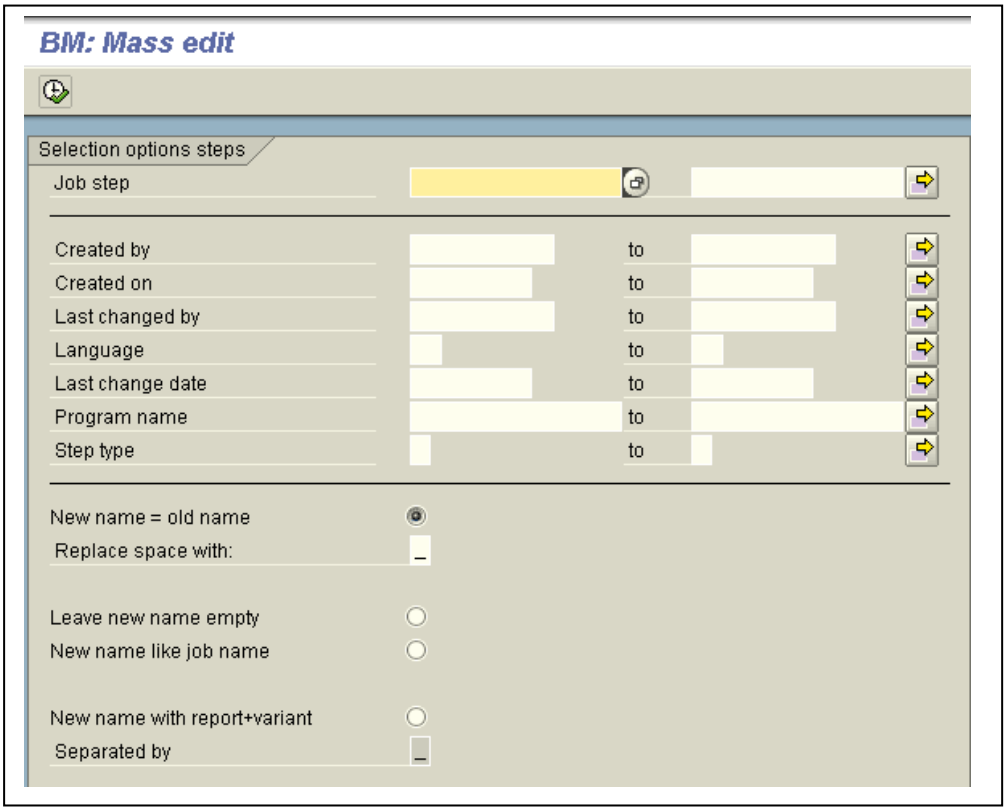
The jobs from the system AF of client ‚000' are copied to client ‚001'. The variants are also copied from the source system to the target client.

The steps are not copied. An existing step is used in several jobs. The target system of the job is the decisive factor for the execution.

2.6.2. Copying/ Renaming of Steps

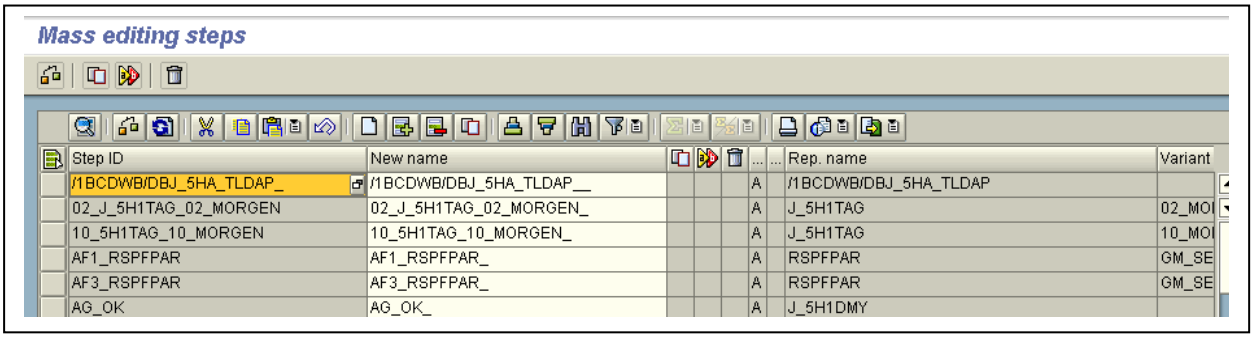
With this program, steps may be renamed in the master data. Also copying and deleting are possible. The selection must be executed accordingly; whereby the new step names may be preassigned:

- new step names preassigned with the old step names (if only one character is to be inserted or exchanged). blank characters can be replaced with a selectable character
- new step names remain empty
- new step names are formed from the job name. If the step is used in several jobs, the name of the first job is used.
- new step names are formed from the report and variants, each separated by underscores.



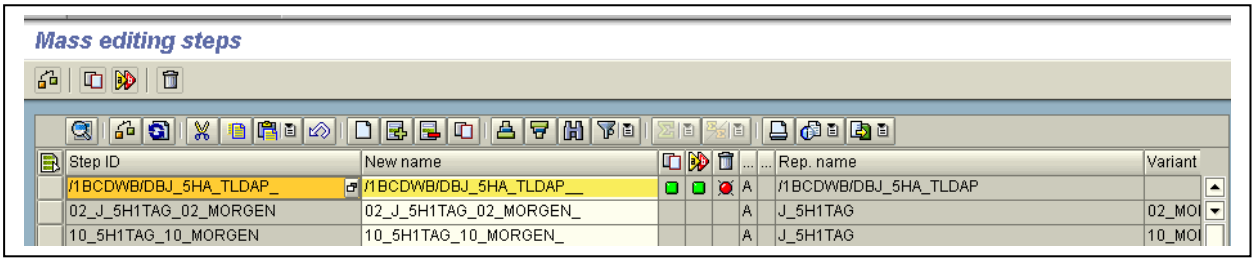
Afterwards a list is displayed in which the step names can be changed. In the example selection with ‚new name = old name', the separator is automatically added. Steps that should be renamed must be marked.

Before the actual renaming is done, the steps should always be checked (icon ‚check').



After checking is done, the columns for ‚Copy status' ‚Rename status' and ‚Delete status' are updated with red or green icons.

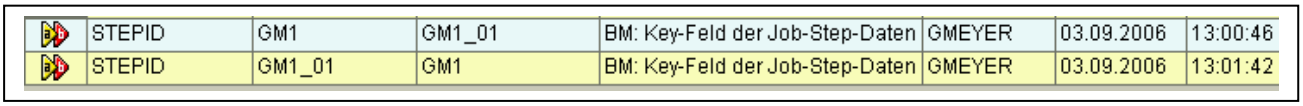
- Green: Action can be executed.
- Rot: Aktion kann NICHT ausgeführt werden. (Löschen ist im Beispiel nicht möglich, da der Step eine Verwendung in Jobs hat.)



Now the actual change can be executed. A confirmation prompt does not appear.

	Checking marked steps
	Copying marked steps
	Renaming marked steps
	Deleting marked steps

The changes are registered in the change history and the fields of the last changes (who, date and time) are updated.



2.6.3. Copying/ Renaming Jobs

Jobs in the master data can be renamed with this program. Copying and deleting is possible as well. The selection must be done accordingly, whereby the new job names can be pre-selected/set up as default.

Afterwards a list is displayed in which the job names can be changed. Jobs should always be checked (icon 'check') before the actual renaming is done.

Further handling can be done as described in ‚Renaming steps'.

2.6.4. Copying/ Renaming of Networks

Networks in the master data can be changed with this program. Copying and deleting is possible as well. The selection must be done accordingly, whereby the new network names can be pre-selected/set up as default.

Afterwards a list is displayed in which the network names can be changed. Networks should always be checked (icon 'check') before the actual renaming is done.

Further handling can be done as described in ‚Renaming steps'.

2.7. Transport

BatchMan supports the SAP idea of a system environment in which data is transferred between the development-, quality-, and productive system via transports. For this purpose, two different transport options are supplied. On one hand the transport option of SAP is used, on the other hand a flat file is used as tool for data transfer.

The second option also offers the possibility to save a certain status/version in a system. This can, for example, be used in a Q-System in order to secure/safe the master data before a P-System is copied back to the Q-System.

2.7.1. Transport from Master Data Maintenance

In the master data objects job and network, a ‚truck-button' is active in the display mode. Via this button, objects can be picked up in an SAP transport order. The pushing of the button calls up an input window in which the existing transport order number can be entered or a new transport order can be created.

The transport can be imported to the target system after it has been released.

2.7.2. Transport via File

In the SAP required 3-system architecture it is necessary to transport steps, networks and jobs that were created in the test or consolidation system, into the productive system. Via this function a file is created in a first step, which is then read again in the second step in the subordinate system.

In addition, this function can also be used to secure/safe the master data. Call:

*Function tree **BatchMan** → Master data / Library → Import Jobs from SAP/ Transport tools → Transport Master Data via File*

In the field ‚file name', a logical file name and path can be used as well. The old values with absolute specifications (path and file name) still work.

- <L= logical file > defines a logical file name (transaction FILE)
- <A =logical path > defines a logical path (transaction FILE)

  Example: <A=SAVE_PATH>transportfile.txt

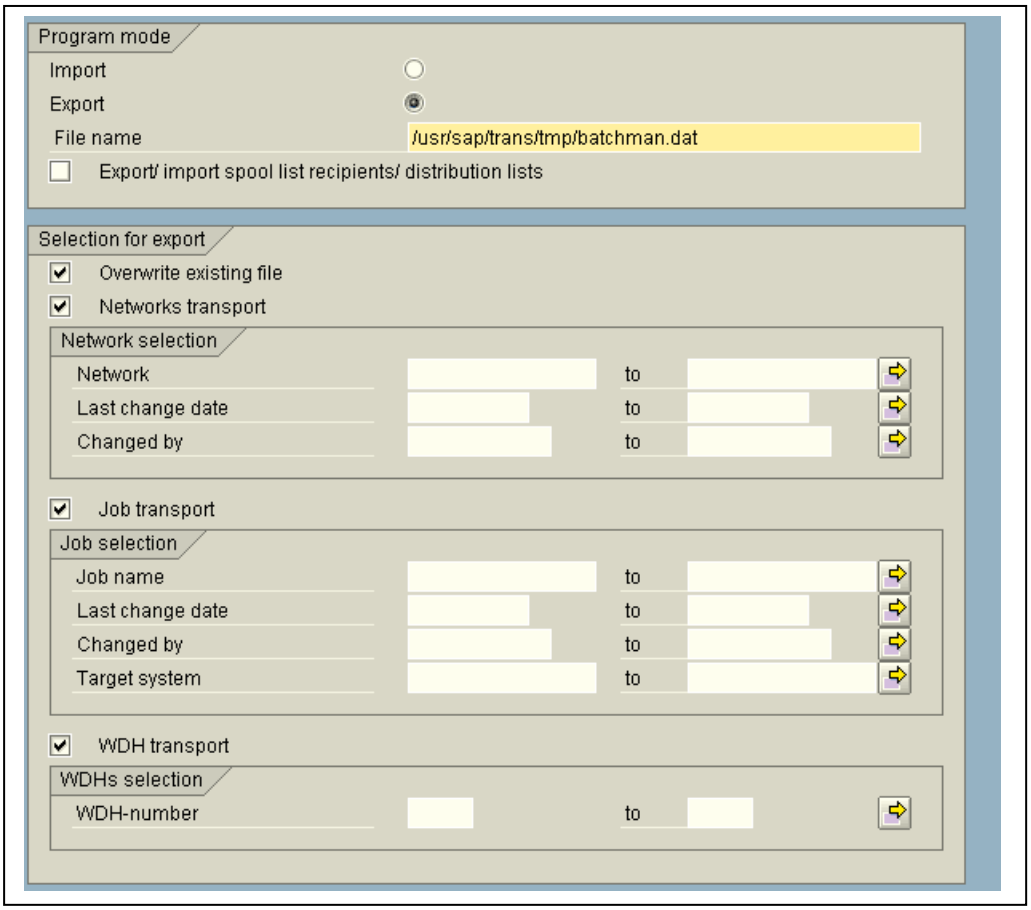
2.7.2.1. Exporting Objects

The **Export** function enables you to export selective steps, jobs, networks and repeating orders in a sequential file, which can be read in the target system with the “Import” function.

Using the “Export/ Import” flag, you can select whether the spool list recipients are also to be exported. These spool list recipients may be inconsistent in the target system since the assignment between the SAP user names, distribution list names, etc. may be displayed by different internal numbers in two different SAP systems. Then the spool list recipients are to be adjusted using the mass change in the target system.

Using the setting in the selection parameters, it is possible to make a preselection for certain objects and also to select within the objects by name conventions.

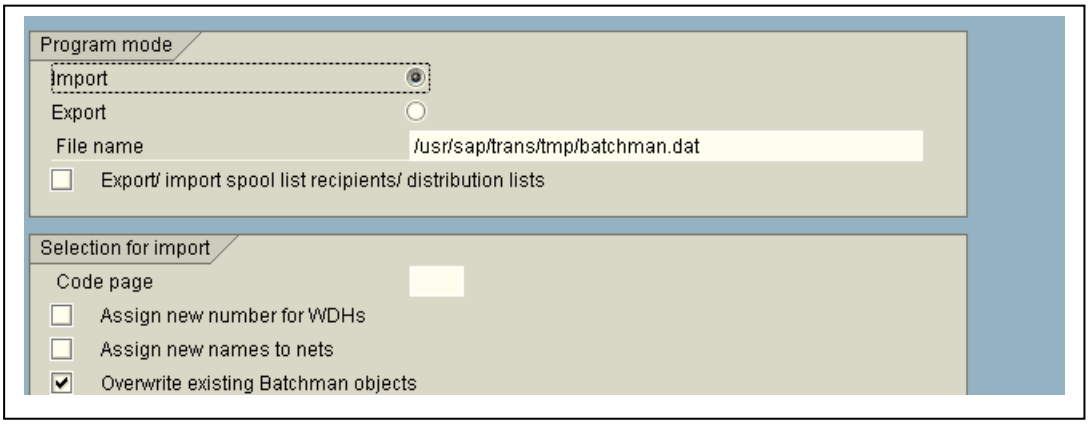
The code page has no significance for the export; the export always goes to the system code page.



When you execute the report in the batch mode, all objects are exported from the selection list.

2.7.2.2. Importing Objects

Using the “Export/ Import” flag, you can select whether the spool list recipients are also to be imported. These spool list recipients may be inconsistent in the target system since the assignment between the SAP user names, distribution list names, etc. may be displayed by different internal numbers in two different SAP systems. Then the spool list recipients are to be adjusted using the mass change in the target system.



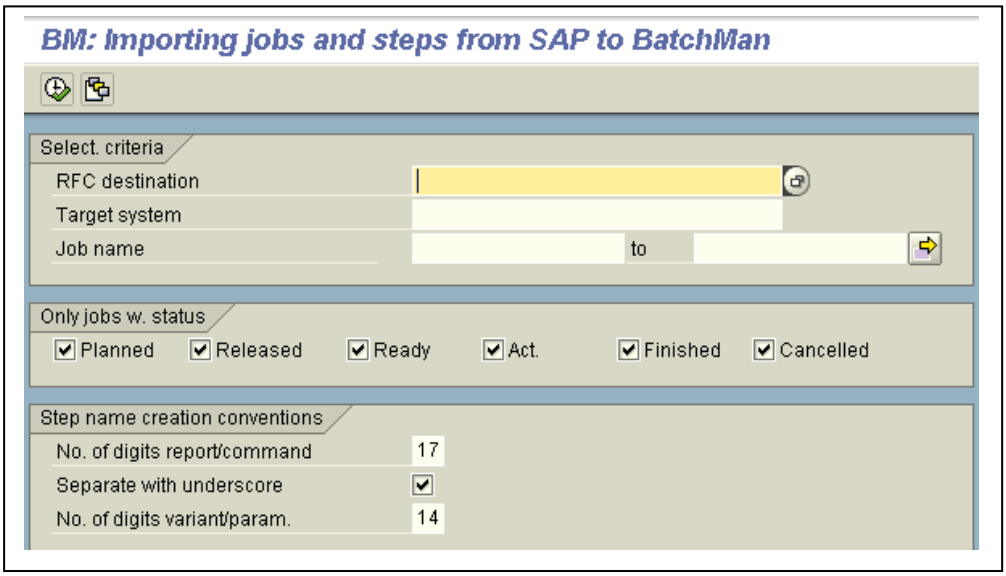
During the execution of the report in batch, all objects from the selection list are imported. You can select if objects should be overwritten. If ‚Overwrite' is selected, then the objects are first deleted and then created new.

The imported jobs contain the target system from the exported system. Here, probably other target systems are wanted in the imported system. For this, you may execute the switch according to the migration table or by manual entry in the selection options. (Although the field is gray, it may be changed by choosing the  pushbutton.)

2.8. Importing Jobs and Steps from SAP to BatchMan

Under this menu option, jobs with their steps can be imported from SAP to **BatchMan** to be available as permanent data of the library to the operator orders. Here, a folder is created, which is imported with "Call transaction."

The start dates and period values are not transferred by SAP since these are also different from **BatchMan**. This additional data for **BatchMan** is to be maintained there. In this case, by linking jobs in networks, the option is available to provide all jobs with a start date and a period via the network.



In SAP, steps have to be created new for each job even though report and variants are the same; and SAP does not know fixed step names. For **BatchMan**, these step names are created during import. Via the naming conventions you can influence the step names.

Note: If external commands or programs are imported, generally parameter 1 is filled with the sequential number internally in the report. This is required since a unique step name formation is not possible with program/ command and related parameter.

on the client's set up.

3. Order Administration

3.1. Introduction: Operator Order

In **BatchMan,** an "**order"** is the formation of manually or automatically compiled work packages to carry out a job scheduling.

The physical execution of the processes scheduled (normal jobs, BW processes, etc.) take place within the orders. An order is thus the prerequisite to execute a job. An order is identifiable by its unique order number. This consists of

- order type with extention
- the scheduled date / a fixed date
- A sequential number (the order number is assigned by the system.)

The following order types exist:

• Standard		**STD + eight-digit plan number / extention**
• Individual	normal	**INDIV**
	Ad-Hoc	**INDIV*_ADHOC_***
	Remote	**INDIV*_REMOTE_***
• Wiederholung	time periodical	**WDHTI + eight-digit possible further Splitting**
•Wiederholung	file periodical	**WDHFI + eight-digit mögliche further Splitting**
•Wiederholung	event periodical	**WDHEV + eight-digit mögliche further Splitting**
•Wiederholung	event periodical	**WDHEH + eight-digit mögliche further Splitting**
• Automatic		**AUTO_**

The work packages consist of sequences of jobs and networks (objects). Jobs from various systems and clients can be scheduled. In this case, the objects scheduled in an order are copies of the master data objects.

The order maintenance can be accessed either

- directly with the transaction J5HDN or
- via the function tree: *Order Administration → Maintain orders*

Before orders are created, you should find out about the possible extentions. The order extention later allows a more detailed view of the monitor.

The order type must always exist first, before an order can be created and maintained.

3.2. Order Types and extention

Function tree BatchMan® → Customizing → Basis Customizing → Order types

For each order type present in the system, there is a descriptive text. The system displays this when you call search help, and you may change it here to the language of your choice. If the placeholder <DATE> is used in the description text, then this place holder is overwritten with the system date when the orders are scheduled automatically.

For the order type AUTO order, no extention is possible; it is just an order that is formed by the system.

The order types starting with INDIV and WDH need no extention. They can be used also without an extention.

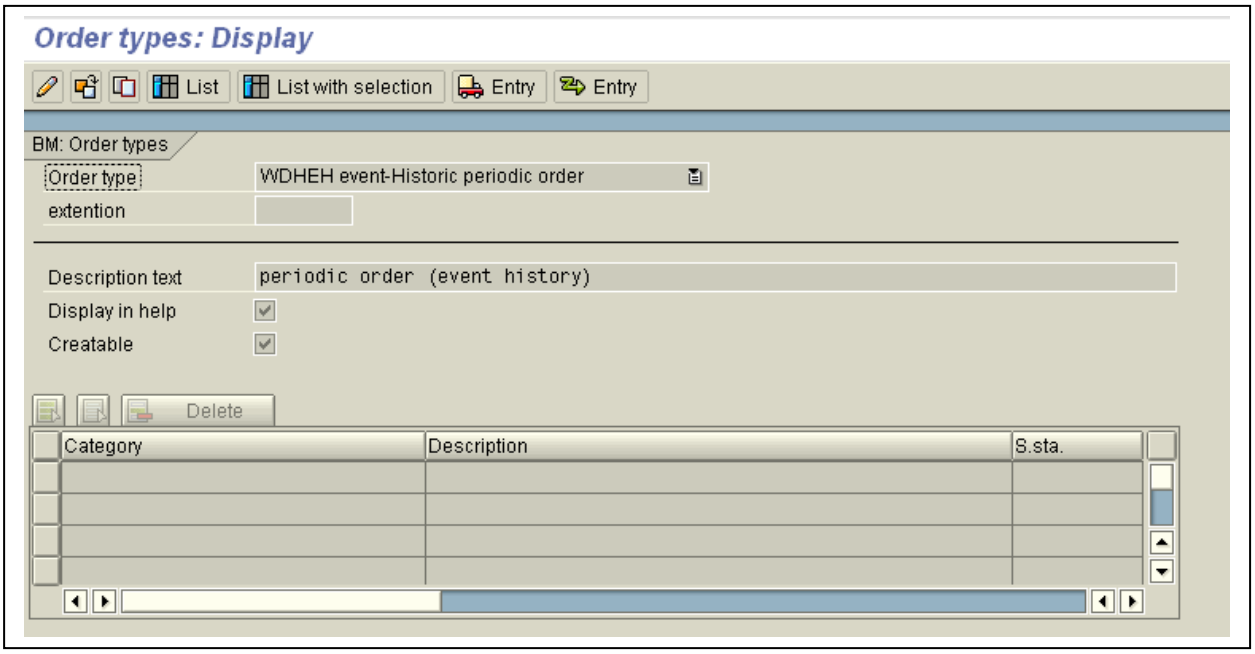
For the INDIV orders, HONICO provides the extentions.

For WDH orders, the extention can be used to compartmentalize the orders further such as by countries, departments, or other classification keys.

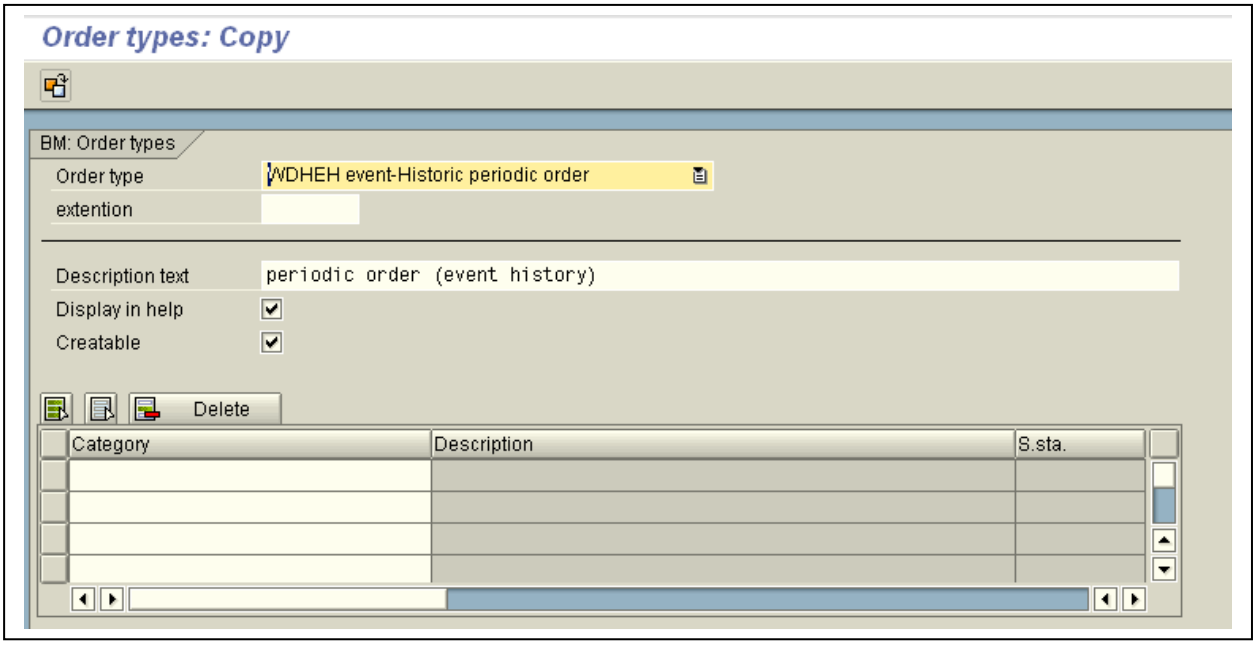
For STD orders, the division is done the same as for the WDH orders. Here the STD orders have a special feature that the extention is also called STD PLAN and can be entered for networks and jobs.

This entry causes the system to read all master data with this plan and to check the scheduling when automatically creating STD orders.

An order extention is created as a copy of an existing entry, here as a copy of the WDHEH (event-periodic WDH order).



The entry is possible after you choose “Copy”.  .


Order types: Copy
BM: Order types
Order type
WDHEH event-Historic periodic order
extention
Description text
periodic order (event history)
Display in help
Creatable
Delete
Category
Description
S.sta.

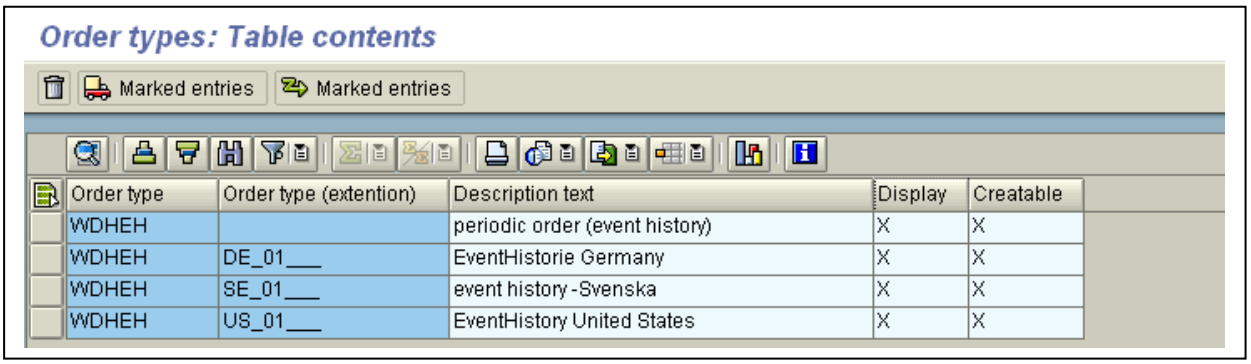
Maintain text. After you maintain data, then save. Any blank in the expansion is replaced by an underscore

“**Display in Help**” flag. This indicates whether or not the order type is displayed in the help.

“**Createtable**” flag: This indicates whether or not this order can be created in the order maintenance.

After maintenance, then the individual order types can be used with the order extentions.

Select the list via “List with selection”.



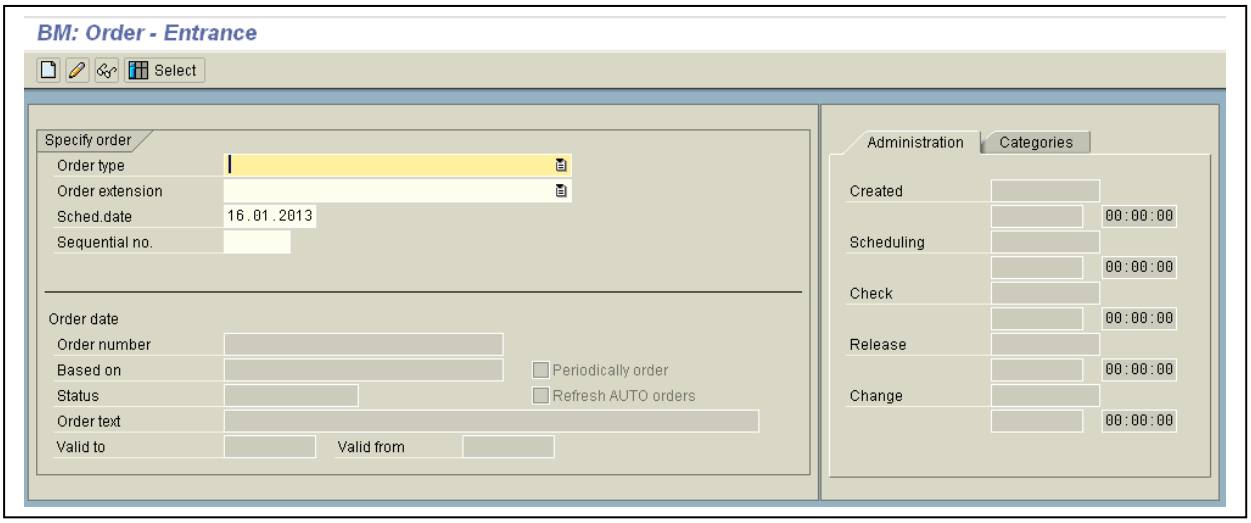
Order type	Order type (extention)	Description text	Display	Creatable
WDHEH		periodic order (event history)	X	X
WDHEH	DE_01___	EventHistorie Germany	X	X
WDHEH	SE_01___	event history - Svenska	X	X
WDHEH	US_01___	EventHistory United States	X	X

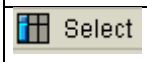
If, as in the example, the classification is made by countries, this should also be done for the other order types.

In the monitor, you can then select by country with DE* in the order extention.

3.3. Order Identification

The first screen shot of the order administration looks as follows. The screen changes according to the selected order type.



	Other order, Call INIT mode of order maintenance
	Create order, first select the **order type** in the list box
	Edit an order. If this is still possible depends on the status of the order.
	Display of an order
	Select an order, first select the **order type** in the list box and possibly the date for **scheduled date**

Order Types

The order type serves to classify orders. The following order types are offered:

STD – STANDARD ORDER

The standard order places all the jobs and job networks for a plan automatically for selection in a proposal list for which:

a) the field "Automatic scheduling" was marked and

b) the scheduling date corresponds with the scheduling day of the operator order.

c) the corresponding plan is maintained

d) the scheduling day is within the validity period

The order type for standard orders is composed of the abbreviation ‚STD' and a eight-digit plan number. That means that at first the plan (see STD Plans/ Shift plans) must be maintained where time, release and start specifications are entered.

In the STD orders, all jobs and networks can run with a period of one day and more. So that networks and jobs are used automatically for this order, the scheduling periodicity must be maintained in the library objects. The scheduler can revise this proposal list, i.e. he/she can add and remove jobs and networks and change the sequence and relationships of the nodes.

INDIV – INDIVIDUAL ORDER

An individual order is created completely manually including the scheduling and is the responsibility of the scheduler. The scheduler must decide which jobs and networks are to be scheduled how when and in which sequence. An automatic support does not take place.

In the start date field, it is possible to implement an execution with periods of less than one day via “Periodic scheduling.” These are described in **BatchMan** as periodic order chains lasting less than one day. Since this function does not only run smoothly, INDIV orders are no longer scheduled periodically. For this, order types STD and WDH are to be used:

WDH – REPEAT ORDER

WDH Orders are obsolete, please don’t use them anymore!

The repeat order represents the basis for repeating periods, in which the period is less than one day (periodic order chains lasting less than one day). The **BatchMan** master job then generates from these WDH orders the corresponding AUTO orders.

This order type is for the function of the periodic orders lasting less than one day that is also supported in the future since more functions can be mapped with it.

The WDH orders are divided into four types:

- **WDHTI – time-periodic WDH orders**

  For the order, the periods of time “from – to” or individual times are maintained.

- **WDHFI – file-periodic WDH orders**

  The processing starts when a file is there.

- **WDHEV – event-periodic WDH orders**

  The processing starts when the event is triggered. The event is scheduled in SAP as a job.

- **WDHEV – event-periodic WDH orders**

  The processing starts when the event is triggered. The event is read from the SAP event history (XBP 3.0 required).

AUTO - AUTOMATICALLY GENERATED ORDER

The “automatic” order type cannot be created manually. Orders of this type are automatically generated by the system. Upon the generation of automatic orders, the objects included in the WDH order are automatically read from the library. Changes to objects in the master data are taken into account upon creation of a new AUTO order.

Order extention

Selection of the extentions selected in section 3.2 for the order types. The system only displays the extentions maintained for the order type

Scheduling day

The scheduling day of an order is the day on which the scheduling of the jobs and job networks in the operator order are to be carried out.

For the "STD standard" order type, the scheduling day is used as selection criteria to generate a proposal list with jobs and job networks to be scheduled.

For “WDH orders,” the date is set fixed to December 31, 9999.

Sequential No.

For the unique identification of an operator order, the system assigns a sequential number within the order type and scheduling day.

Order number

The order - order number consists of

- Order type

- Scheduling day

- Sequential number

The order number is automatically inserted into this field by the system.

Background of the number assignment for periodic orders

For all orders, an internal procedure exists for the assignment of numbers. A sequential number is created for the individual order types per day. This procedure guarantees that all orders can be traced back individually in the timing sequence and archived.

In WDH orders, the date(internal) is set to December 31, 9999.

All periodical orders carry along the basis order from which the periodic orders are generated.

Based on

In this field, the system displays the order number of the copy template for the current order.

Order status

The following status types exist:

- in planning
- checked
- released
- active
- faulty
- aborted
- cancelled

Text

Here, an explanatory text may be created for the order.

Consider scheduling dates

Mark if the system is also to consider the scheduling dates for the order types INDIV in WDH*.

Periodic

This indicator is used for periodic orders that are subject to automatic repeat scheduling and are to start up hourly, daily or weekly, for instance.

(it compare section 3.3.3 Automatic Rescheduling of Orders).

Refresh AUTO order

For the creation of an AUTO order, this indicator shows that changes to the job and network master data in the library are to be taken into account in the order. The flag is set automatically if the order is periodic.

Created

This specifies when (date and time) and who entered the order (order number assignment).

Scheduling

This specifies when (date and time) and who scheduled this order. In contrast to the Entry field, this field specifies when the order was saved.

Check

This specifies when (date and time) and who checked this order.

Release

This specifies when (date and time) and who released this order. When an order is released, it may then be executed. The jobs contained in the order are executed.

Change

This specifies when (date and time) and who last changed this order (e.g. stop for order).

3.4. Create/Maintain Order

You have to enter an order type + extention and the scheduling day in order to create a new order. These specifications plus the ‘sequential number’ automatically assigned by the system will generate an order number:

For the order type, INDIV, WDH* or STD must be selected at the time of creation. STD should only be selected if the standard order of the day was not processed correctly. AUTO-orders are automatically created by the system.

3.4.1. Order type STD

For a **BatchMan** system set up completely, the **BatchMan** master job automatically creates this order type. Here, all jobs and networks are automatically placed into the order whose master data meet these prerequisites:

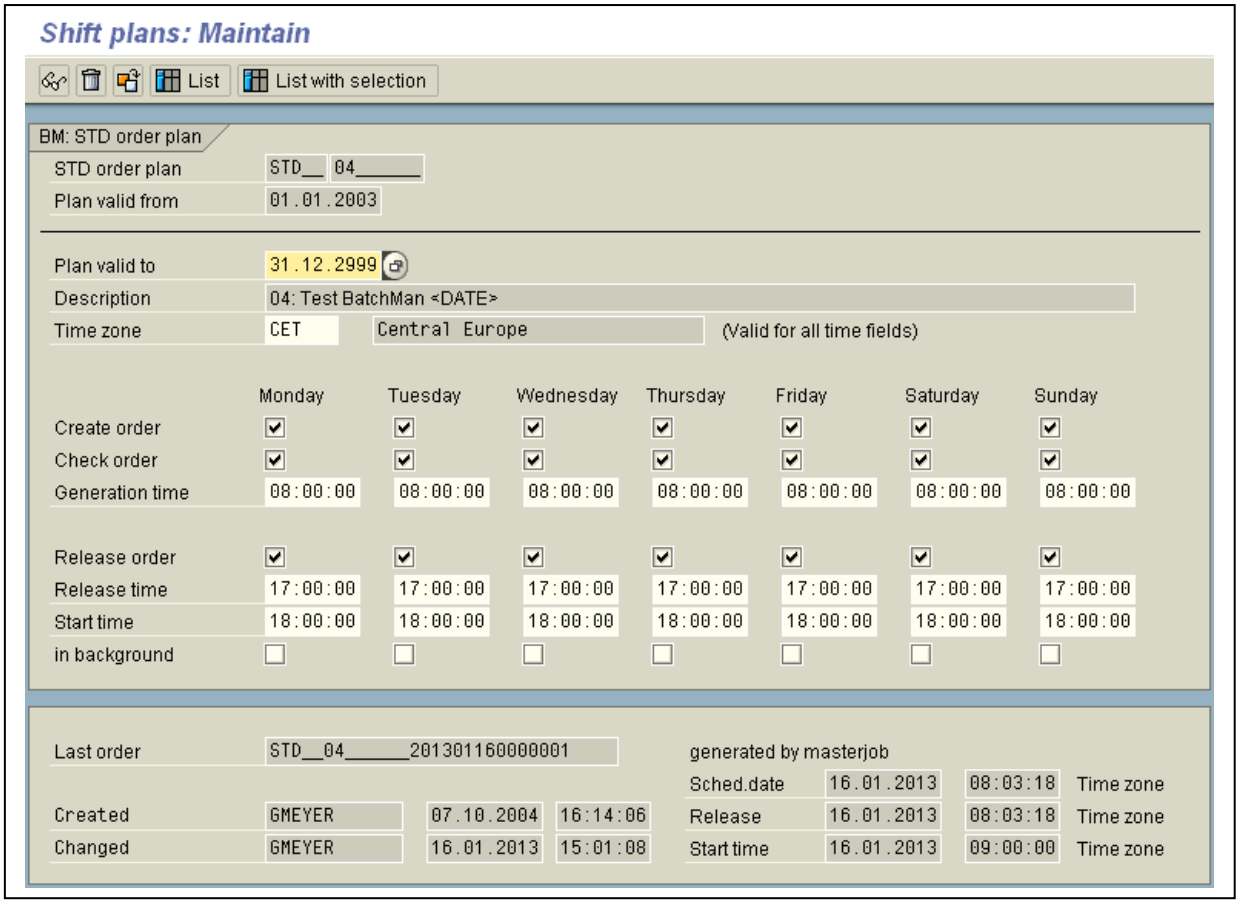
a) the field "Automatic scheduling" was marked and

b) the scheduling date corresponds with the scheduling day of the operator order

c) the corresponding plan is maintained

d) the scheduling day is within the validity period

In addition, the shift plan must be maintained in customizing (see STD Plans/Shift plans) for the master data objects (networks and jobs) so that the master job can recognize when (day and time) specific actions should be triggered.



If the "**in background**" flag was activated, the system executes the action as a job in the background, otherwise directly in the master job.

Per weekday, the shift plan should indicate if and when (**generation time**) the STD should be **created** and if it has to be **checked**.

In Batchman prior to version 5.2, the entire content of the order was checked when STD orders were generated. This had two disadvantages. The check could take a very long time and then delayed the order release or could be canceled, in which case there was no order release at all.

As of Batchman 5.2, content validation is now asynchronous and no longer affects job release.

If the content check produces errors, an alarm is raised (as it was the case before BM 5.2).

However, this alert may then be after the order has been released and will no longer impede it.

The second part determines if or when (**release time**) the STD should be **released**, and also at which time the STD should start (**start time**).

All time specifications correspond to the entered time zone.

This makes it possible to automatically generate a STD order (in this case, e.g. at 8 a.m.) with all jobs and networks. Between 8:00 a.m. and 17:00, manual changes (adding jobs/networks,

remove jobs/networks, changes in the process flow/ moving of jobs/networks) can be done. At the time of release, the STD order (in this case, e.g. at 17:00) will be released automatically.

Significance of the setup parameters relevant for the STD order

Leave out jobs in networks with different date: If this parameter is activated (**X**), the system takes into account the period dates of the jobs within a network for the release of the STD order. The jobs are emptied/skipped (color of the jobs turns PINK) and then executed.

If the field is not activated (**empty**), the system ignores the period dates of the jobs within a network. This means that only the periodicity of the network is valid. The system executes the jobs with all defined steps.

Example:

Network 1 should run from Monday until Friday (automatic scheduling) and contains three jobs A, B, and C

Job A should also run daily.

Job B should run on Monday, Wednesday and Friday.

Job C should run on Tuesday and Wednesday.

In all three jobs, the period values are maintained correspondingly.

X – aktiv the network is executed daily from Monday until Friday. Job A is also executed daily with all defined steps. Job B, according to definition, is only executed Monday, Wednesday and Friday with all defined steps. On Tuesday and Thursday, job B is executed empty. Job C is being processed similar.

|_| -nicht aktiv Network 1 is executed daily from Monday until Friday and all three jobs as well, because the period values of the jobs are not valid in the network.

Examples for objects, which can be (and should be) processed in a STD order:

FI Network	Daily	Open item list
	Every fortnight	Dunning run
	Weekly	Payment run
MM Network	Monthly	Period closing
	Daily	Billing
BC Network	Daily	Reorg Jobs
	Weekly	Reorg Jobs

Examples of objects, which can not be placed in the STD :

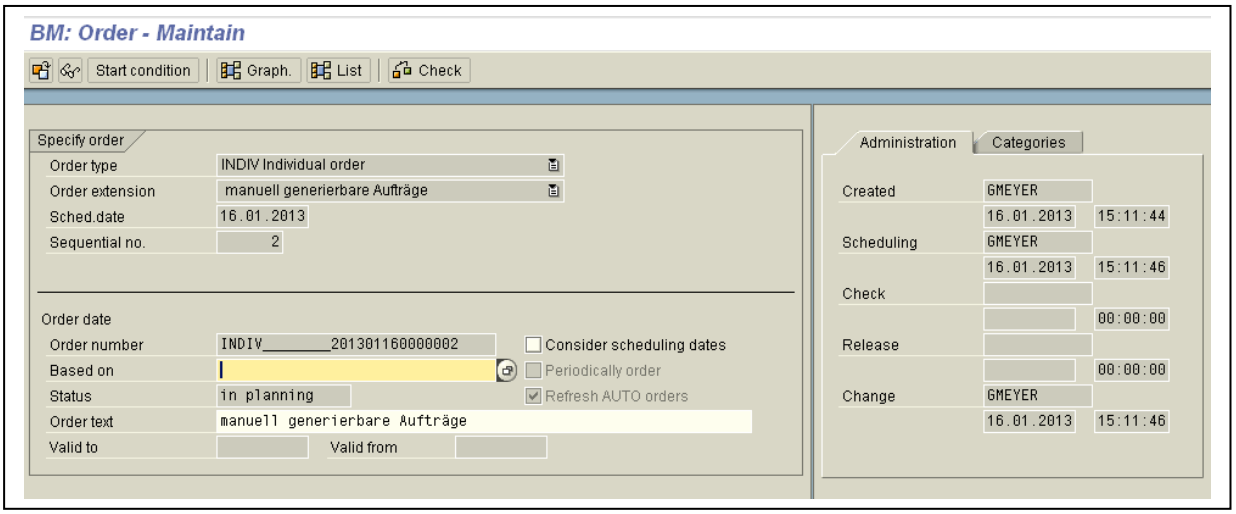
Objekt zwischen 03:00 und 21 Start object every 15 minutes

Execute object between 03:00 and 21:00 every hour

In general, all objects with a period longer/equal 24 hours should run in the STD, and all with a period shorter than 24 hours should not run in the STD.

3.4.2. Order type INDIV

Select the order type INDIV from the list box and click on the button **Create**. The order is pre-entered and must therefore be saved.



The status of the order is ‚in planning'. Order positions (jobs and networks) and start date can be maintained.

The functions can be called via the buttons or the menu functions.

Start condition	Maintain start date
Positions	Maintain positions of the order (order contents)
Check	Check order
Release	Release order
Order	Delete order
	Temporary STOP of order; active jobs continue running
	START after stop; order continues after order stop

After maintaining the order positions and the start date, the order can be **checked**. The status change to ‚checked' (if nothing is faulty).

After switching to the display mode, the order can be released. The status is set to ‚released'.

In this case, the system also updates the responsibilities.

When the order is processed, the status is ‚active' as long as no jobs are aborted. If this should happen, then the status changes to ‚faulty'.

The order status after finishing all jobs is ‚finished', or ‚aborted' (if jobs in the order were aborted).

3.4.2.1. Periodic scheduling of INDIV orders

Periodic INDIV orders are obsolete, please don't use them anymore!

In order to let the system generate orders for the automatic re-scheduling, you have to first create an order of the type INDIV.

After pressing the button **Start conditions** you have to mark the check box 'Periodic job' in the appearing pop-up.

After performing the functions **Check** and **Release**, this order is available for processing, so that it can start at the specified start date. After the order is finished, a new order of the type "AUTO" is generated based on the already processed order, which is now called 'basic order". The new order receives its own order number.

This newly generated order has a future start date which is calculated based on the periodicity specified in the basic order.

This periodic scheduling, starting, and new-creating with an individual order number in each case continues to run until the periodicity of the basic order is finished. The basic order in an auto order is always deposited in the field **based on**. Via double click on this field you can switch to the basic order.

3.4.2.2. Finishing the Periodicity of a Basic Order

To end the periodicity of the basic order, take the following steps:

- Select the periodic basic order.
- You can make a first selection by entering the order type **INDIV** or marking check box **Periodical order**.
- After pressing the button **Select**, a list is displayed which contains the selected orders.
- The wanted order is selected via a double click.
- After pressing the button **Edit,** a pop-up appears which asks if the periodic scheduling should be cancelled.
- The buttons **No** and **Abort** have no effect on the periodicity.
- If you press the button **Yes,** the periodic scheduling is cancelled.
- In addition, the system asks concerning the orders which are dependent on the basic order and have the status „Released", whether these orders should be cancelled and therefore receive the status „Cancelled".

3.4.3. Order Type WDH*

WDH orders are obsolete, please don't use them anymore!

(Periodic orders with periods with less than 24 hours → orders lasting less than one day)

The logic of the previous scheduling for periodic order chains lasting less than one day:

- Create INDIV order, enter objects (jobs/networks) (perhaps maintain start criteria of the objects)
- Maintain start date (with restrictions, no scheduling on non-working days)
- Determine period
- Check and release order
- After the end of the INDIVs, the system generates an AUTO order, which then generates the next AUTO order, and so on.

That old logic is now replaced with this new logic:

- Create WDH order, enter objects (jobs/networks)
- Maintain start dates
    - Execution “on all days,” “only on working days,” “only on non-working days;” determine periods
    - Determine calendar as well as the individual weekdays; Determine whether parallel processing is desired
- Check and release order
- From this order, the required AUTO orders are generated by the master job. The AUTO orders are generated for the future. The time period is determined by the setup parameter: **“Time frame (LOOK AHEAD) for AUTO orders**.”

WDH orders are not executed. The AUTO orders scheduled by the master job are the orders executed.

This logic eliminates error sources of the old logic:

- The archiving was not allowed to archive any INDIVs that were the basis for periodic chains. In practice, these basic operator orders were often deleted.
- After a system shutdown and tablespace full, the periodic order chains did not automatically start in all situations.
- Different periods within a chain were not possible.
- The restrictions “no scheduling on non-working days” did not always work smoothly.

Due to these errors, only the WDH* / AUTO logic will be supported in the long term. Starting with version BatchMan 3.4.1, there is a check at the end of an AUTO order, if the period is smaller than the value of the setup parameter (Time period ...). If yes, the next AUTO order is generated immediately. If this is not possible, then the master job will undertake the generating.

With this method you can realize shorter periods (up to one minute) for time periodic WDH orders than the period of the master job. The period of the master job can then be between three to five minutes.

Tasks when scheduling WDH-/AUTO-orders, which the BatchMan-Masterjob executes, as well as relevant setup-parameter

For each run, the master job reads the table of the WDH orders and calculates for the individual WDH orders when the next scheduling has to take place.

If this time is in the current time, plus the setup parameter “**Time frame (LOOK AHEAD) for AUTO orders,**” so then an AUTO order is scheduled. If the period is smaller than the time frame of the setup parameter, then the system schedules several AUTO orders. If the **Parallel execution** is permitted, then the period can also be smaller than the master job.

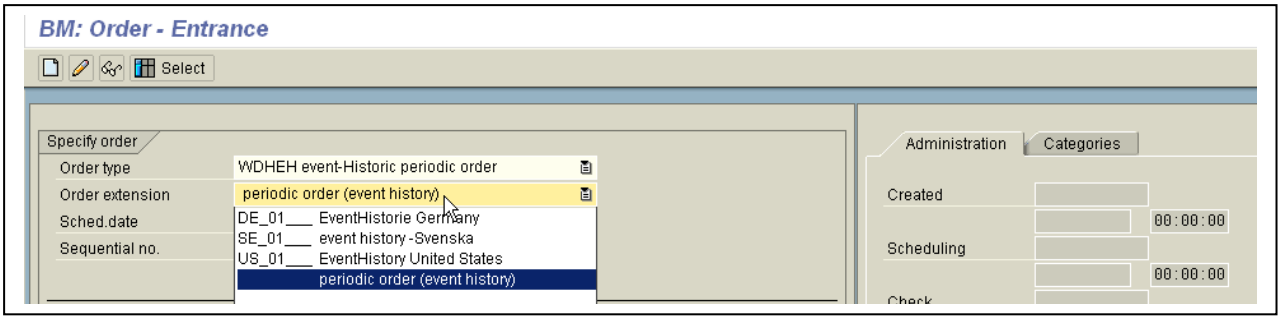
If the **parallel** execution is not activated, the release of the next AUTO order is always delayed until the end of the current AUTO order. Here the smallest period is determined by the period of the master job.

From BatchMan 4.3, the WDH orders are provided as independent WDH order types with the option of extension.

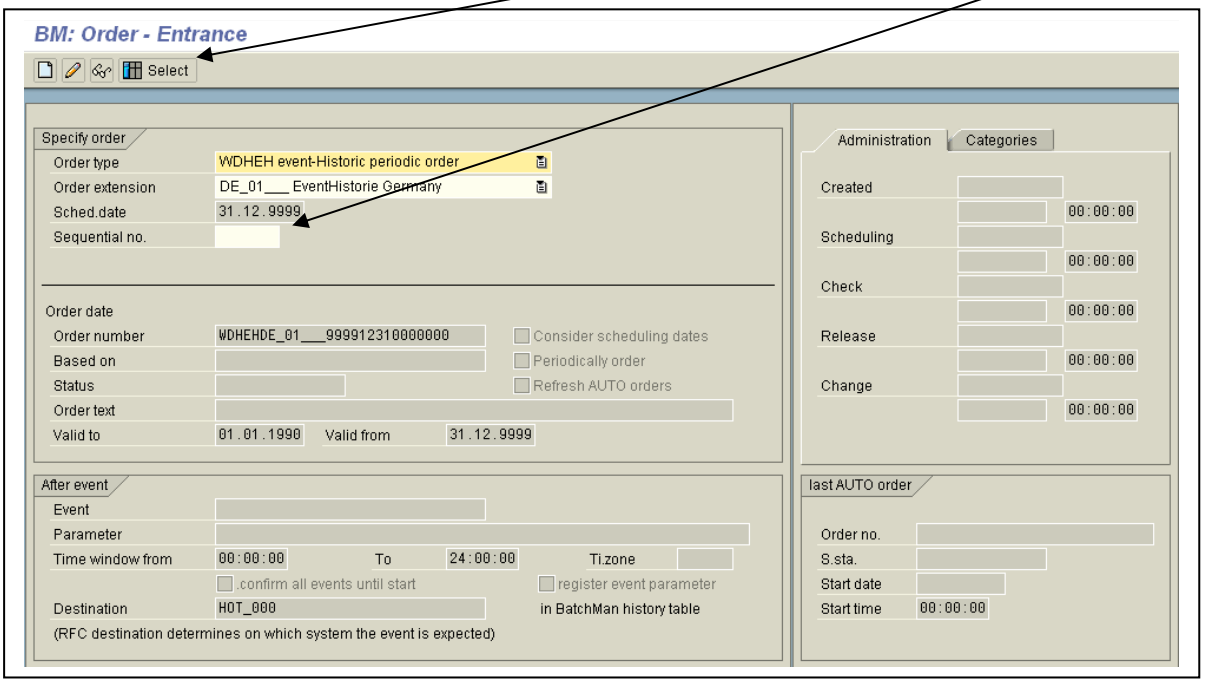
Create: Order Administration (Transaction J5HD)

Please execute the following steps to create WDH order:

- Select order type and extention from the list boxes
- Create



- The next to-be-assigned number is traced and the WDH is created with this number.
- If the WDH needs to be changed, you can either select the option or enter the number directly

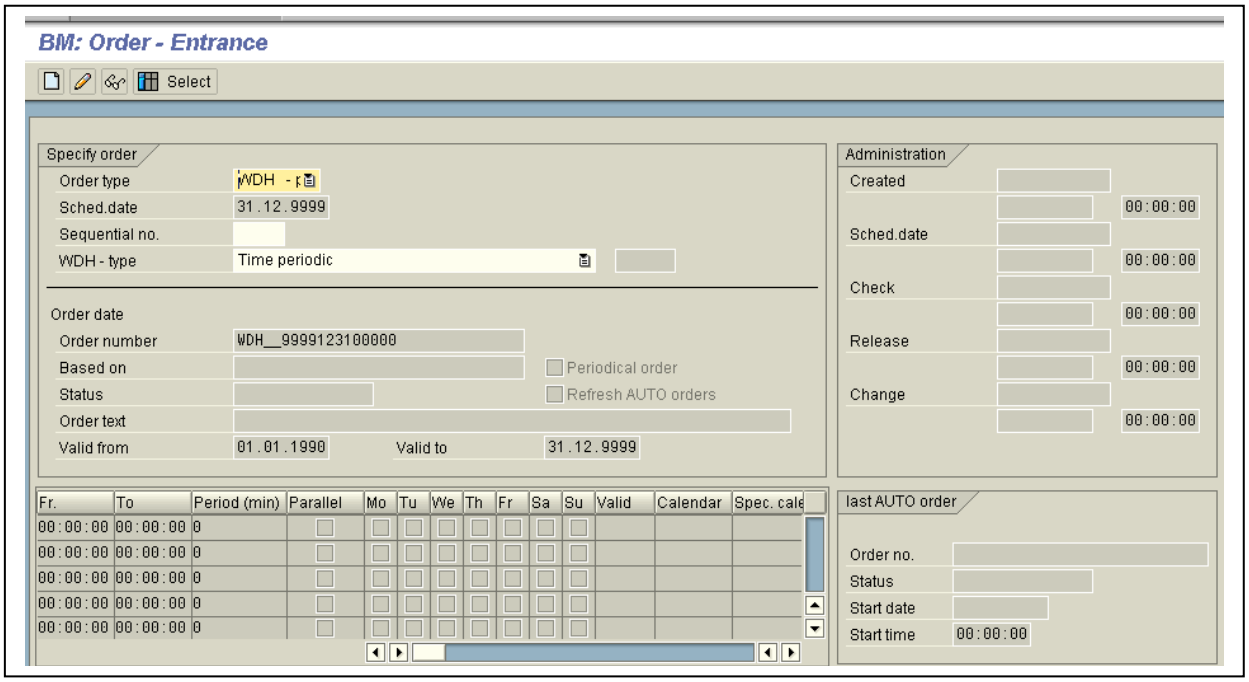


- After Creation, the start conditions become visible.
- Via positions, the objects (jobs/networks) as previously entered and the links of the objects are determined amongst each other.
- The scheduling day is generally set to December 31, 9999 within the program for WDH orders.
- Via the fields **'Valid from'** and '**Valid to'**, you can specify from what time on or until when a WDH should be active. These fields can also be changed in the display mode via the button '**Validity'**.
- Additionally, the last generated AUTO (belonging to the WDH) order is displayed below on the right.

After the release, generating of the AUTO orders is called up at once in the dark area. If the start conditions are fulfilled, then an AUTO order is created.

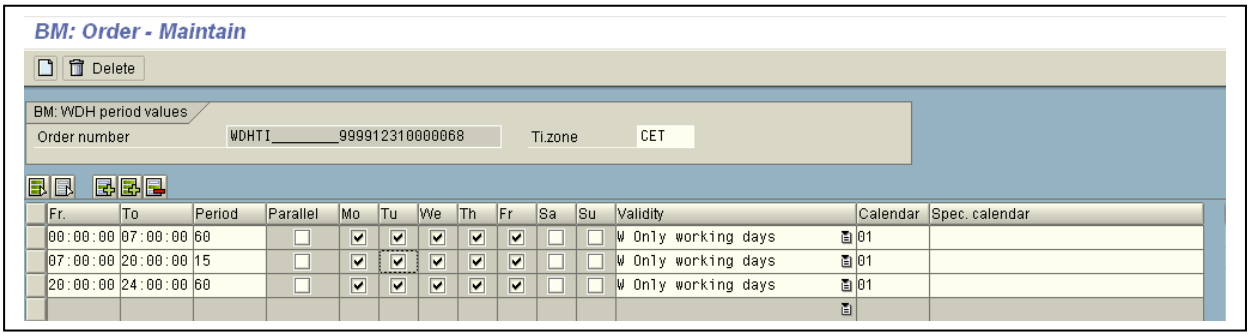
3.4.3.1. Time-Periodic Orders - WDHTI

WDH orders are obsolete, please don't use them anymore!



The display of the period values on the screen of the order administration is mainly for the purpose of clarity. The period values are maintained via the **Start date** pushbutton or via double-clicking on the table.

Screen **PERIOD VALUES** for WDH-orders



	Insert/add single lines
	Insert/add several lines
	Delete marked lines
	Mark all lines
	Un-highlight/Remove marking of all lines

New lines are always at "Period 1 Minute" so that the "To" field is ready for input.

If the **Time zone** is filled, all times entered will refer to this time zone. If the time zone is empty, the system enters the system time zone while saving.

Valid from: Begin of the period

Valid to: End of the period

Period in minutes: Specification of the period value. Specifies after how many minutes the next AUTO order is generated from this WDH-period.

Parallel allowed: Specifies if a new AUTO order can be generated, before the previous AUTO order is finished.

Weekdays: Days on which the AUTO orders should be generated.

Validity of the period: Specifies the days on which this period is valid:

- **SPACE/ blank** is always valid, that means for all days (working days and non-working days)
- **N** Valid only on non-working days
- **W** Valid only on working days

Since the values are provided via a list box, you can see the verbal description and not the values of the validity.

If ‚working days' or ‚non-working days' are selected in the validity, then the specification of an SAP factory calendar or a **BatchMan** special calendar becomes necessary.

Calendar: SAP factory calendar which specifies if a day is a working day or not.

Special calendar: **BatchMan** special calendar which specifies if a day is a working day or not.

If two periods are created which touch each other (e.g. valid to period 1 = 14:00 hours and valid from period 2 = 14:00 hours), only one AUTO-order is created.

The periods are recalculated each day at midnight; i.e. periods of for instance five hours do not run. (On the second day, the order starts again at midnight (period four hours) and at 5:00 a.m. and not at 1:00 a.m.)

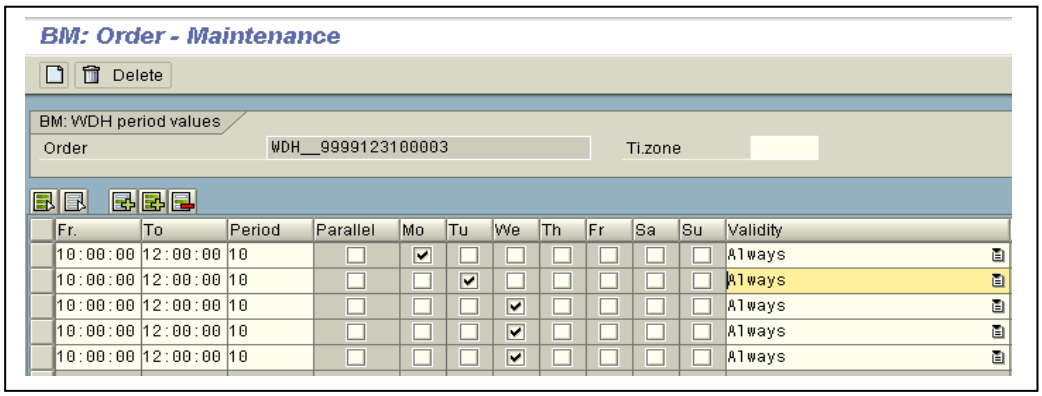
The following overlaps of periods are not possible:

→ It is not possible to create a period with valid=empty/blank and 'all days', and a second one with valid=non-working days and day=Monday.

→ Within one day only periods are allowed that maximally ‚touch' each other. For example, period 1 ends at 06:00 hours; period 2 begins at 06:00.

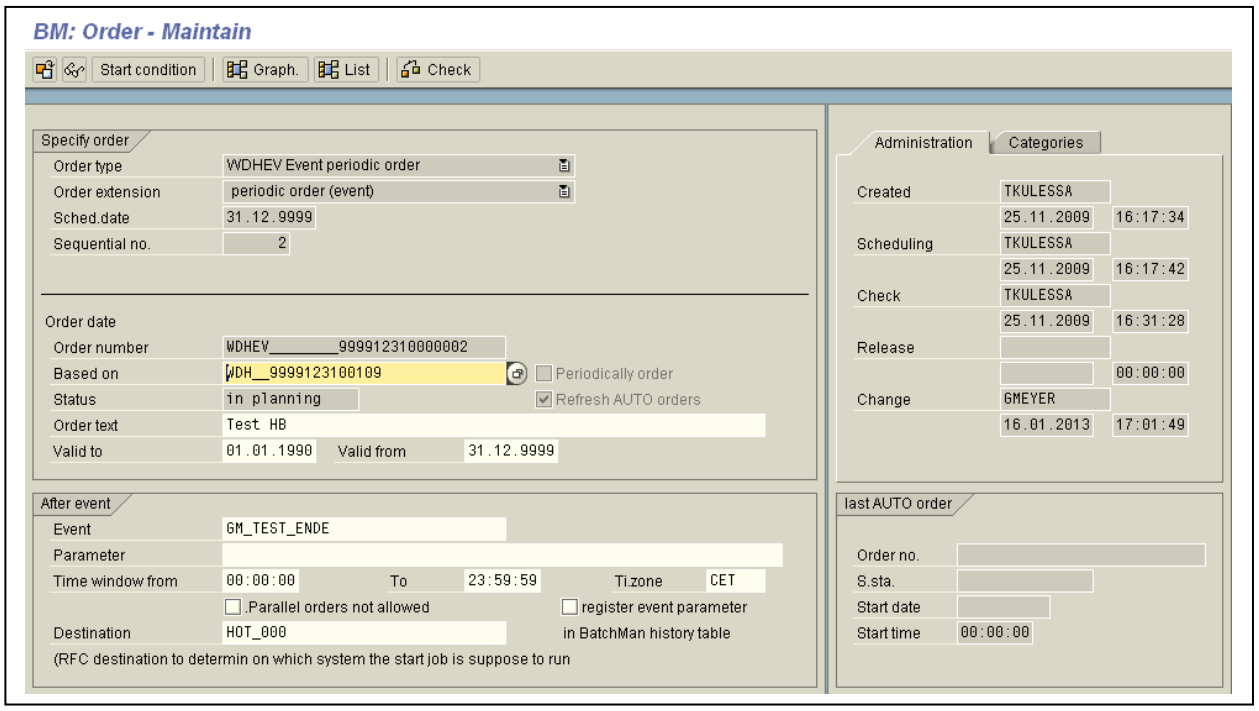
After period value creation, the order maintenance can appears as shown (e.g.)

After saving, the period times are also shown in the screen of the order administration.



3.4.3.2. Event-Periodic Orders – WDHEV

WDH orders are obsolete, please don't use them anymore!



In the screen of the order administration, the **EVENT** and the **PARAMETER** (case sensitive) are entered for the start of the order.

The system always releases an AUTO order ahead, which waits for the event to be triggered. The start job of this AUTO order is physically scheduled as job (order number plus "OA" at the end of the job name). The scheduling is made on the target system that is defined by the **destination**. The event is then also to be defined on the target system.

The master job reads per individual run, if a new order should be created and released for this chain. In the time period from triggering the event until the next run of the master job, it is not possible to process an identical event.

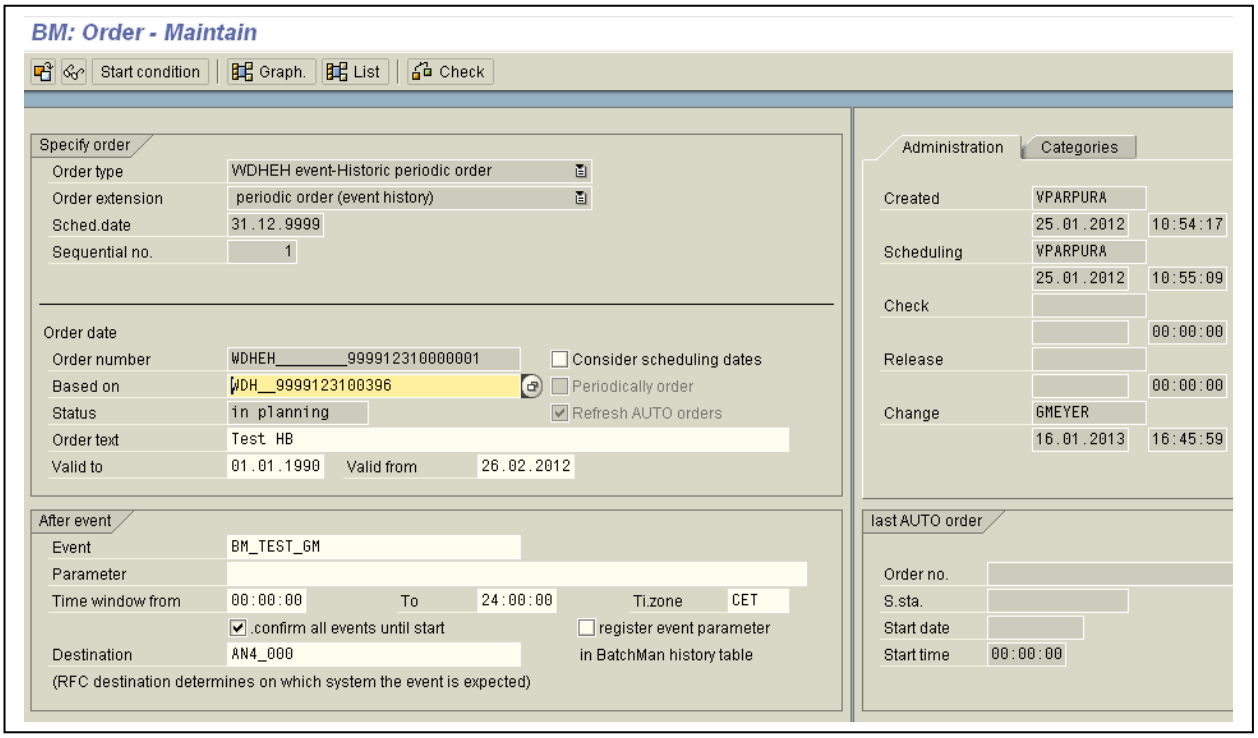
If the flag ‚**Parallel orders not allowed'** is activated, a new AUTO order is not created immediately after receiving the event and start of the order. This is only done after the processing end of the order.

The **destination** defines on which system the job, which waits for the event, is scheduled. This need not be the current system. If the event is triggered in a client system, the client system must be entered here as the destination.

When you set the flag “Register event parameter in BatchMan history table", the system saves the event-triggering parameter in the BM event history, and it can be transferred to a job in the processing sequence of the jobs.

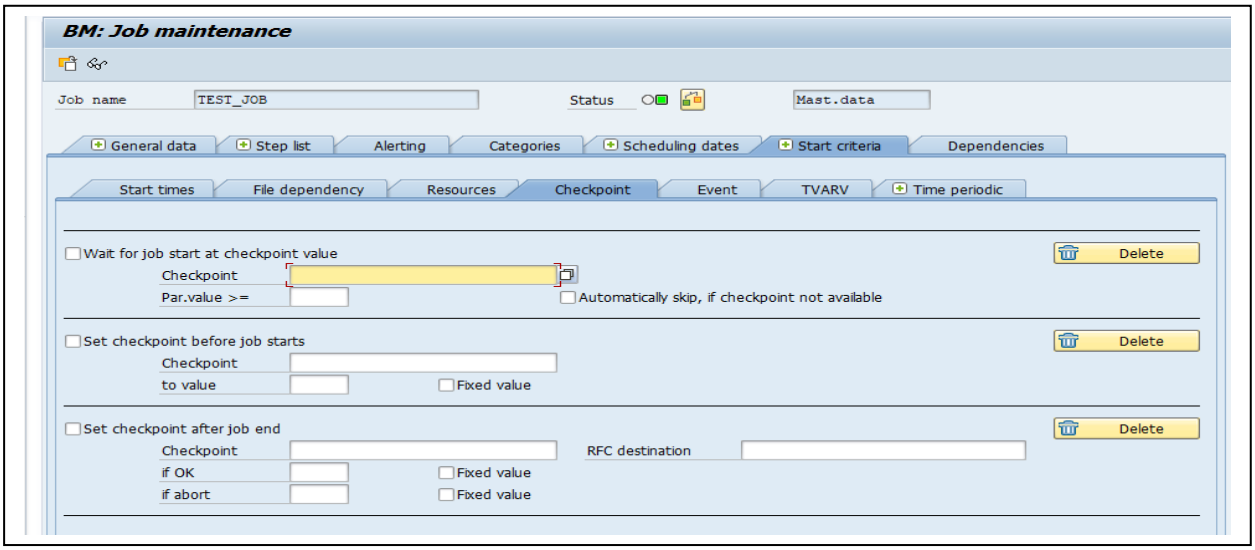
3.4.3.3. Event-Historic Periodic Orders – WDHEH

WDH orders are obsolete, please don't use them anymore!



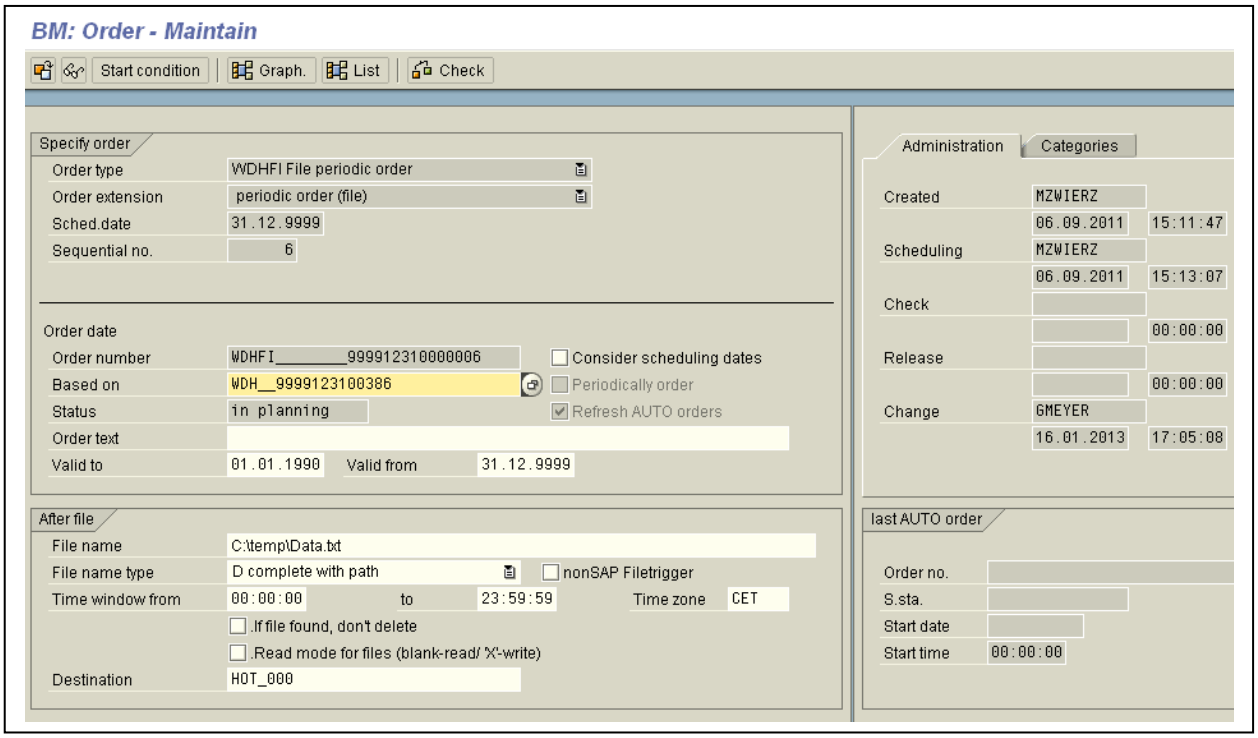
From XBP release 3.0 you can use the so-called event history. Use transaction SM64 or CRIT to maintain the event history.

In this tranaction you maintain the criteria, on which the SAP system respond. This is on the target system, defined through the RFC destination.



3.4.3.4. File-Periodic Orders -WDHFI

WDH orders are obsolete, please don't use them anymore!



In the order administration, the file is specified which starts the order when that file is present. The master job reads per run whether for this chain a new order is to be created, released and started. That means, no order is defined ahead of time. If the system finds a file, it tries to open this file to READ. In this case, the SIDADM must have write authorization for the file system.

Normally, the file that triggered this order is immediately deleted after detection by the master job. You can prevent this with the "Don't delete if file found" flag.

Via the **Read mode** of a file you can determine how the file is read:

Blank - Reading of the file in shared mode (= no exclusive reading)

‚X' - Reading in append mode (= exclusive reading)

Three types can be used as file name:

- Direct entry of the file with path, name and extension – Type ‚**D**'.
- Supply of the file name via a TVARV variable - type "**T"**. This makes is possible to process a file with the current date, for example. Created like TVARV variables supplied and updated with rules; more information is in the chapter on TVARV. The TVARV variable is read locally on the master system.
- Supply of the file name via a logical file name - type "L**"**. The determination of the logical file name occurs with the following destination, so that the variable parts of the logical file name are converted/ substituted according to the system on which the file is stored.

The **destination** defines for which system the file is expected. This need not be the current system. If the file is stored in a client system, the client system must be entered here as the destination.

3.4.3.6. Interrupting/Stopping the Periodicity of WDH (Repeat) Orders

Repeat orders can be discontinued in different ways.

1. Editing a released order (Status ‚released').

   If a released order is edited, the status changes to ‚in planning'. In this case an inquiry follows about what should happen with the perhaps still released AUTO-orders. You can select whether to cancel them or not.

   After further checking and releasing, the periodic chain starts to run again.

2. Setting of the STOP-Flag: By pressing ‚lock closed', the order is stopped. No further AUTO-orders are created.

3.4.4. Order Type AUTO (Automatic Re-Scheduling of Orders)

Should an order be executed periodically (hourly, every 15 minutes, etc.), it is practical to use the „automatic re-scheduling" of orders (see previous sections).

The special order type *AUTO* meant for such orders can not be entered directly by the user - it is automatically generated by the system.

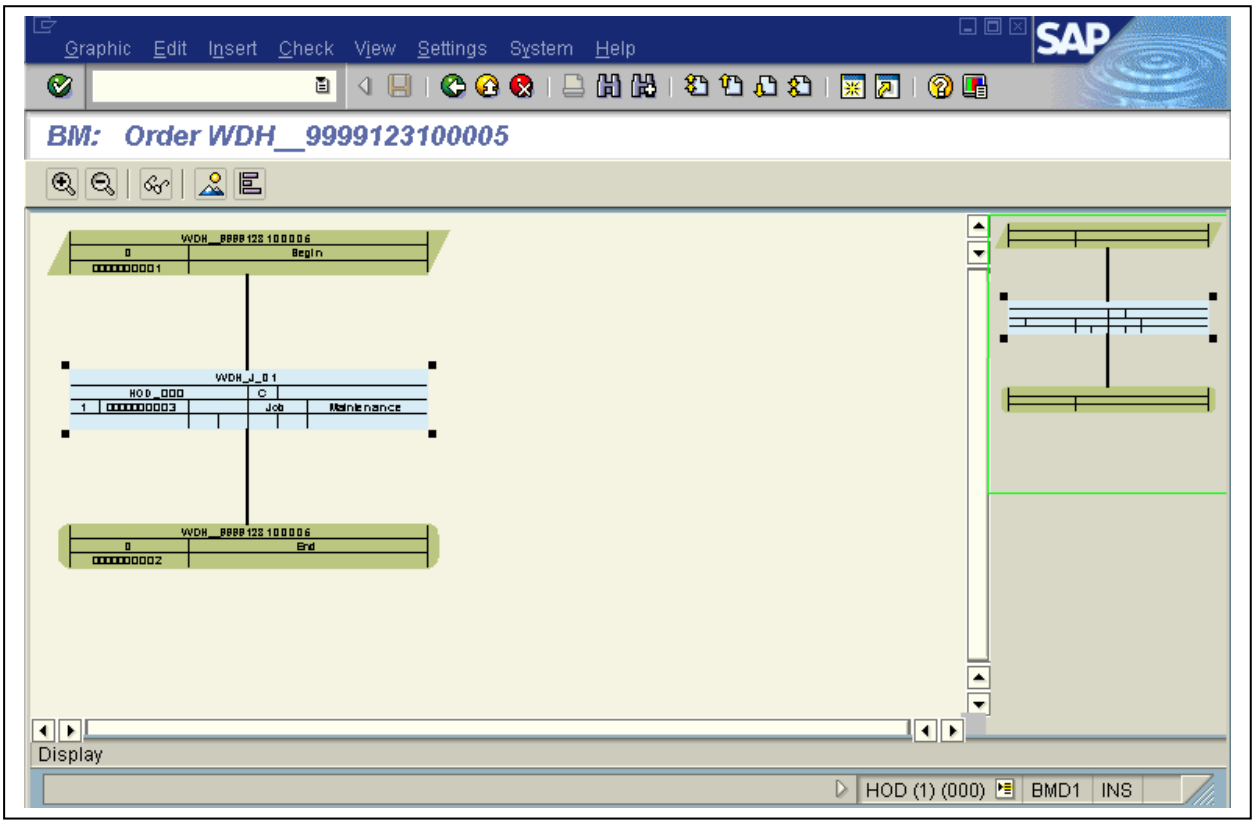
3.5. Network Plan Graphics for the Orders

In the network plan graphics, the objects (jobs and networks) for the orders are inserted; and interdependencies between the objects are maintained.

The network plan graphics is called via the button **Positions.**

When an order is called for the first time, are two nodes have already been created; the start node (node number 1) and the end node (node number 2). All other nodes are positioned and linked between these two nodes.

This function behaves like already described in the network maintenance (see Network Plan Graphics of a Network). Only the beginning node has a different form because less information items can be included at the order beginning.



3.6. Start Date Operator Order

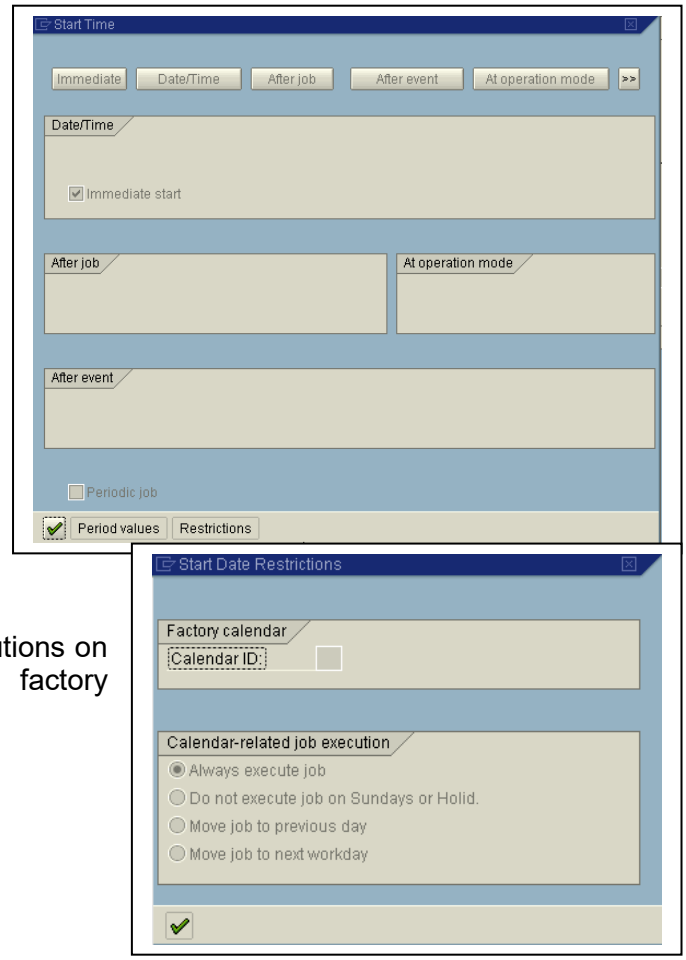
The following only applies to INDIV and manually created STD orders.

For the “Check and release an operator order” function, a start date must have been entered.

- Call via the button **Start conditions:**
- Maintain the start date according to the specification
  - Immediately
  - Date / Time
  - After job (job must already exist in SAP)
  - After event
  - At operation mode
  - On certain workdays

Periodic executions are possible but should not be used here in BatchMan.

- Then maintain **period values**



Via ‚Restrictions’ one can maintain the executions on working and non-working days via SAP factory calendars.

3.7. Checking and Releasing an Order

Only after an order has been released may the objects in this order be executed. Since networks are only a compounding of job processes, only the jobs are executed as processes.

BatchMan offers a two-step release procedure. The first step is the **checking** of the order; the second step is the **releasing**. This way a two-step release procedure can be realized. This has lost importance because of the automatic creation of STD and AUTO orders - these functions are (mostly) executed in one step by a background user (or also a batch-/system user).

With the function **checking,** a complete consistency check is done to determine if all objects of the order are error-free. This constitutes the main handling efforts of the object checks (contrary to earlier BatchMan - Releases: there the main work was done during the release procedures).

With the function **release** the status of the order is set, the order may start. Possibly order beginning jobs are scheduled in SAP (start conditions EREIGNIS/ EVENT).

3.8. Menu Functions

3.8.1. Edit → Stop / Start

With this function, orders with the "***active"*** or "***faulty"*** status can be stopped. They retain their status. Active processes continue to run, but the system no longer starts a successor.

Via **Start,** you can re-release the processing. After the next master job, the processes run again.

3.8.2. Edit → Cancel order

The order is cancelled so that released orders no longer start. This function is used mainly within **BatchMan** to cancel those orders already created and released when editing WDH orders. The status is displayed as "**cancelled**." After cancellation, orders can no longer be edited.

3.8.3. Edit → Finish orders

This function can finish orders with the status ,***active'*** or ***,faulty'.*** They then receive the status ,***finished'*** or ,***aborted'***. This function is used in inconsistent orders, which can be changed to a defined final state this way.

Orders can not be edited anymore after this process has been finished.

3.8.4. Edit → Transport object

This function offers the option to add orders of the type WDH to a transport order and transport them to a different system. However, this only makes sense if the WDH order does not exist yet in the other system and therefore has to be created new (in the other system).

If the WDH order already exists in the target system, then it will be completely overwritten.

3.8.5. Goto→ Job monitor

Call the **BatchMan** job monitor. It offers similar functions like the transaction SM37. However, the jobs are read via XBP interface, so that jobs from remote SAP systems can also be displayed.

3.8.6. Goto → Application log

Call the function application log (transaction SLG1).

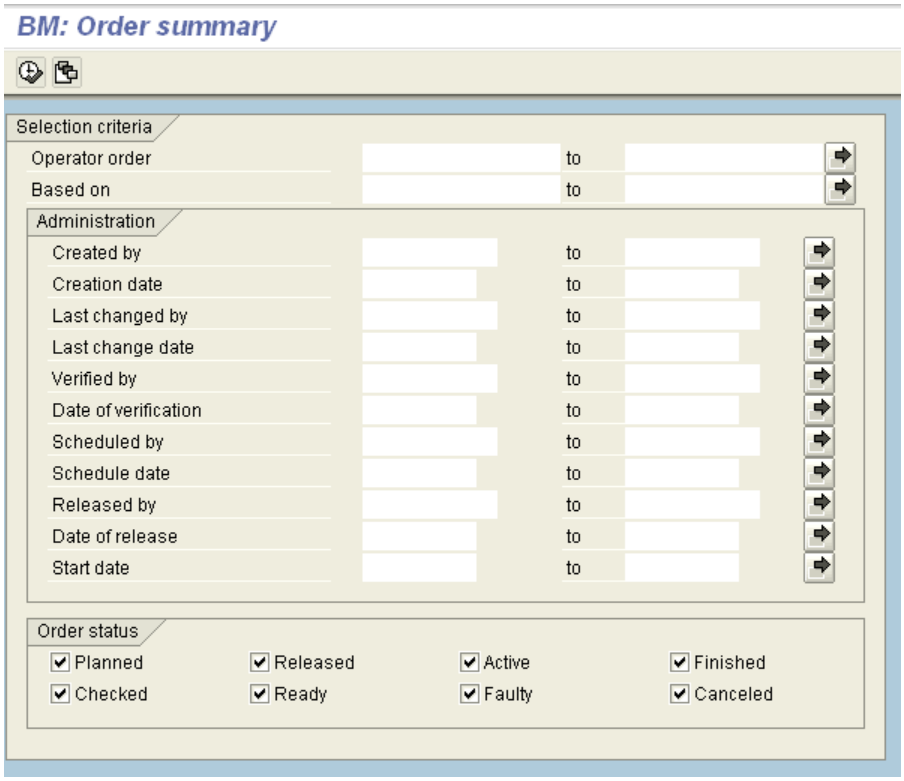
3.8.7. Goto → System log

Call the function System log (transaction SM21).

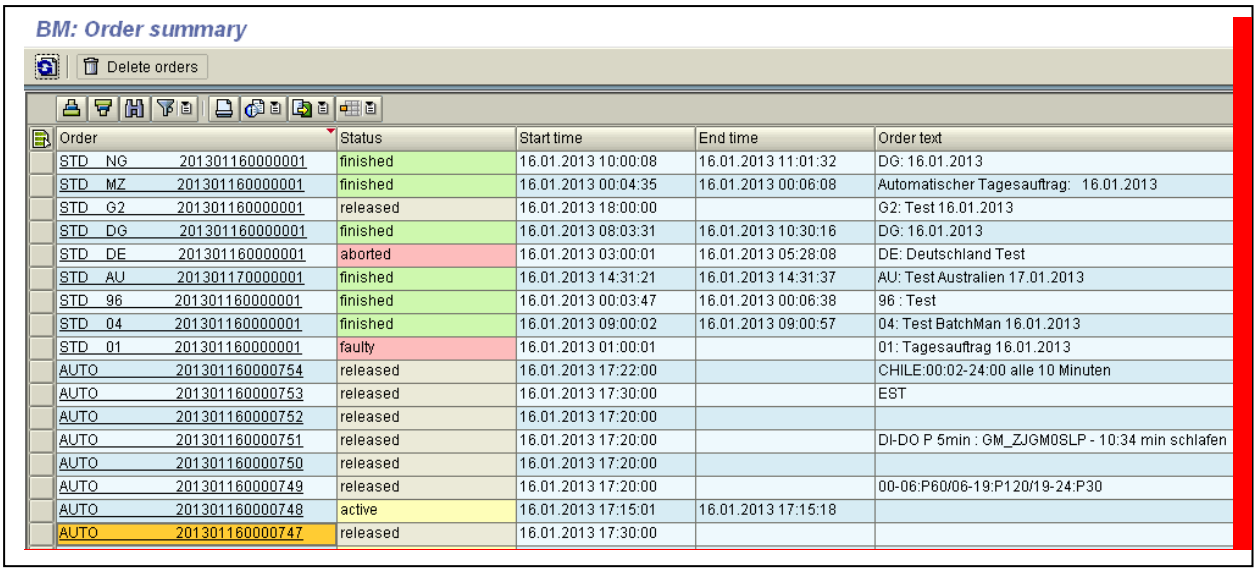
3.9. Reporting about Orders

3.9.1. Order overview

This program delivers an overview of selected orders. Since a large amount of orders exist over time, you should make a SELECTION here. If the selection is not restricted, the program may need to run for longer time (until a TIME OUT).



Results list:

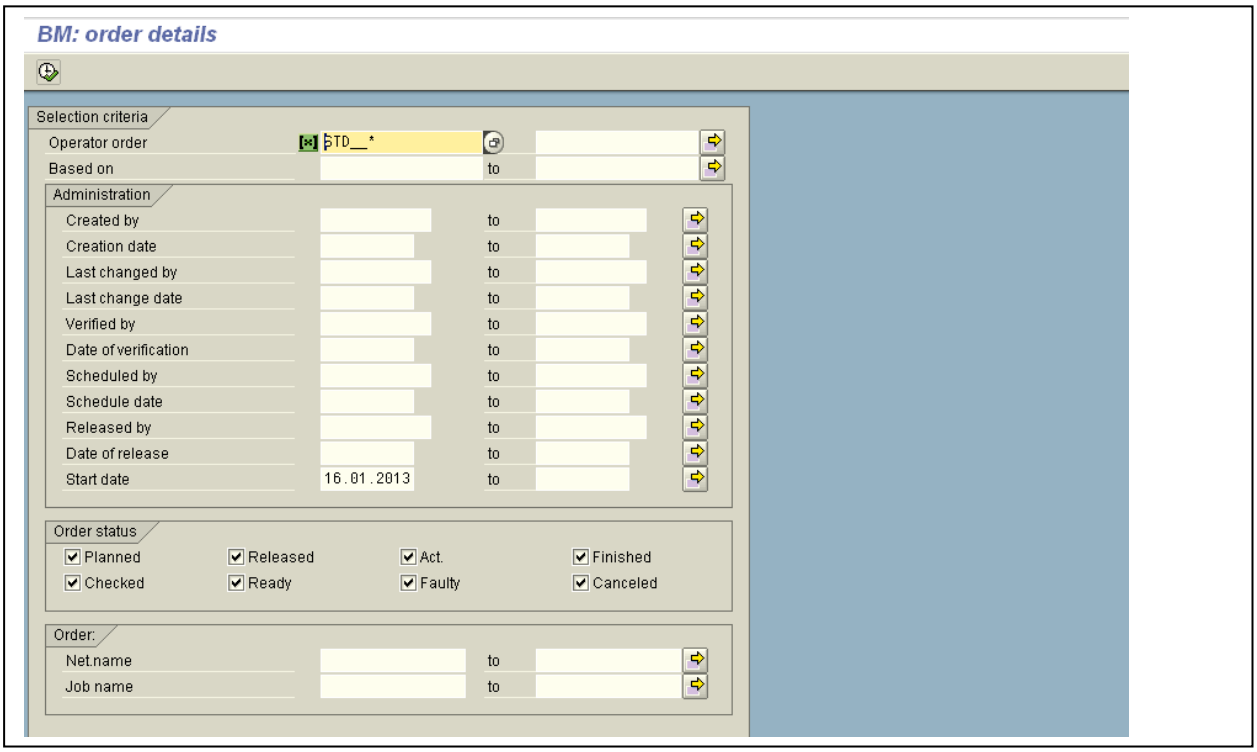


Order	Status	Start time	End time	Order text
STD NG 201301160000001	finished	16.01.2013 10:00:08	16.01.2013 11:01:32	DG: 16.01.2013
STD MZ 201301160000001	finished	16.01.2013 00:04:35	16.01.2013 00:06:08	Automatischer Tagesauftrag: 16.01.2013
STD G2 201301160000001	released	16.01.2013 18:00:00		G2: Test 16.01.2013
STD DG 201301160000001	finished	16.01.2013 08:03:31	16.01.2013 10:30:16	DG: 16.01.2013
STD DE 201301160000001	aborted	16.01.2013 03:00:01	16.01.2013 05:28:08	DE: Deutschland Test
STD AU 201301170000001	finished	16.01.2013 14:31:21	16.01.2013 14:31:37	AU: Test Australien 17.01.2013
STD 96 201301160000001	finished	16.01.2013 00:03:47	16.01.2013 00:06:38	96 : Test
STD 04 201301160000001	finished	16.01.2013 09:00:02	16.01.2013 09:00:57	04: Test BatchMan 16.01.2013
STD 01 201301160000001	faulty	16.01.2013 01:00:01		01: Tagesauftrag 16.01.2013
AUTO 201301160000754	released	16.01.2013 17:22:00		CHILE:00:02-24:00 alle 10 Minuten
AUTO 201301160000753	released	16.01.2013 17:30:00		EST
AUTO 201301160000752	released	16.01.2013 17:20:00		
AUTO 201301160000751	released	16.01.2013 17:20:00		DI-DO P 5min : GM_ZJGM0SLP - 10:34 min schlafen
AUTO 201301160000750	released	16.01.2013 17:20:00		
AUTO 201301160000749	released	16.01.2013 17:20:00		00-06:P60/06-19:P120/19-24:P30
AUTO 201301160000748	active	16.01.2013 17:15:01	16.01.2013 17:15:18	
AUTO 201301160000747	released	16.01.2013 17:30:00		

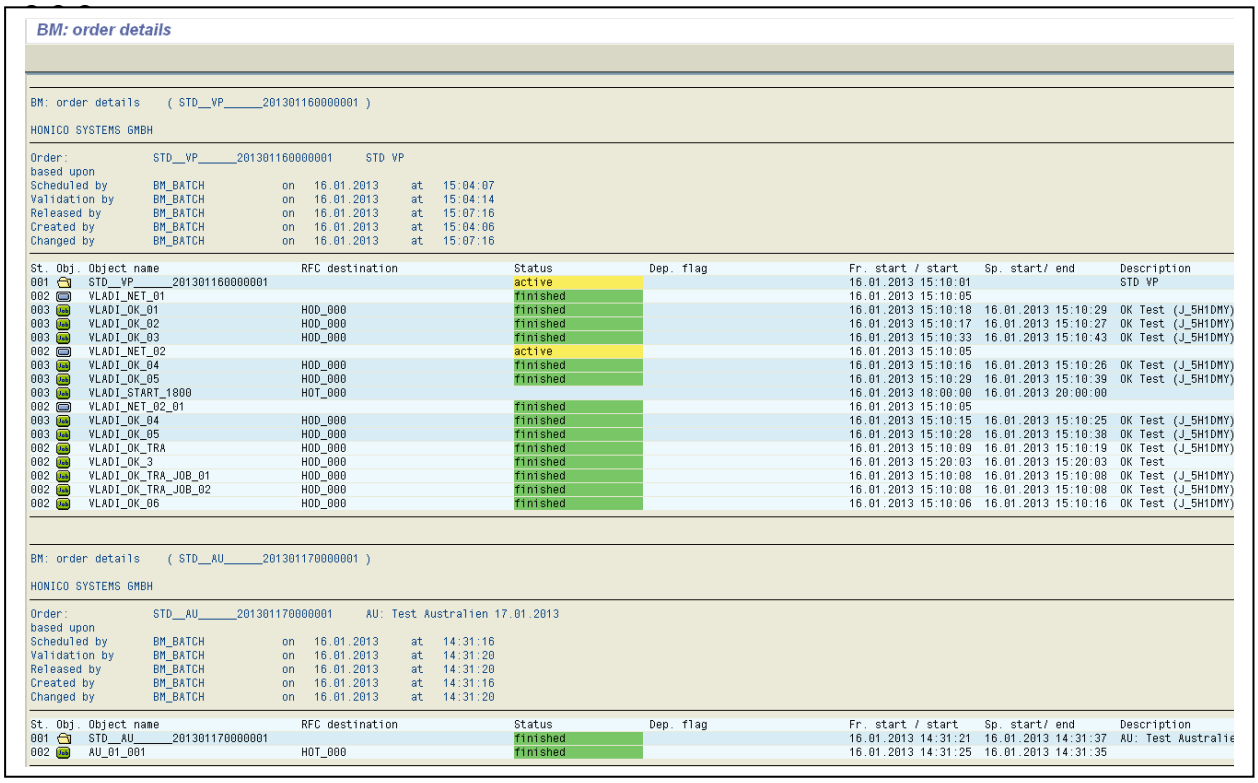
By clicking on the order, you can branch to the order maintenance.

3.9.2. Order details

This program delivers an overview of selected orders. Since a large amount of orders exist over time, you should make a SELECTION here. If the selection is not restricted, the program may need to run for longer time (until a TIME OUT).



The results list shows the details for the respective orders:



BM: order details

BM: order details (STD__VP_____201301160000001)

HONICO SYSTEMS GMBH

Order:	STD__VP_____201301160000001	STD VP				
based upon						
Scheduled by	BM_BATCH	on	16.01.2013	at	15:04:07	
Validation by	BM_BATCH	on	16.01.2013	at	15:04:14	
Released by	BM_BATCH	on	16.01.2013	at	15:07:16	
Created by	BM_BATCH	on	16.01.2013	at	15:04:06	
Changed by	BM_BATCH	on	16.01.2013	at	15:07:16	

St.	Obj.	Object name	RFC destination	Status	Dep. flag	Fr. start / start	Sp. start/ end	Description
001		STD__VP_____201301160000001		active		16.01.2013 15:10:01		STD VP
002		VLADI_NET_01		finished		16.01.2013 15:10:05		
003		VLADI_OK_01	HOD_000	finished		16.01.2013 15:10:18	16.01.2013 15:10:29	OK Test (J_5H1DMY)
003		VLADI_OK_02	HOD_000	finished		16.01.2013 15:10:17	16.01.2013 15:10:27	OK Test (J_5H1DMY)
003		VLADI_OK_03	HOD_000	finished		16.01.2013 15:10:33	16.01.2013 15:10:43	OK Test (J_5H1DMY)
002		VLADI_NET_02		active		16.01.2013 15:10:05		
003		VLADI_OK_04	HOD_000	finished		16.01.2013 15:10:16	16.01.2013 15:10:26	OK Test (J_5H1DMY)
003		VLADI_OK_05	HOD_000	finished		16.01.2013 15:10:29	16.01.2013 15:10:39	OK Test (J_5H1DMY)
003		VLADI_START_1800	HOT_000			16.01.2013 18:00:00	16.01.2013 20:00:00	
002		VLADI_NET_02_01		finished		16.01.2013 15:10:05		
003		VLADI_OK_04	HOD_000	finished		16.01.2013 15:10:15	16.01.2013 15:10:25	OK Test (J_5H1DMY)
003		VLADI_OK_05	HOD_000	finished		16.01.2013 15:10:28	16.01.2013 15:10:38	OK Test (J_5H1DMY)
002		VLADI_OK_TRA	HOD_000	finished		16.01.2013 15:10:09	16.01.2013 15:10:19	OK Test (J_5H1DMY)
002		VLADI_OK_3	HOD_000	finished		16.01.2013 15:20:03	16.01.2013 15:20:03	OK Test
002		VLADI_OK_TRA_JOB_01	HOD_000	finished		16.01.2013 15:10:08	16.01.2013 15:10:08	OK Test (J_5H1DMY)
002		VLADI_OK_TRA_JOB_02	HOD_000	finished		16.01.2013 15:10:08	16.01.2013 15:10:08	OK Test (J_5H1DMY)
002		VLADI_OK_06	HOD_000	finished		16.01.2013 15:10:06	16.01.2013 15:10:16	OK Test (J_5H1DMY)

BM: order details (STD__AU_____201301170000001)

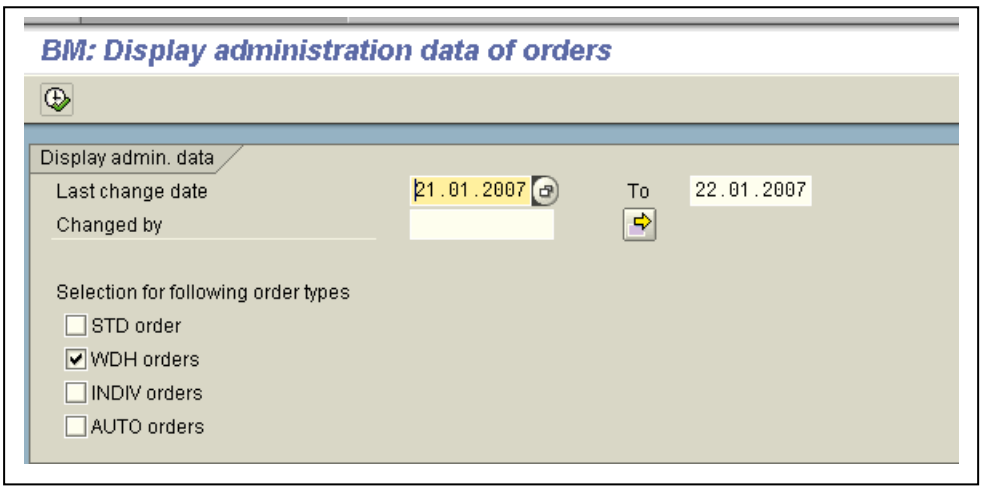
HONICO SYSTEMS GMBH

Order:	STD__AU_____201301170000001	AU: Test Australien 17.01.2013				
based upon						
Scheduled by	BM_BATCH	on	16.01.2013	at	14:31:16	
Validation by	BM_BATCH	on	16.01.2013	at	14:31:20	
Released by	BM_BATCH	on	16.01.2013	at	14:31:20	
Created by	BM_BATCH	on	16.01.2013	at	14:31:16	
Changed by	BM_BATCH	on	16.01.2013	at	14:31:20	

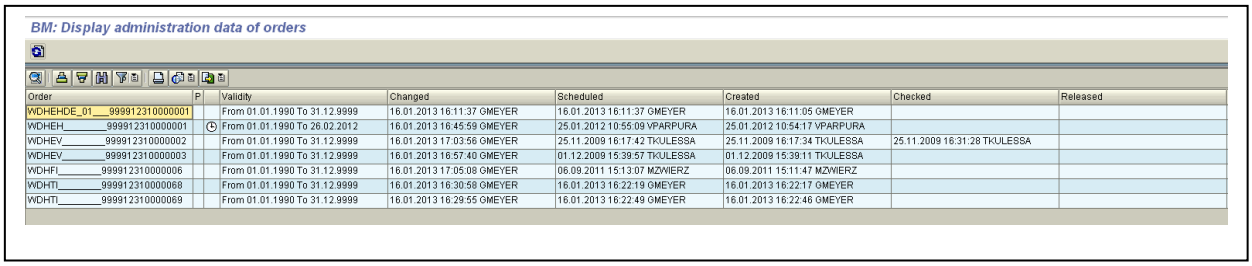
St.	Obj.	Object name	RFC destination	Status	Dep. flag	Fr. start / start	Sp. start/ end	Description
001		STD__AU_____201301170000001		finished		16.01.2013 14:31:21	16.01.2013 14:31:37	AU: Test Australie
002		AU_01_001	HOT_000	finished		16.01.2013 14:31:25	16.01.2013 14:31:35	

Administrative Data Orders

This report displays the administrative data of orders. It can particularly be used for the verification of the last changes of WDH orders (stopping and starting of WDH orders).



Result list:

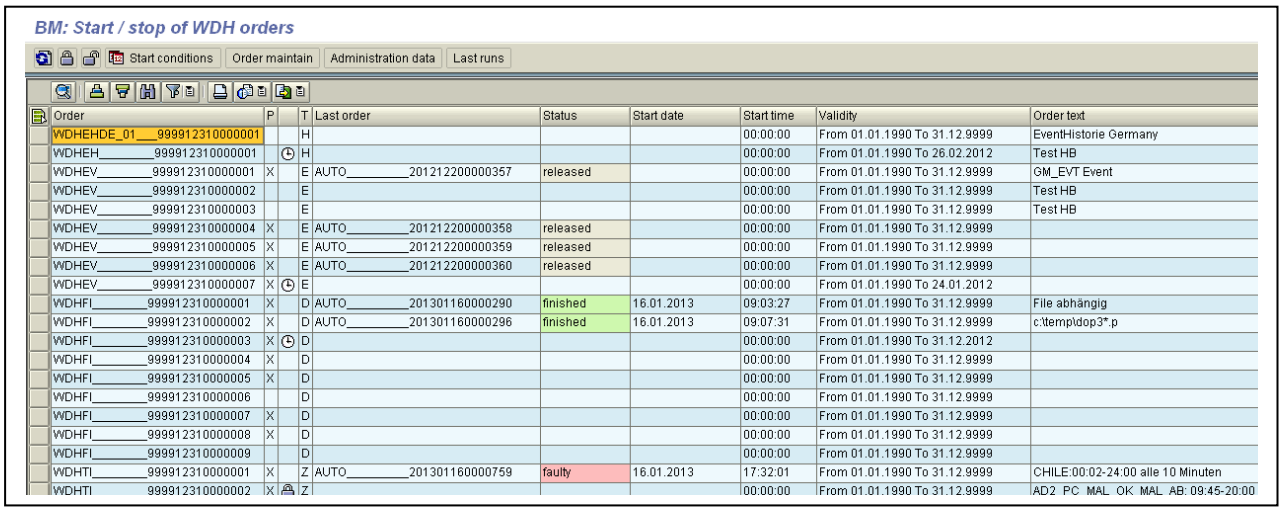


Via double click you can access the order maintenance

3.9.4. Starting/ Stopping WDH Orders

This report is used for stopping and starting WDH orders and belongs to the monitoring tasks. This function can therefore be called directly from monitoring.

In the list, the last AUTO order that was generated for the respective WDH order can be seen. This way all WDH orders can be monitored in one screen which shows if these orders are still active or if failures occurred.



Order	P		T	Last order	Status	Start date	Start time	Validity	Order text
WDHEHDE_01___999912310000001			H				00:00:00	From 01.01.1990 To 31.12.9999	EventHistorie Germany
WDHEH_____999912310000001			H				00:00:00	From 01.01.1990 To 26.02.2012	Test HB
WDHEV_____999912310000001	X		E	AUTO_____201212200000357	released		00:00:00	From 01.01.1990 To 31.12.9999	GM_EVT Event
WDHEV_____999912310000002			E				00:00:00	From 01.01.1990 To 31.12.9999	Test HB
WDHEV_____999912310000003			E				00:00:00	From 01.01.1990 To 31.12.9999	Test HB
WDHEV_____999912310000004	X		E	AUTO_____201212200000358	released		00:00:00	From 01.01.1990 To 31.12.9999	
WDHEV_____999912310000005	X		E	AUTO_____201212200000359	released		00:00:00	From 01.01.1990 To 31.12.9999	
WDHEV_____999912310000006	X		E	AUTO_____201212200000360	released		00:00:00	From 01.01.1990 To 31.12.9999	
WDHEV_____999912310000007	X		E				00:00:00	From 01.01.1990 To 24.01.2012	
WDHFI_____999912310000001	X		D	AUTO_____201301160000290	finished	16.01.2013	09:03:27	From 01.01.1990 To 31.12.9999	File abhängig
WDHFI_____999912310000002	X		D	AUTO_____201301160000296	finished	16.01.2013	09:07:31	From 01.01.1990 To 31.12.9999	c:\temp\dop3*.p
WDHFI_____999912310000003	X		D				00:00:00	From 01.01.1990 To 31.12.2012	
WDHFI_____999912310000004	X		D				00:00:00	From 01.01.1990 To 31.12.9999	
WDHFI_____999912310000005	X		D				00:00:00	From 01.01.1990 To 31.12.9999	
WDHFI_____999912310000006			D				00:00:00	From 01.01.1990 To 31.12.9999	
WDHFI_____999912310000007	X		D				00:00:00	From 01.01.1990 To 31.12.9999	
WDHFI_____999912310000008	X		D				00:00:00	From 01.01.1990 To 31.12.9999	
WDHFI_____999912310000009			D				00:00:00	From 01.01.1990 To 31.12.9999	
WDHTI_____999912310000001	X		Z	AUTO_____201301160000759	faulty	16.01.2013	17:32:01	From 01.01.1990 To 31.12.9999	CHILE:00:02-24:00 alle 10 Minuten
WDHTI_____999912310000002	X		Z				00:00:00	From 01.01.1990 To 31.12.9999	AD2 PC MAL OK MAL AB: 09:45-20:00

The orders are to be marked and can then be stopped or started.

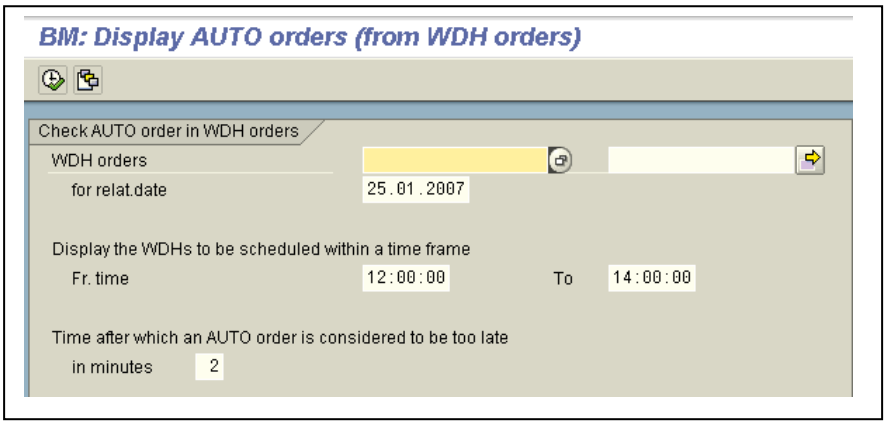
	Stopping marked WDH orders
	Starting marked WDH orders

Furthermore you can switch to **administrative data** (what was changed last and by whom) and to **Last runs** in the WDH orders (see following sections).

Via **Start dates**, the start conditions can be displayed for every WDH order.

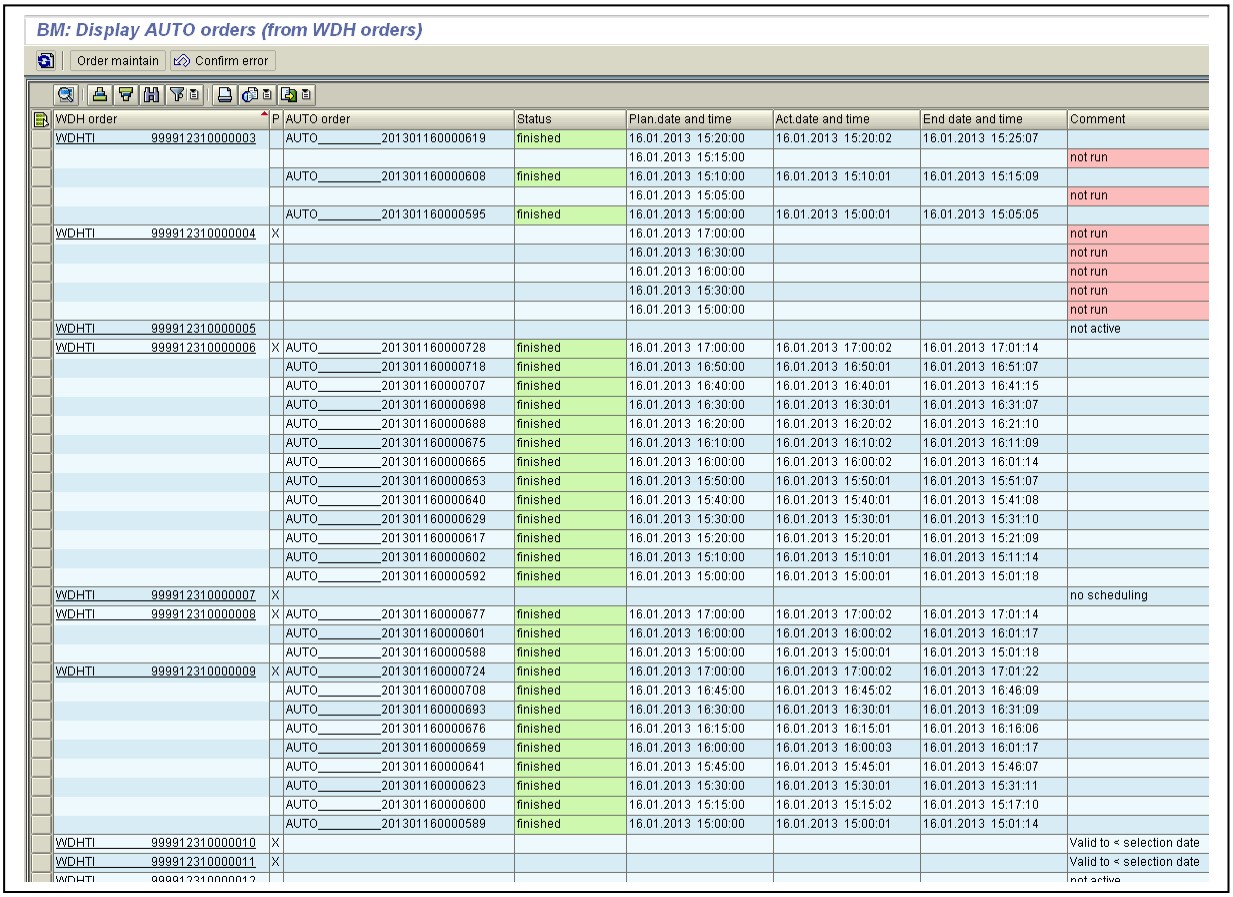
3.9.5. AUTO Orders of WDH orders

Here, the system displays the AUTO orders to be generated for the WDH orders.



The calculation of the AUTO orders derived from the WDH orders is a time-consuming function, if it is supposed to be done for all WDH orders over a long period of time. Therefore, the selection defaults are set on the current day and the last two hours.

Result list:



BM: Display AUTO orders (from WDH orders)

Order maintain | Confirm error

WDH order	P	AUTO order	Status	Plan.date and time	Act.date and time	End date and time	Comment
WDHTI 999912310000003		AUTO_____201301160000619	finished	16.01.2013 15:20:00	16.01.2013 15:20:02	16.01.2013 15:25:07	
				16.01.2013 15:15:00			not run
		AUTO_____201301160000608	finished	16.01.2013 15:10:00	16.01.2013 15:10:01	16.01.2013 15:15:09	
				16.01.2013 15:05:00			not run
		AUTO_____201301160000595	finished	16.01.2013 15:00:00	16.01.2013 15:00:01	16.01.2013 15:05:05	
WDHTI 999912310000004	X			16.01.2013 17:00:00			not run
				16.01.2013 16:30:00			not run
				16.01.2013 16:00:00			not run
				16.01.2013 15:30:00			not run
				16.01.2013 15:00:00			not run
WDHTI 999912310000005							not active
WDHTI 999912310000006	X	AUTO_____201301160000728	finished	16.01.2013 17:00:00	16.01.2013 17:00:02	16.01.2013 17:01:14	
		AUTO_____201301160000718	finished	16.01.2013 16:50:00	16.01.2013 16:50:01	16.01.2013 16:51:07	
		AUTO_____201301160000707	finished	16.01.2013 16:40:00	16.01.2013 16:40:01	16.01.2013 16:41:15	
		AUTO_____201301160000698	finished	16.01.2013 16:30:00	16.01.2013 16:30:01	16.01.2013 16:31:07	
		AUTO_____201301160000688	finished	16.01.2013 16:20:00	16.01.2013 16:20:02	16.01.2013 16:21:10	
		AUTO_____201301160000675	finished	16.01.2013 16:10:00	16.01.2013 16:10:02	16.01.2013 16:11:09	
		AUTO_____201301160000665	finished	16.01.2013 16:00:00	16.01.2013 16:00:02	16.01.2013 16:01:14	
		AUTO_____201301160000653	finished	16.01.2013 15:50:00	16.01.2013 15:50:01	16.01.2013 15:51:07	
		AUTO_____201301160000640	finished	16.01.2013 15:40:00	16.01.2013 15:40:01	16.01.2013 15:41:08	
		AUTO_____201301160000629	finished	16.01.2013 15:30:00	16.01.2013 15:30:01	16.01.2013 15:31:10	
		AUTO_____201301160000617	finished	16.01.2013 15:20:00	16.01.2013 15:20:01	16.01.2013 15:21:09	
		AUTO_____201301160000602	finished	16.01.2013 15:10:00	16.01.2013 15:10:01	16.01.2013 15:11:14	
		AUTO_____201301160000592	finished	16.01.2013 15:00:00	16.01.2013 15:00:01	16.01.2013 15:01:18	
WDHTI 999912310000007	X						no scheduling
WDHTI 999912310000008	X	AUTO_____201301160000677	finished	16.01.2013 17:00:00	16.01.2013 17:00:02	16.01.2013 17:01:14	
		AUTO_____201301160000601	finished	16.01.2013 16:00:00	16.01.2013 16:00:02	16.01.2013 16:01:17	
		AUTO_____201301160000588	finished	16.01.2013 15:00:00	16.01.2013 15:00:01	16.01.2013 15:01:18	
WDHTI 999912310000009	X	AUTO_____201301160000724	finished	16.01.2013 17:00:00	16.01.2013 17:00:02	16.01.2013 17:01:22	
		AUTO_____201301160000708	finished	16.01.2013 16:45:00	16.01.2013 16:45:02	16.01.2013 16:46:09	
		AUTO_____201301160000693	finished	16.01.2013 16:30:00	16.01.2013 16:30:01	16.01.2013 16:31:09	
		AUTO_____201301160000676	finished	16.01.2013 16:15:00	16.01.2013 16:15:01	16.01.2013 16:16:06	
		AUTO_____201301160000659	finished	16.01.2013 16:00:00	16.01.2013 16:00:03	16.01.2013 16:01:17	
		AUTO_____201301160000641	finished	16.01.2013 15:45:00	16.01.2013 15:45:01	16.01.2013 15:46:07	
		AUTO_____201301160000623	finished	16.01.2013 15:30:00	16.01.2013 15:30:01	16.01.2013 15:31:11	
		AUTO_____201301160000600	finished	16.01.2013 15:15:00	16.01.2013 15:15:02	16.01.2013 15:17:10	
		AUTO_____201301160000589	finished	16.01.2013 15:00:00	16.01.2013 15:00:01	16.01.2013 15:01:14	
WDHTI 999912310000010	X						Valid to < selection date
WDHTI 999912310000011	X						Valid to < selection date

No Scheduling	Means that the WDH could not be scheduled in the selected time period (example WDH 2 only runs Monday)
File periodically and Event periodically	Can not be calculated
Not active	WDH order is not released
Stopped	WDH order is released, but stopped
Valid to < selection date	The TO validity in the WDH order is smaller than the selected dated
Valid from > selection date	The FROM validity in the WDH order is larger than the selected date
Not run	An AUTO order was not scheduled although, according to the periods, a scheduling was supposed to be done (Example: WDH 7 would have had to have an AUTO order at 10:30 hours. The reason for this could be that the preceding AUTO ran too long and the concurrency for the WDH is not active, or that the system was switched off. (In the example above, the AUTO order was deleted manually in order to generate the malfunction).

Too late	If the AUTO orders starts running too late, the time can be indicated as parameter when calling the program (2 minutes in this example).

3.10. Archiving Orders

Function tree BatchMan → Reporting → Archiving/ Reorganization

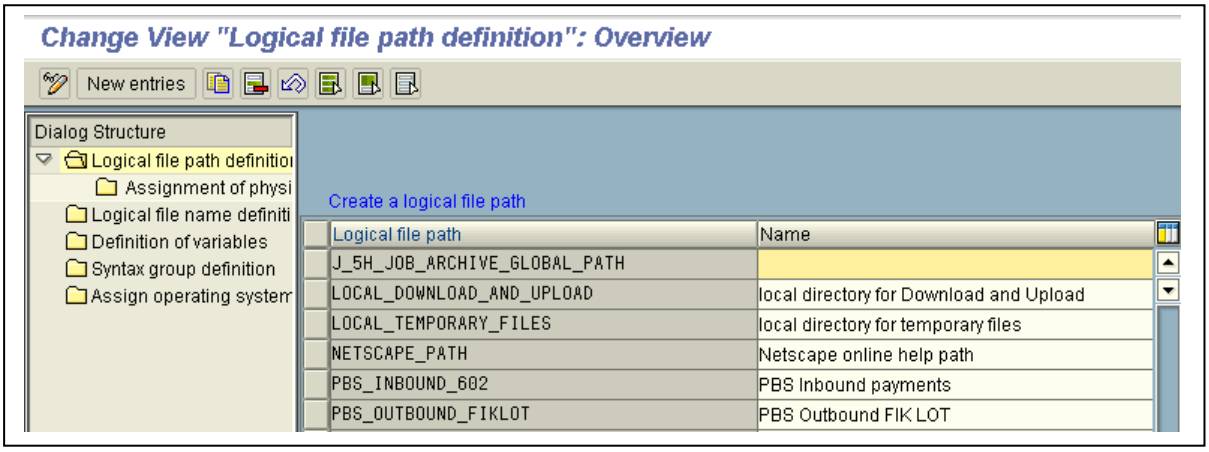
Since orders represent recurring schedulings with at times many objects, they also account for the disk space used by BatchMan. Therefore, it is imperative that the orders be removed regularly from the SAP system.

3.10.1. Customizing Archiving

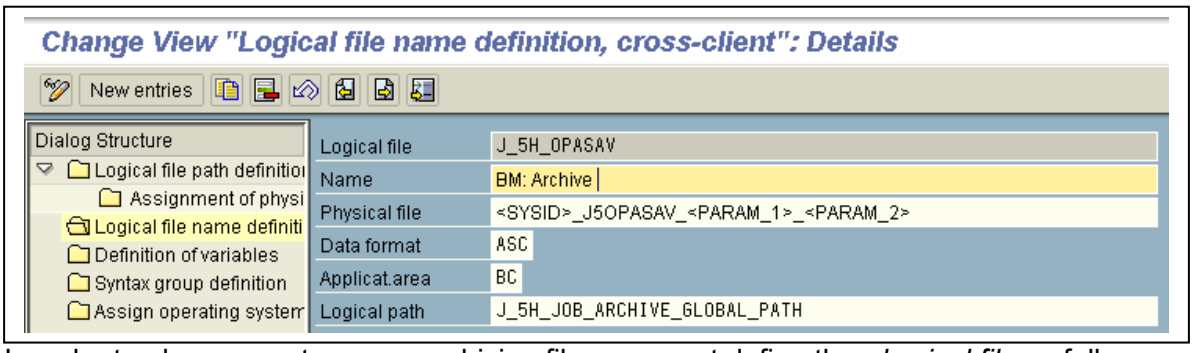
If orders are to be written physically to a file, the related entries are to be maintained in transaction FILE. There the physical name of the file is defined for the file name (J_5H_OPASAV) during archiving. Via logical paths, a path that is independent from the operating system is defined.

The logical files are maintained for the development system and then transported by the system landscape.

Transaction FILE:



- On the left section, selection of ‚*logical file names client – independent*'.
- Double click, so that the right section is updated with the logical file name. Via the button ‚**Positioning...**' an entry field is called up in which the logical file name ‚J_5H_OPASAV' is entered (Entering ‚J' will lead to the first file name which starts with a ‚J').
- Double click on , J_5H_OPASAV' accesses the details of the definition



In order to always create a new archiving file, you must define the *physical file* as follows:

J5OPASAV_<PARAM_1><PARAM_2>

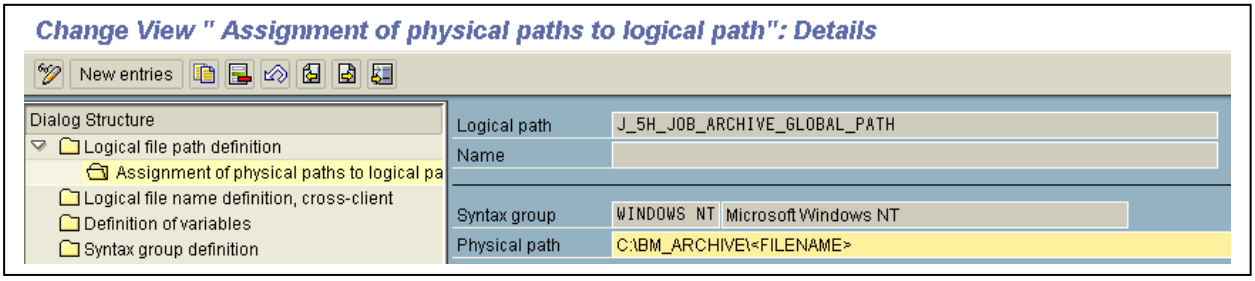
During the creation process, the parameter is replaced accordingly:
PARAM_1 with SY-DATE and
PARAM_2 with SY-TIME

With <DATE> and <TIME> the same result can be reached for archiving; when returning the orders, the system does not find the file since the current date and time would always be taken by SAP.

In the displayed example, parameter **<SYSID>** was still used. This is the SAP system ident (e.g. P01). In job processing with BatchMan only one directory exists that is used by several systems.

Logically, you should create subdirectories in which each SAP system writes for itself, however. This is implemented for archiving via the logical path.

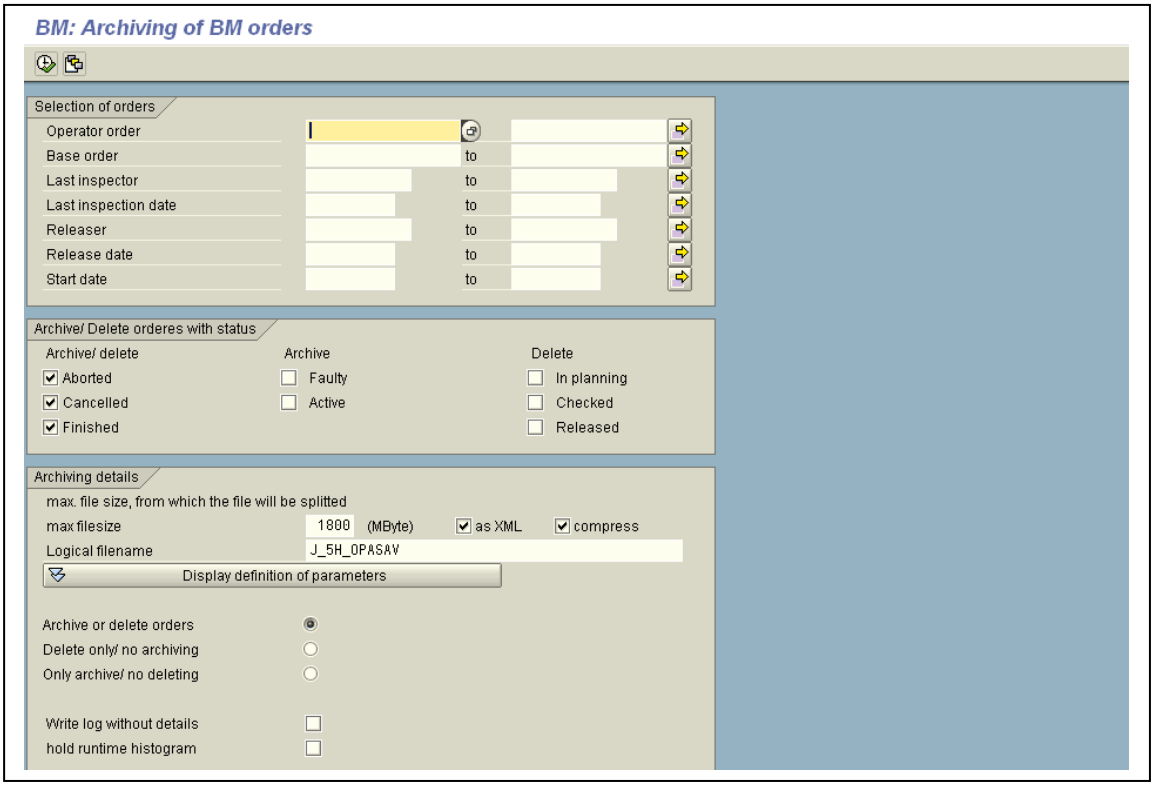
- Selection of *‚Definition of logical file paths'* on the left section
- Double-click to update the right side with the logical file paths. Via the **Position...** pushbutton, search for and mark J_5H_JOB_ARCHIVE_GLOBAL_PATH Then double-click left on “Assignment of physical paths to logical path.” On the right, the system updates the view.
- Select the appropriate operating system (in this case WINDOWS NT). The physical path can now be maintained.



3.10.2. Archiving Orders (/BTCMAN/M_SAV)

Orders can be archived depending on their status. Since BatchMan version 3.0, the archiving information is stored in the table /BTCMAN/M_ARCHIV.

The size of the output file can be limited so that it does not interfere with the 2 Gbyte limit in certain file systems. As standard, 1.8 Gbyte are stored.



Orders with the “Finished,” “Cancelled” and “Aborted” status may be archived. After the archiving, the system will delete the orders from **BatchMan**.

Orders with the “Faulty” and “Active” status may be archived, but are not deleted. Orders with the “In planning,” “Checked” and “Released” status can only be deleted.

If you execute the **Archive and delete orders** function, the system writes to one file the complete orders with all information, such as replies and runtime results. **Then these orders are deleted from BatchMan.**

Only deletion/ no archive: Via this function, you can also delete orders without archiving. A file will not be created on the operating level.

Only archive/ no delete: The entries are written in the Flatfile. The orders stay in the system and must be deleted later (without archiving).

Write Log without details: Entries are made for each archiving run in the application log (transaction SLG1). If the parameter is EMPTY(Space), an entry is made for each deleted order.

If you set the parameter (X), then the system does not write the entry for the orders individually deleted. The system writes only one total entry. This specifies how many records were deleted in total.

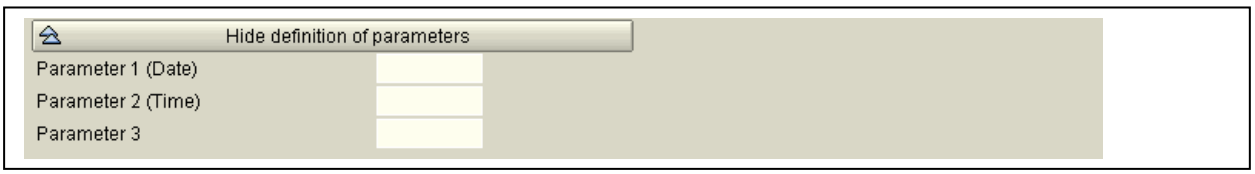
Don’t delete run time histogram: With this option, the run time histogram is not deleted when the archiving option ‚Delete’ appears. This way, the detailed information about objects is available for a longer period of time; but the orders can be deleted earlier, and the **BatchMan** tables can be kept smaller.

Basic order: Entering a basic order enables you to create different variants for archiving/deleting AUTO orders, which are based on WDH orders.

Max file size: This is the size that the archive file may reach at most. For this value, 10% of the maximum file size is to be deducted as assurance.

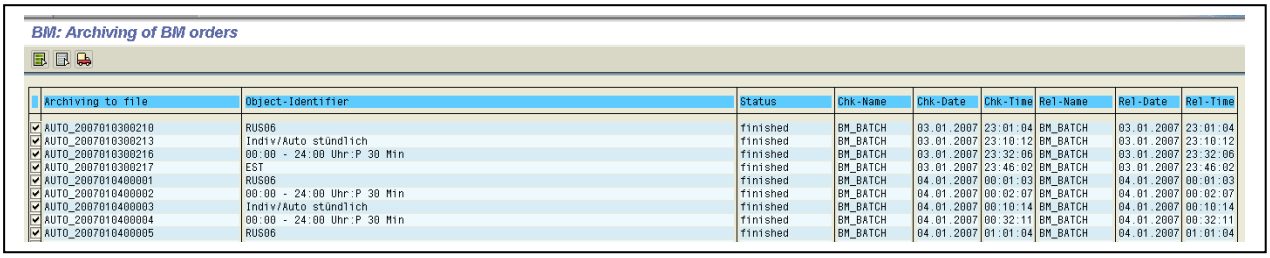
Logical file name: Logical file name that must be defined in transaction FILE. If the file already exists in which the orders are to be archived, then this is **always** expanded.

Via "**Display definition for parameters"** you can replace the standard variables <PARAM_1>, <PARAM_2> and <PARAM_3> with separate definitions.



You can fill the values with fixed values or TVARV variables.

Auswahlliste im Dialog



	Mark all lines
	Cancel marking in all lines
	Write complete orders in the file and delete orders in BatchMan®

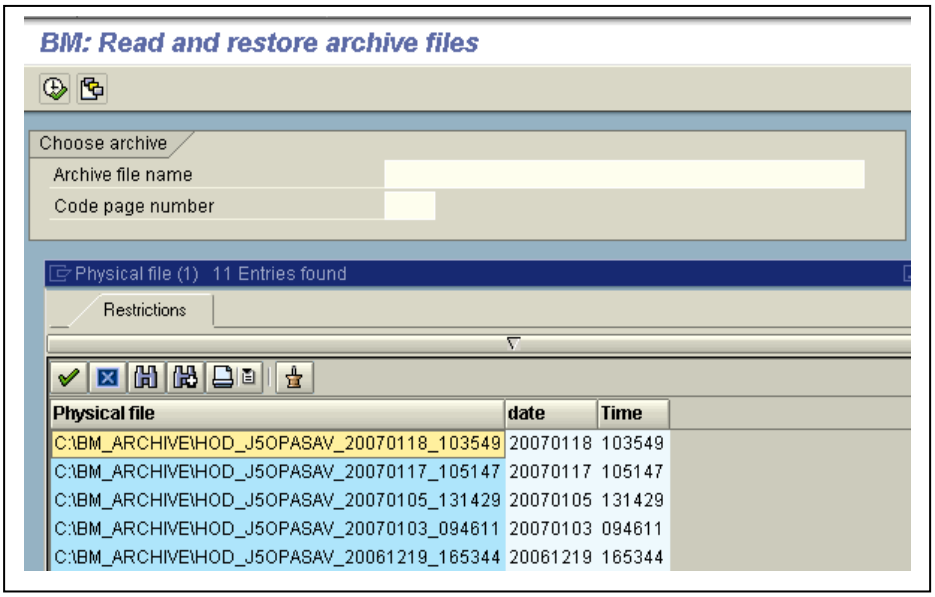
The report is batch-capable and should be added to the normal re-organization of SAP system.

The **Start date** should be deposited during the selection process, including a dynamic date variable or a TVARV variable.

3.10.3. Restoring Orders (/BTCMAN/M_RST)

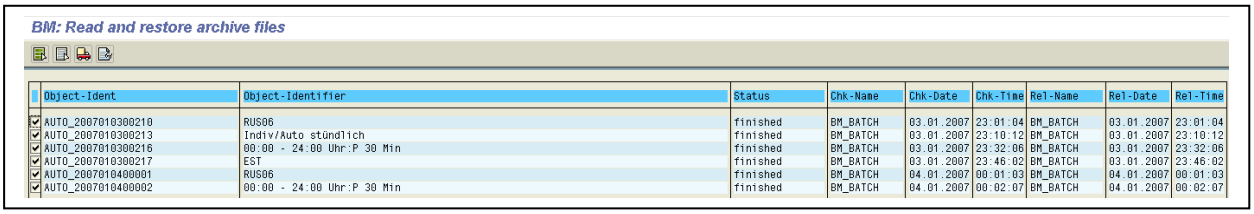
With this function you can restore archived orders in the BatchMan table. It is also possible to display only the log of archived orders.

When old files are read, then they could have been saved with a different **code page** (character set). The code page must be specified accordingly.



Via the search help for the ‚File name for archiving' you can select from the previous archiving runs. A selection mask appears.

Result list in dialog mode:

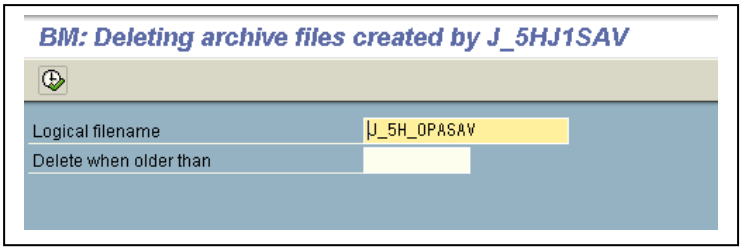


	Mark all lines
	Cancel marking in all lines
	Reading and restoring of the complete orders from the file
	Reading the order log from the file. The orders will not be restored.

An execution in batch is not possible.

3.10.4. Delete Old Archive Files (/BTCMAN/M_SFD)

This report deletes files generated through /BTCMAN/M_SAV (Archiving) on the operating system level. If you work with at least <PARAM_1>, a file is created for each archiving run.



If “Delete if older than” is filled with a TVARV, old files may be automatically deleted. If you leave the field empty, all archiving files are deleted.

You can set up a job network with the reports /BTCMAN/M_SIZ – /BTCMAN/M_SAV – /BTCMAN/M_SFD– /BTCMAN/M_SIZ. In this case, /BTCMAN/M_SIZ delivers the information on the current memory usage by BatchMan tables before and after the archiving.

4. Monitoring

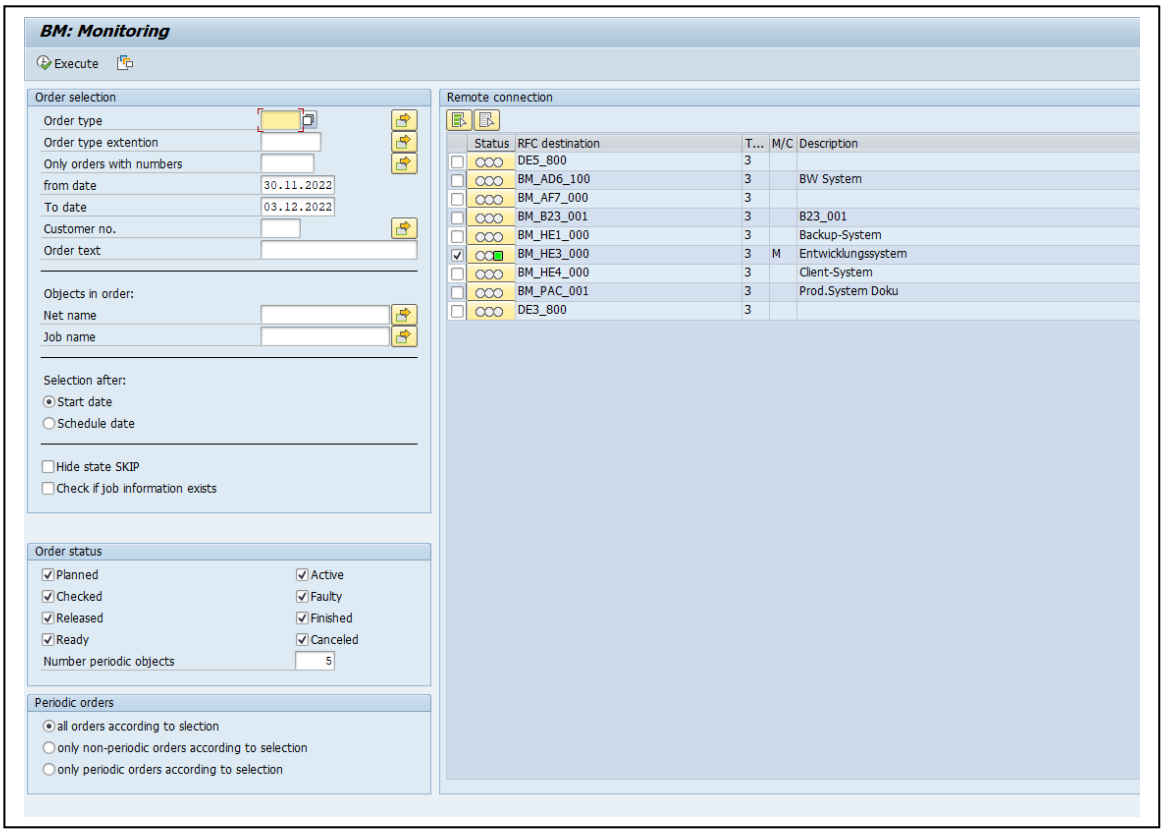
In the menu option ‘Monitoring’ you can follow up on the daily production flow in your own or other/remote systems. Jobs that are not scheduled via orders can also be displayed via the job overview (similar to SM37).

4.1. Order Monitoring

4.1.1. Initial Steps - Monitoring

The order monitor can read and display orders from **BatchMan** systems, which are installed as master systems. It is used for local and also remote order monitoring.

In the initial screen, the selected number of orders can be specified with the help of selection criteria.



On the left side are the selection conditions; on the right side, the system displays the systems to be monitored.

Whether a system is included in the monitoring, is to be determined already for the maintenance of the connection. This is described in the Installation Manual - Section 9 “Make known RFC connections in **BatchMan**”

As a rule, only **BatchMan** master systems can be included in monitoring since only these orders schedule and can thus deliver data.

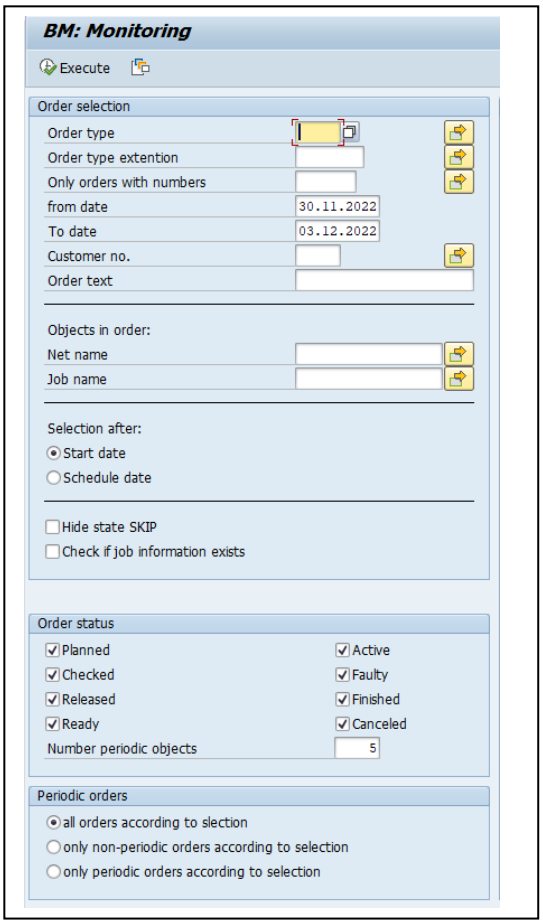
However, you may want the system to display here the availability of a **BatchMan** client system. For this, a “ping” command is triggered. If this is successful, the traffic light turns green. Via the availability (login possible) of the SAP systems, this reveals nothing, however.

By selecting the systems that are to be included in monitoring, you can change the view. This only applies to **BatchMan** master systems.

These settings are saved when you exit the monitoring in user-specific values within **BatchMan**. When you call monitoring again, the system reads these and makes the settings accordingly.

With the user parameter **J5K,** you can influence the default value of the selection screen. The first four bytes of the parameter are evaluated as follows:

1. Byte: If it is active ‚X‘, skipped orders are not indicated.
2. Byte: If it is active ‚X‘, the field INFO will indicate if information was added to a network or job during order execution
3. Byte: Specification of a single-digit day number to preset the ‚From date‘ field with the system date minus this day number
4. Byte: Specification of a single-digit day number to preset the ‚To date‘ field with the system date plus this day number.



Selection

Orders are standardized via the **order type.**

The following order types exist (see also: Order Identification):

INDIV	Individual order
STDxx	Standard order (xx for the plan)
WDH	Repeat order
AUTO	Automatically generated order

An individual order can be selected via the order type, the date selection and a **sequential number**.

From date – to date
The orders are selected based on the start date according to selection.

Orders without start date (start condition event) always appear in the display/overview.

Hide status SKIP:
If the flag is active, the system does not display skipped jobs (i.e., jobs that are/were not executed) in the list display of the order monitor. By using the **J5K** user parameter, the flag was set to "**X"** by default.

Check if job information exists:
If the flags is active ‚**X**', then in the field INFO **an Icon** is displayed if at least to one network or job an information was deposited during order execution.

All orders according to selection:
All orders are selected.

Only non-periodic orders according to selection:
All orders are selected in which the flag ‚periodic' **is not** set.

Only periodic orders according to selection:
All orders are selected in which the flag ‚periodic' is **set**.

The selection can be executed for the following order stages:

Planned

Planned orders encompass the orders that are to be processed in the future

Checked

Checked orders are those orders which have passed the consistency check successfully and therefore could be executed in the SAP batch system in their current format, including jobs and networks.

Released

Released orders have been checked successfully and have been released. These orders have not yet been started.

Ready

These orders have been transformed into the status ‘Ready’ by the scheduler. This state lasts only a very short time and is therefore usually not recognizable.

Active

Orders which are currently being processed have the status ‘Active’.

Finished

Successfully edited orders receive the status ‘finished’. This means that all processes in the order were completed.

Faulty

Orders which had errors in jobs, networks ore in the order itself during processing, receive the status 'faulty'. In the order, not all processed were completed yet.

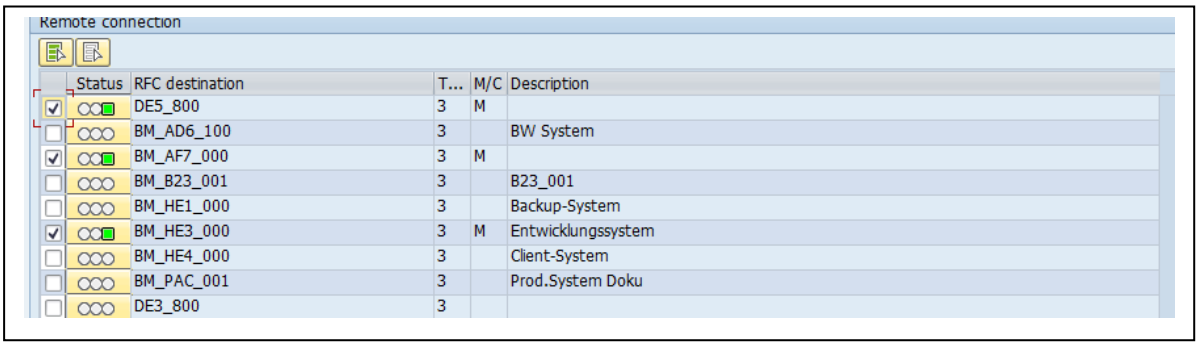
Aborted

Orders, in which errors occurred during processing and in which, nevertheless, all processes were completed, receive the status ’Aborted’.

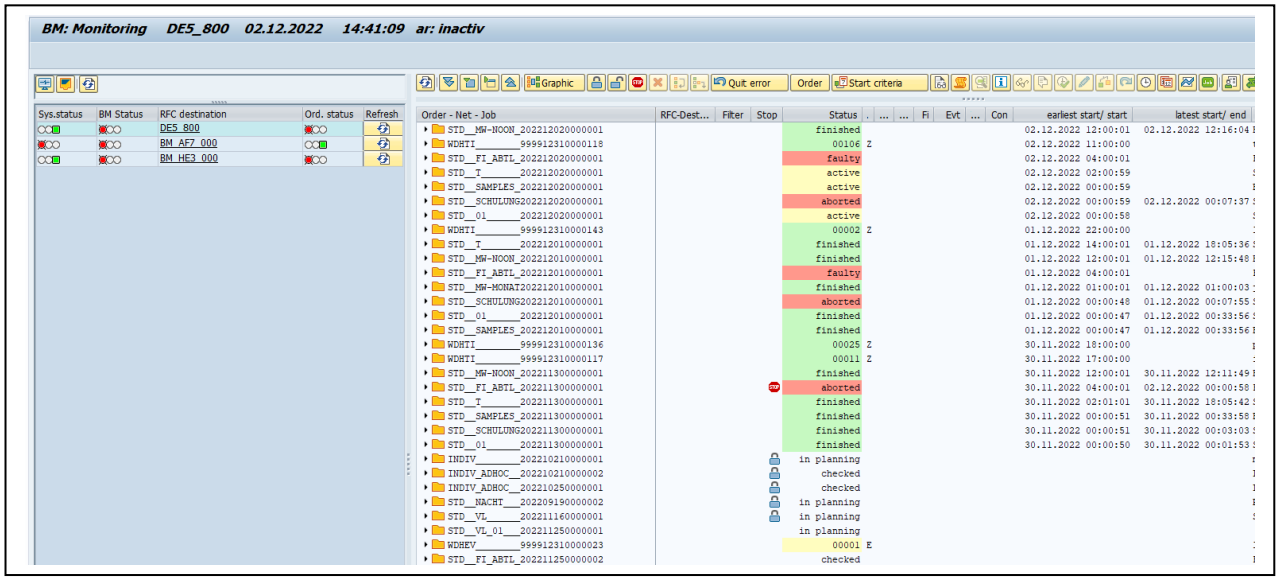
4.1.2. Selection of Orders and Systems

If more than one system was selected (in the system selection), you can switch between these systems in the order display by clicking on the RFC destination.

The status information delivers the information as to whether or not a system (RFC Ping) can be reached.



Above selection deliveres the following display


BM: Monitoring DE5_800 02.12.2022 14:41:09 ar: inactiv
Graphic
Quit error
Order
Start criteria
Sys.status
BM Status
RFC destination
Ord. status
Refresh
DE5_800
BM_AF7_000
BM_HE3_000
Order - Net - Job
RFC-Dest...
Filter
Stop
Status
Fi
Evt
Con
earliest start/ start
latest start/ end

This icon can suppress (hide) the overview of the systems. The view of the orders can be enlarged this way.

This icon restores the overview of the systems.

This icon switches to a pure view of the systems

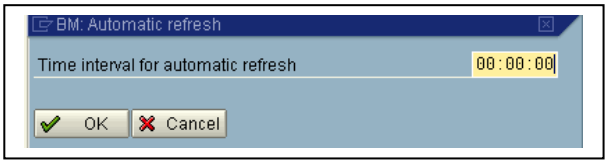
If only one system is marked in the initial screen, switching is not possible.

In the headline, the system (actually the RFC destination), the system date and the system time from the last refresh action is visible.

4.1.3. Refreshing / Updating the Display

Via this button, the display of the orders is updated manually. It is also possible to set up an automatic user-specific update. The function is switched off as standard/default.

The refresh time must be specified first. Call via the menu *Settings → Auto-refresh → Time interval.* Set up the interval accordingly.



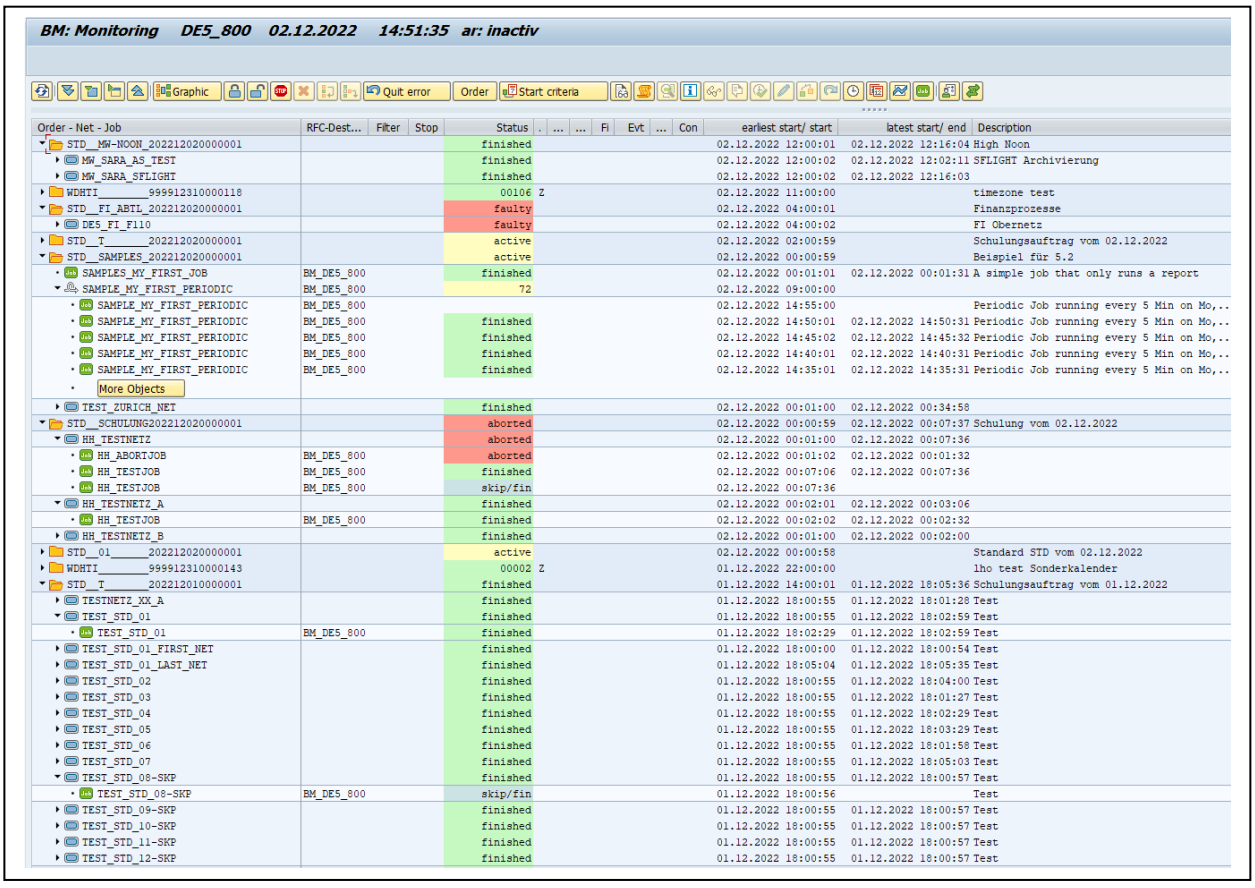
Afterwards, you switch on the automatic update via menu *Settings → Auto-Refresh → Activate*.

Via menu *Settings → Auto-Refresh → Deactivate* you can switch the function off.

4.1.4. Functions in Monitoring

In the description of the individual functions, only the view of one master system is selected because the functions work identically local and remote.

The display is realized via ALV Tree Control. In the left section you can find the tree that contains all selected orders, networks and jobs (objects). Then the columns follow which contain the object information.



In the tree, each object is displayed through an icon

	Order closed, the contained objects are hidden. Clicking on the triangle will open the order.
	Order is opened, the contained objects are displayed. Clicking on the triangle will close the order.
	Network closed, the contained objects are hidden. Clicking on the triangle will open the network.
	Network is opened, the contained objects are displayed. Clicking on the triangle will close the network.
	Periodic job closed, the contained objects are hidden. Clicking on the triangle will open the periodic individual executions of the job.
	Periodic job opened, the contained objects are displayed. Clicking on the triangle will close the periodic individual executions of the job.
	Periodic net closed, the contained objects are hidden. Clicking on the triangle will open the periodic individual executions of the net.
	Periodic net opened, the contained objects are displayed. Clicking on the triangle will close the periodic individual executions of the net.
	Process ‚normal job'
	BW process ‚Process chain'
	FDM Job
	BW process ‚load info package'
	Event process; in a job, ‚Wait for event' was defined. For this event, a physical waiting job is scheduled in SAP.

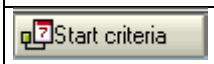
Columns in Monitoring

Order - Network - Job	Tree structure for the monitoring
RFC- Destination	RFC Destination, in which the process is started. Only filled in processes, blank/empty in orders and networks
Stop	Icon (lock closed), if an object (order, network; job) was stopped in monitoring Icon (STOP character/indicator), if the order was finished
Status	Status of the object as text, marked in the appropriate color
P	indicator (X), if the order is periodic
Dep. Flag	Dependency flags for networks and jobs, if these have additional start criteria +D file dependent +R Resource dependent +C Checkpoint dependent

Earliest Start/ Start	Earliest start time as start criterion
Latest Start	Latest start time as start criterion
Description	Text from order maintenance

The list is sorted according to the start date of the orders. Within an order / network, the individual jobs appear according to level, and then the networks contained in the jobs. This sorting can not be changed.

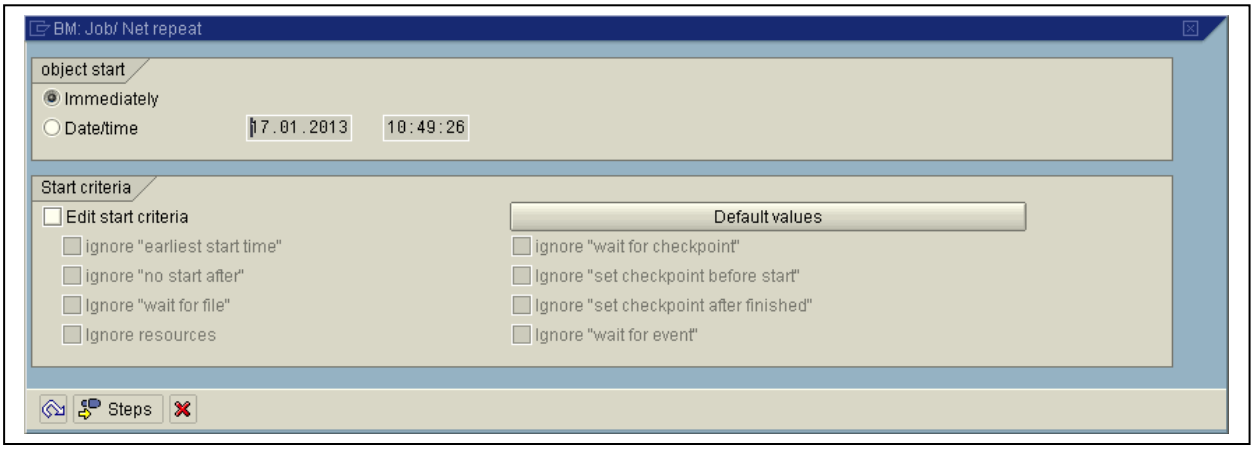
After you mark a node object in the tree, the editing buttons are active. The buttons available depend on which type of process the node represents.

	Order	Calls the graphical display of the order (see Graphic display in monitoring)
	Network	Calls the graphical display of the network (see Graphic display in monitoring)
	Order	Stopping of a marked order. The already active processes (Job/BW) are finished, hence not aborted; but following processes do not start anymore.
	Network	Stopping a marked network. If the network is already running, then a stop is not possible anymore; otherwise the processing proceeds up to the network and waits.
	Job	Stopping a marked job. If the job is already running, then a stop is not possible anymore; otherwise the processing proceeds up to the job and waits.
	O / N / J	Canceling a stop on a locked object. The object must be marked for this.
	Order	Finishing an order, afterwards the order can not be reactivated anymore.
	Job	Aborting a marked ‚normal' job. This function is not possible anymore in most cases.
	Network/ Job	Skipping the marked object
	Network/ Job	Undo Skipping of the marked object that means revoking it. This is only possible if the status is still set on ‚SKIPPED'.
	Order	Error acknowledgement on the marked order. All faulty objects contained in the order are set to the status ‚aborted/ confirmed'. This status is accepted as GOOD (OK) within the processing process, status-dependent processing continues.
	Network	Error acknowledgement on the marked network. All faulty objects contained in the network are set to the status ‚aborted/ confirmed'. This status is accepted as GOOD (OK) within the processing process, status-dependent processing continues.
	Job	Error acknowledgment on the marked job
	Order	Switch/jump to the order maintenance
	O / N / J	Calling the start criteria for the marked object
	O / N / J	Display of the log for the marked object
	Job	Display of the spool output of a job
	Job	Detailed information of a job

	Network/ Job	Deposit information for the marked object. If the internal documentation of the object was already maintained in the master data, then it will be used as model.
	Network/ Job	Emergency start of an object; this is set up/specified to start immediately - independent of the start criteria.
	Job	Maintain job marked
	Job	Check status of the job marked
	Net	Repeat the network (see Repeat Job)
	Job	Repeat the job. In this case, you can select as of which step the repetition occurs from the step list for aborted “normal” jobs. (see Repeat Job)
	Order	Evaluate runtime info for the order marked.
		Start time of an order. If the order is “released,” the start condition may be changed; if the order is (was) already active, the start condition may only be displayed.
		Call of monitoring for WDH orders; see Start/ Stop repeating orders
		Call of **BatchMan** master job monitoring
		Call of user-specific Customizing

More detailed information on functions follows

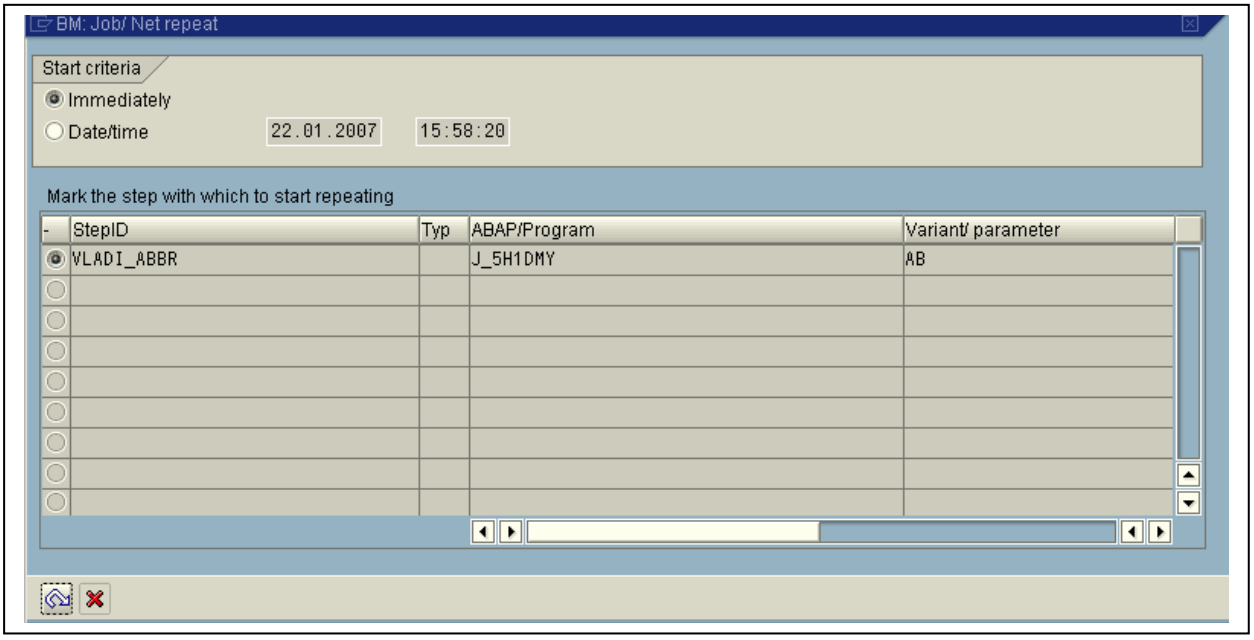
Repeating a job: Mark job and click on button repeat



For the start times, you may select “Immediately” and a date with time here. If the job is to be repeated as defined, you can trigger the start with “Repeat.

You can also decide whether the system is to heed the start criteria.

If the job to be repeated is aborted for instance, you may use “Steps” to specify as of which step the job is to be repeated or whether only certain steps are to be repeated.

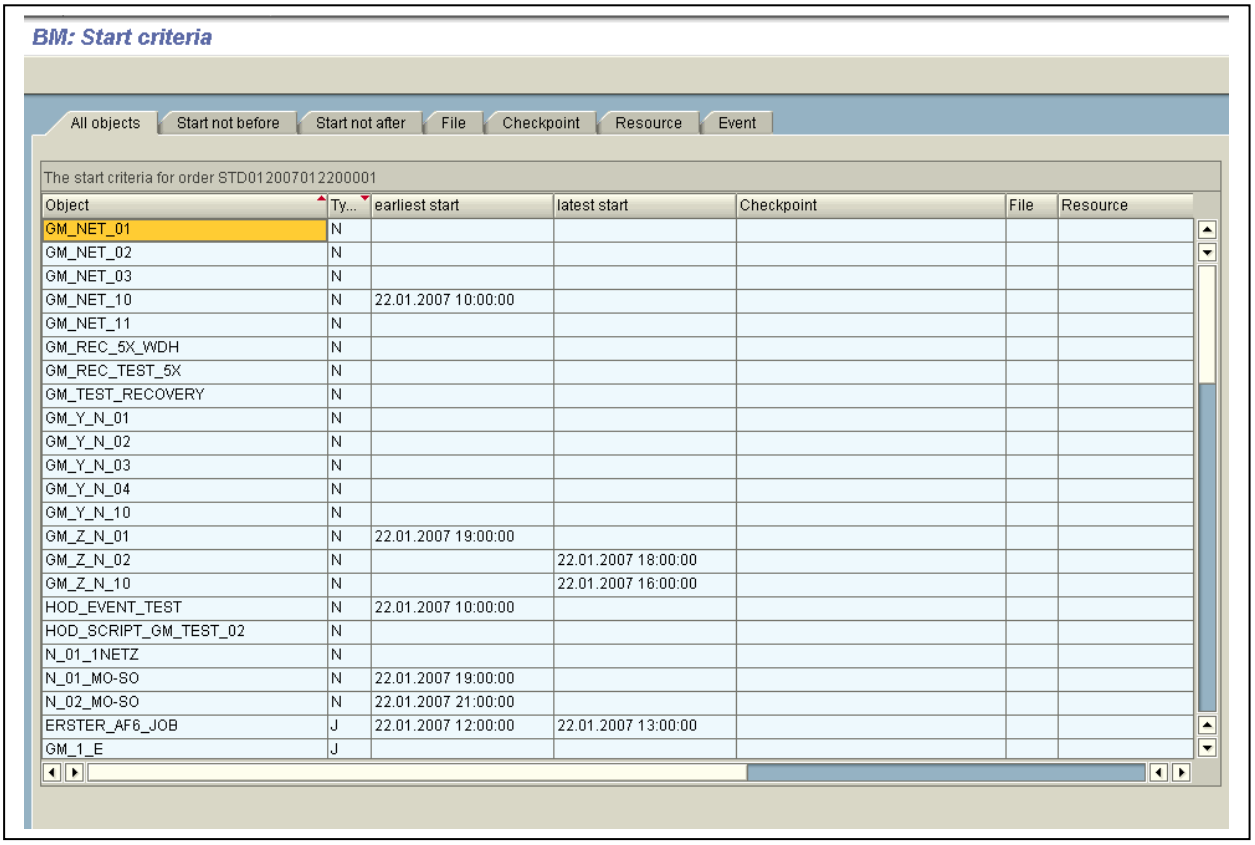


In the case of aborted jobs, the step is marked that was aborted, otherwise it is the first step.

In order to repeat networks, start conditions must be specified. Repeating of networks is only possible if all contained objects were finished.

The status of the object, that is repeated, is changed from ‚aborted’ to ‚aborted/confirmed’, if the object was aborted.

Start criteria of an order: This delivers an overview for the start conditions of an object (in the example an STD01 order).

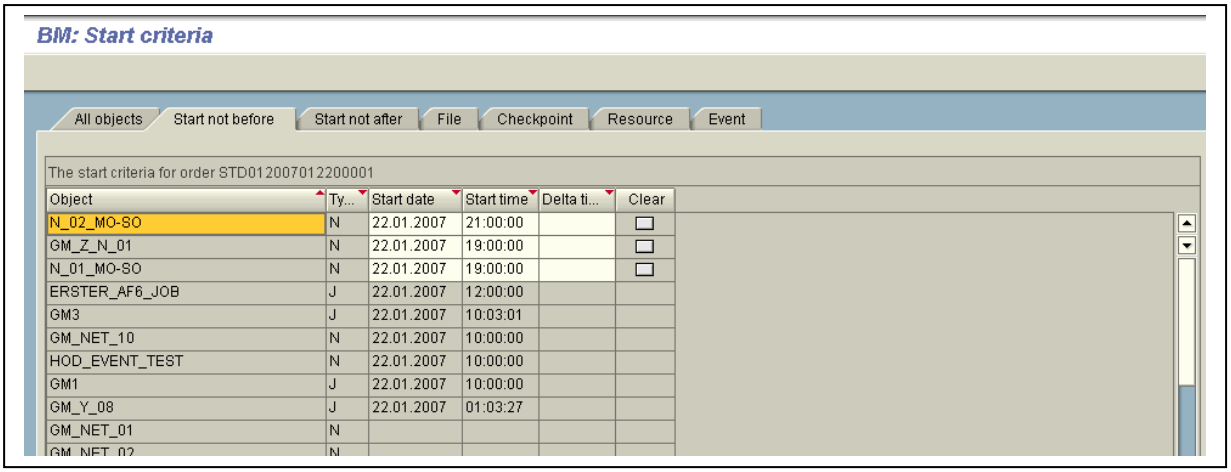


Object	Ty...	earliest start	latest start	Checkpoint	File	Resource
GM_NET_01	N					
GM_NET_02	N					
GM_NET_03	N					
GM_NET_10	N	22.01.2007 10:00:00				
GM_NET_11	N					
GM_REC_5X_WDH	N					
GM_REC_TEST_5X	N					
GM_TEST_RECOVERY	N					
GM_Y_N_01	N					
GM_Y_N_02	N					
GM_Y_N_03	N					
GM_Y_N_04	N					
GM_Y_N_10	N					
GM_Z_N_01	N	22.01.2007 19:00:00				
GM_Z_N_02	N		22.01.2007 18:00:00			
GM_Z_N_10	N		22.01.2007 16:00:00			
HOD_EVENT_TEST	N	22.01.2007 10:00:00				
HOD_SCRIPT_GM_TEST_02	N					
N_01_1NETZ	N					
N_01_MO-SO	N	22.01.2007 19:00:00				
N_02_MO-SO	N	22.01.2007 21:00:00				
ERSTER_AF6_JOB	J	22.01.2007 12:00:00	22.01.2007 13:00:00			
GM_1_E	J					

Via the file cards, you can switch to the individual start criteria.

Under ‚**All objects'** all start criteria of all objects are displayed; even if none were defined. Changes are not possible in this file card.

File card ‚**Start not before**'



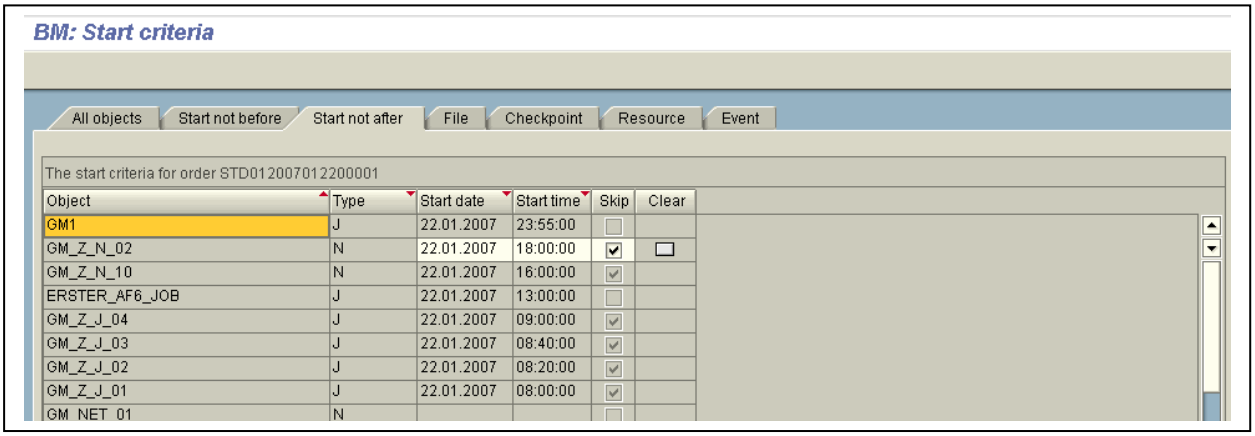
Object	Ty...	Start date	Start time	Delta ti...	Clear
N_02_MO-SO	N	22.01.2007	21:00:00		
GM_Z_N_01	N	22.01.2007	19:00:00		
N_01_MO-SO	N	22.01.2007	19:00:00		
ERSTER_AF6_JOB	J	22.01.2007	12:00:00		
GM3	J	22.01.2007	10:03:01		
GM_NET_10	N	22.01.2007	10:00:00		
HOD_EVENT_TEST	N	22.01.2007	10:00:00		
GM1	J	22.01.2007	10:00:00		
GM_Y_08	J	22.01.2007	01:03:27		
GM_NET_01	N				
GM_NET_02	N				

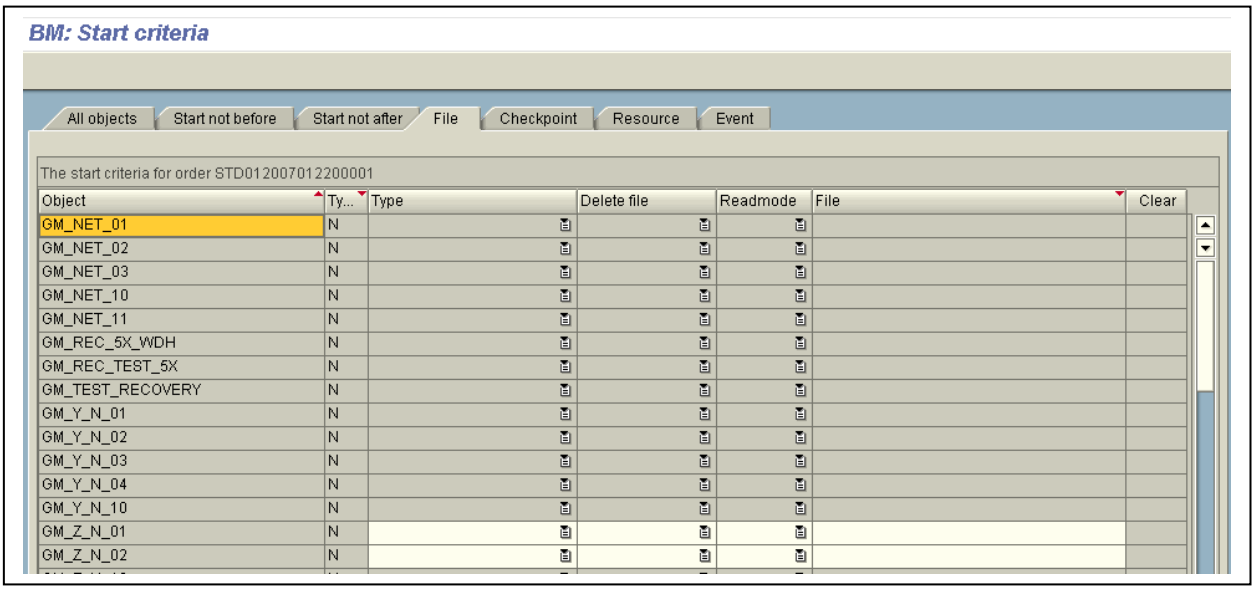
Changes are only possible for objects (networks and jobs) that have not yet run. For these, the fields are ready for input; otherwise that is not the case.

In the fields ready for input, you may change the values. If values are only to be deleted, choose the button for the object in the “Clear” column.

Likewise, the functions described are possible for the other tab pages.

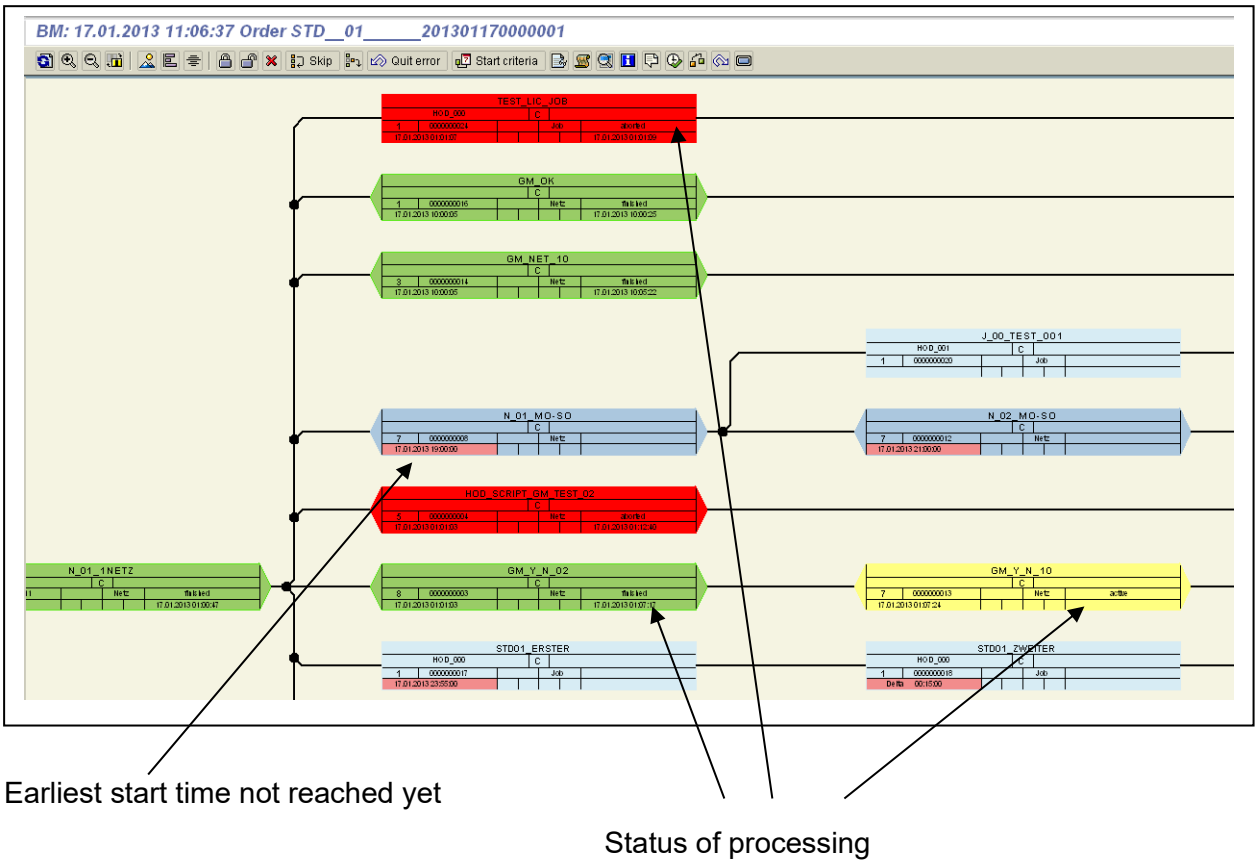
If a field is ready for input, you may also maintain values there subsequently. These are then only valid for this execution/order.





4.1.5. Graphical Displays in Monitoring

If an order or network is marked, then the graphical monitoring can be called. The already described functions are available in the toolbar.



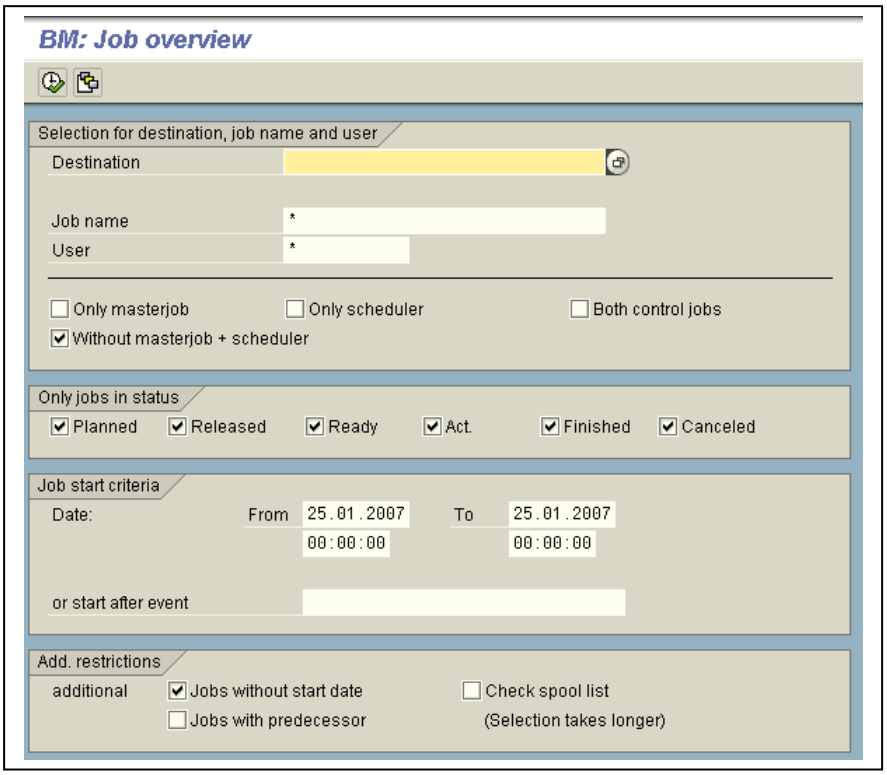
Earliest start time not reached yet

Status of processing

4.2. Job Overview

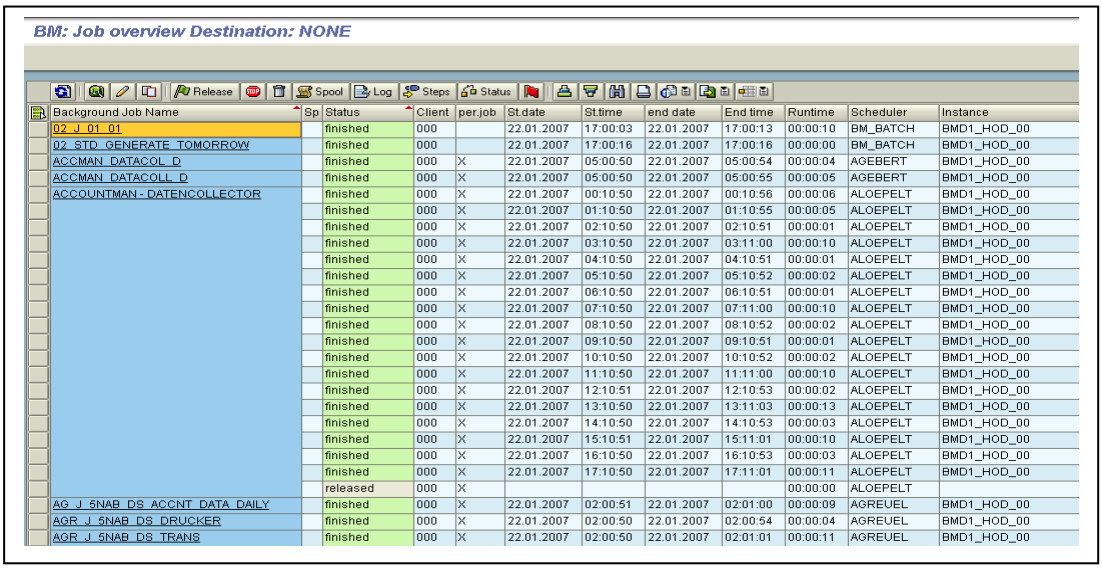
In this menu option, a listed job overview occurs like in SM37. When you use the RFC destination, the jobs may be read on any system.

Whether or not the function works on the remote system depends on what is installed on the system as **BatchMan** transport. The key is which XBP version is available. This report only works as of XBP Version 2.0 since from that version on the functions required are provided.



If you leave the destination field empty, the standard value is “NONE.” The local jobs are read.

Although the list is built like transaction SM37, some functions behave differently or are not possible at the XBP interface.



Background Job Name	Sp	Status	Client	per.job	St.date	St.time	end date	End time	Runtime	Scheduler	Instance
02_J_01_01		finished	000		22.01.2007	17:00:03	22.01.2007	17:00:13	00:00:10	BM_BATCH	BMD1_HOD_00
02_STD_GENERATE_TOMORROW		finished	000		22.01.2007	17:00:16	22.01.2007	17:00:16	00:00:00	BM_BATCH	BMD1_HOD_00
ACCMAN_DATACOL_D		finished	000	X	22.01.2007	05:00:50	22.01.2007	05:00:54	00:00:04	AGEBERT	BMD1_HOD_00
ACCMAN_DATACOLL_D		finished	000	X	22.01.2007	05:00:50	22.01.2007	05:00:55	00:00:05	AGEBERT	BMD1_HOD_00
ACCOUNTMAN - DATENCOLLECTOR		finished	000	X	22.01.2007	00:10:50	22.01.2007	00:10:56	00:00:06	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	01:10:50	22.01.2007	01:10:55	00:00:05	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	02:10:50	22.01.2007	02:10:51	00:00:01	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	03:10:50	22.01.2007	03:11:00	00:00:10	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	04:10:50	22.01.2007	04:10:51	00:00:01	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	05:10:50	22.01.2007	05:10:52	00:00:02	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	06:10:50	22.01.2007	06:10:51	00:00:01	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	07:10:50	22.01.2007	07:11:00	00:00:10	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	08:10:50	22.01.2007	08:10:52	00:00:02	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	09:10:50	22.01.2007	09:10:51	00:00:01	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	10:10:50	22.01.2007	10:10:52	00:00:02	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	11:10:50	22.01.2007	11:11:00	00:00:10	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	12:10:51	22.01.2007	12:10:53	00:00:02	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	13:10:50	22.01.2007	13:11:03	00:00:13	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	14:10:50	22.01.2007	14:10:53	00:00:03	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	15:10:51	22.01.2007	15:11:01	00:00:10	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	16:10:50	22.01.2007	16:10:53	00:00:03	ALOEPELT	BMD1_HOD_00
		finished	000	X	22.01.2007	17:10:50	22.01.2007	17:11:01	00:00:11	ALOEPELT	BMD1_HOD_00
		released	000	X					00:00:00	ALOEPELT	
AG_J_5NAB_DS_ACCNT_DATA_DAILY		finished	000	X	22.01.2007	02:00:51	22.01.2007	02:01:00	00:00:09	AGREUEL	BMD1_HOD_00
AGR_J_5NAB_DS_DRUCKER		finished	000	X	22.01.2007	02:00:50	22.01.2007	02:00:54	00:00:04	AGREUEL	BMD1_HOD_00
AGR_J_5NAB_DS_TRANS		finished	000	X	22.01.2007	02:00:50	22.01.2007	02:01:01	00:00:11	AGREUEL	BMD1_HOD_00

The job name is set up as active link. A click calls up the job log display.

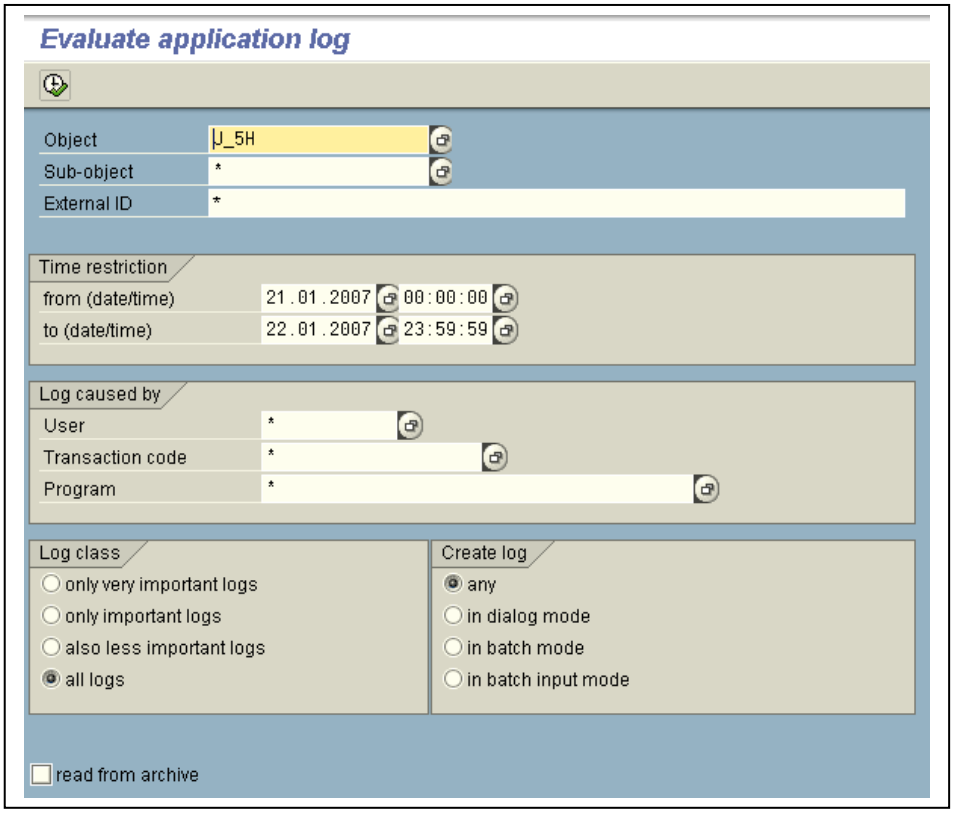
	Refreshing the list
	Display detailed information of a job. The cursor must be placed in the line
	Change job. This function however, is not possible at the XBP interface without problems and not realized at the moment.
	Copy marked job
	Release marked job
	Abort marked job. This function does not work correctly anymore.
	Delete marked job
	Display spool order of marked job
	Calling the job log display
	Display step list of marked job The step list is read from the remote system; the local function from the SM37 is used for the display. If a spool number appears in the step list, then it can only displayed error-free if the jobs were read from the local system.
	Check status of the marked jobs
	Un-schedule/ Cancel released job, for this purpose mark the job
	Sort list ascending, Standard ALV behavior
	Sort list descending, Standard ALV behavior
	Search in the list
	Printing the list
	Calling print preview and Excel inplace
	Exporting the list like e.g. in Excel or as local copy on the frontend
	Calling the Layout functions. This way predefined layout settings can be called, or new layout settings can be defined.

4.3. Application Log

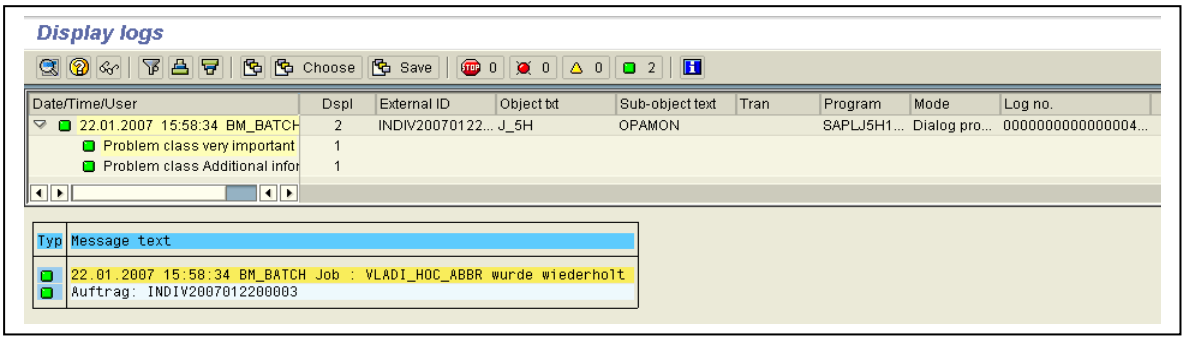
By default, BatchMan uses the application log function provided by SAP for storing information. For this, object J_5H (BatchMan log) and the following sub-objects are used.

ARCH	Logging archive run
AUTO	Logging events of the AUTO-orders (WDH)
CP	Checkpoints
MAINTAIN	Maintenance Window
OPACRE	Ordes created (Remote)
OPADEL	Logging orders deleted SAP/ BATCHMAN®
OPAMON	Logging all actions in running jobs/networks
RES	Resources
SCHED	Job scheduling
SETUP	Logging Setup-usage
STD	Protocols of STD-order generation
SYST	Logging system START/ STOP
WDH	Checking WDH orders

Transaction SLG1 is used for the evaluation. This standard-SAP transaction is **client-dependent**.



When calling from the *BatchMan Function tree: Monitoring → Application log,* the object is already set up as default with ‚J_5H'.



The following actions are written, e.g. for the sub-object OPAMON:

- Stop order/ network/job
- Continue running order/network/job
- Finish/ Finalize an order
- Change job
- Repeat job
- Change start criteria
- Change job status
- Emergency start on network/job
- Skipping and take back Skipping network/job

Errors during the release of jobs in orders are partly still recorded in the SM 21 of the respective system, in which the job should run. These error messages are produced through SAP function modules of the job maintenance.

4.4. Additional Monitors

4.4.1. Monitorino job interface

Via the interface jobs, a special interface is provided, which enables jobs to be registered at this interface. Then the jobs are started and processed by a load and resource distribution differentiated by individual instances and operating modes on the systems.

You can find more detailed information in the BM5_InterfaceManual_EN.pdf document.

4.4.2. Monitoring user jobs

The user jobs form one subset of the interface jobs in BatchMan. User jobs are jobs that are defined, scheduled and started independent of BatchMan via the standard SAP job scheduling. The objective of the user job control in BatchMan is the global transfer and control of these jobs. This has the advantage that end users can continue to schedule and start their jobs themselves, but the administrators receive control over these jobs. All user jobs are subject to a user-job-specific authorization check and are then started according to the resource distribution defined in the job interface.

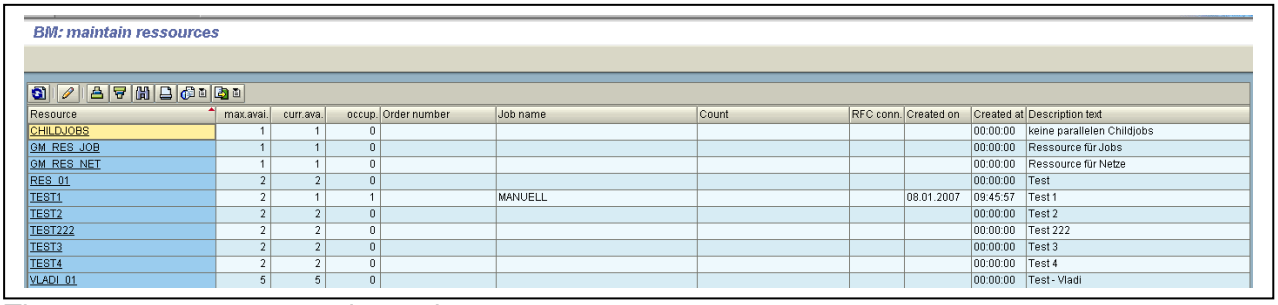
In addition, jobs may also be placed in the user job queue that have the “Users may request this job” flag set. You can find more detailed information in the BM4_d_InterfaceHB.pdf document.

4.5. Checkpoints and Resources

4.5.1. Maintain Resources

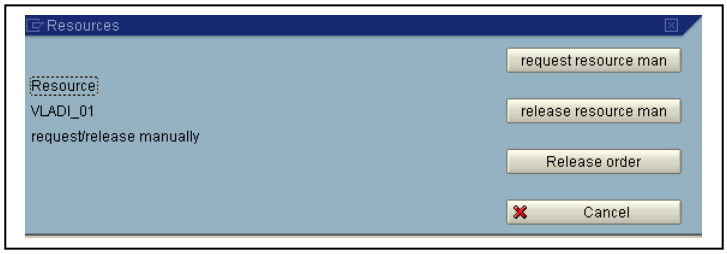
This report is used to monitor resources, to request manual resources and to release manual and order-dependent resources.

Call via ***BatchMan*** *Function tree: Monitoring → Checkpoints and Ressources→ Ressources maintainance*.



Resource	max.avai.	curr.ava.	occup.	Order number	Job name	Count	RFC conn.	Created on	Created at	Description text
CHILDJOBS	1	1	0						00:00:00	keine parallelen Childjobs
GM_RES_JOB	1	1	0						00:00:00	Ressource für Jobs
GM_RES_NET	1	1	0						00:00:00	Ressource für Netze
RES_01	2	2	0						00:00:00	Test
TEST1	2	1	1		MANUELL			08.01.2007	09:45:57	Test 1
TEST2	2	2	0						00:00:00	Test 2
TEST222	2	2	0						00:00:00	Test 222
TEST3	2	2	0						00:00:00	Test 3
TEST4	2	2	0						00:00:00	Test 4
VLADI_01	5	5	0						00:00:00	Test - Vladi

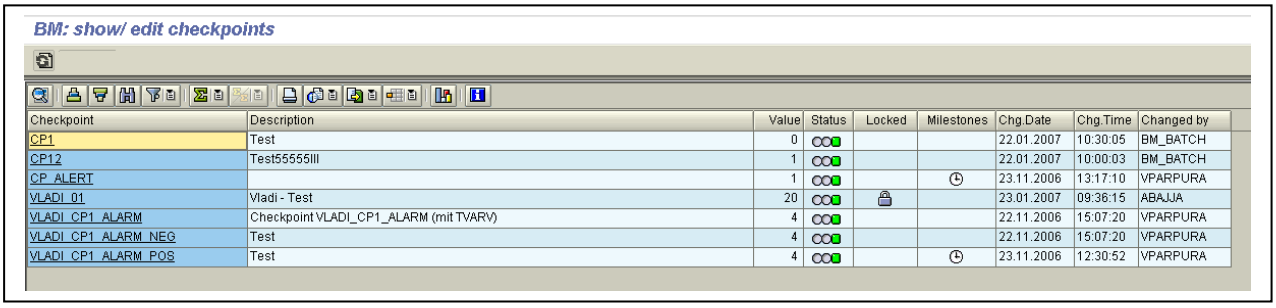
The resources are created as a hotspot in the list. When you click on the resources (or on the change button), the system calls the change function.



4.5.2. Maintain Checkpoints

This report serves to monitor the checkpoints and to change the checkpoint values.

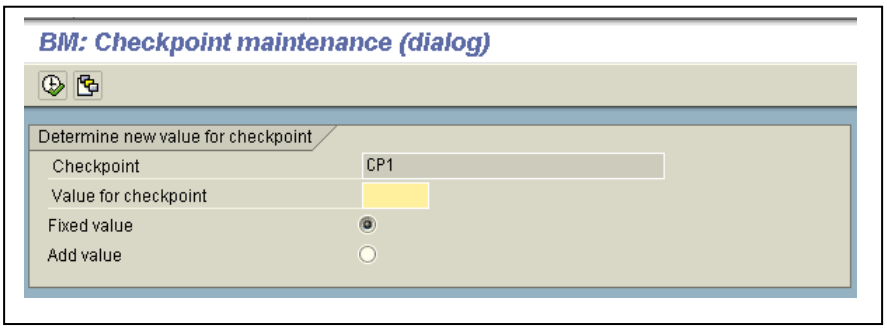
Call via ***BatchMan*** *Function tree: Monitoring → Checkpoints and Resources→ Checkpoint maintenance*. Via selection, the selection of the checkpoints can be limited/restricted.



Checkpoint	Description	Value	Status	Locked	Milestones	Chg.Date	Chg.Time	Changed by
CP1	Test	0				22.01.2007	10:30:05	BM_BATCH
CP12	Test55555III	1				22.01.2007	10:00:03	BM_BATCH
CP_ALERT		1				23.11.2006	13:17:10	VPARPURA
VLADI_01	Vladi - Test	20				23.01.2007	09:36:15	ABAJJA
VLADI_CP1_ALARM	Checkpoint VLADI_CP1_ALARM (mit TVARV)	4				22.11.2006	15:07:20	VPARPURA
VLADI_CP1_ALARM_NEG	Test	4				22.11.2006	15:07:20	VPARPURA
VLADI_CP1_ALARM_POS	Test	4				23.11.2006	12:30:52	VPARPURA

Alert	Lock alert; the alert function that was deposited at the checkpoint is locked. This is made visible through the icon ‚*lock closed*' in the column *locked*.
Alert	Release alert; the icon is deleted.
Value	Change the value of the checkpoint. This function can also be executed by double-clicking on the checkpoint.

To change, the system calls an additional selection screen. There you are to enter and release the value.

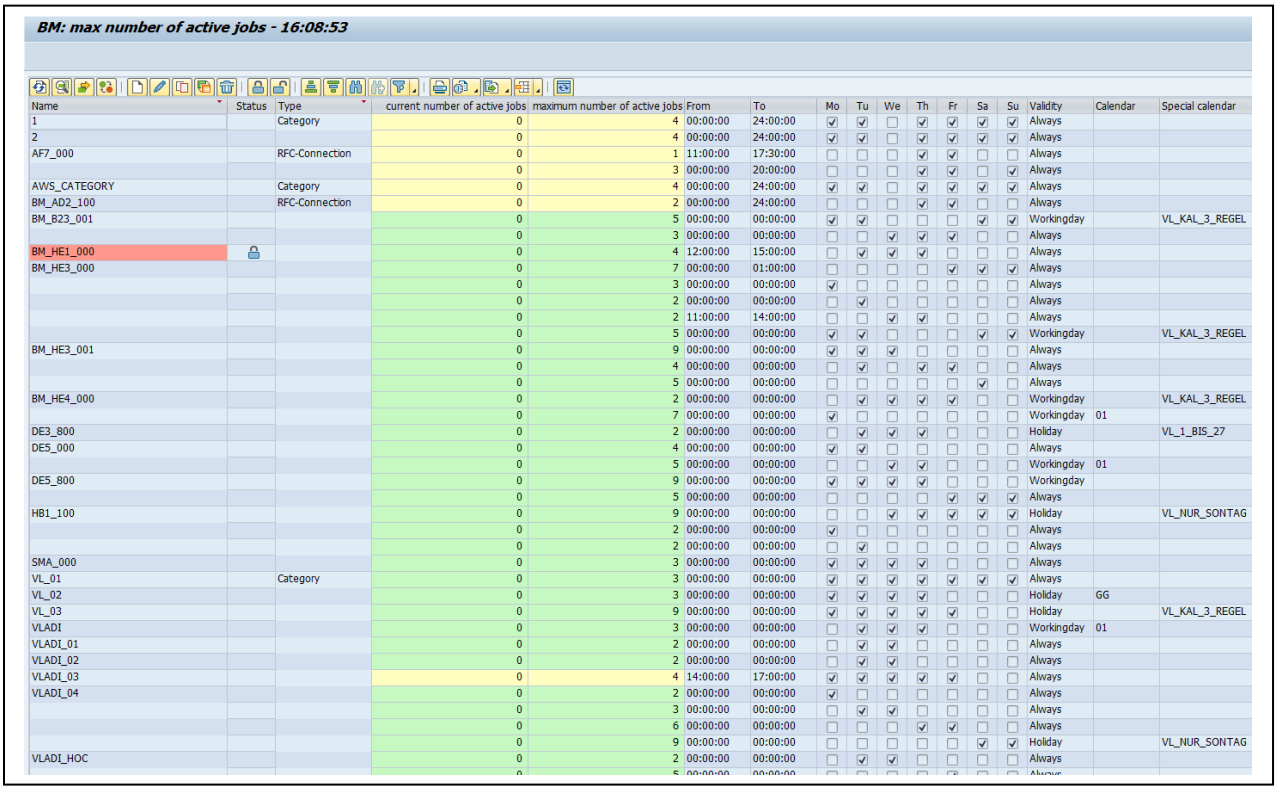


Checkpoints alerting is set up in customizing. This is made visible through a ‚*clock'* icon in the column ‚*milestone'*. If the values are properly maintained, then the alert function is locked after the alert has been triggered. This prevents continuous alert triggering.

4.6. Lastbegrenzung Jobs

This function serves to limit the maximum number of simultaneously active jobs. The maximum number can be set per target system (type: RFC connection) or per category.

The "current number of active jobs" column shows all jobs that correspond to the type of RFC connection or category and are currently active. The "maximum number of active jobs" column shows the maximum possible number of currently active jobs of this type. The difference between these two columns corresponds to the sum of the jobs of this type that could still start at this moment. If the difference is 0 (“current number of active jobs” = “maximum number of active jobs”), no more jobs of this type are started and the active jobs are awaited to be completed.

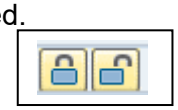


Name	Status	Type	current number of active jobs	maximum number of active jobs	From	To	Mo	Tu	We	Th	Fr	Sa	Su	Validity	Calendar	Special calendar
1		Category	0	4	00:00:00	24:00:00	☑	☑	☐	☑	☑	☑	☑	Always		
2			0	4	00:00:00	24:00:00	☑	☑	☐	☑	☑	☑	☑	Always		
AF7_000		RFC-Connection	0	1	11:00:00	17:30:00	☐	☐	☐	☑	☑	☐	☐	Always		
			0	3	00:00:00	20:00:00	☐	☐	☐	☑	☑	☐	☑	Always		
AWS_CATEGORY		Category	0	4	00:00:00	24:00:00	☑	☑	☐	☑	☑	☑	☑	Always		
BM_AD2_100		RFC-Connection	0	2	00:00:00	24:00:00	☐	☐	☐	☑	☑	☐	☐	Always		
BM_B23_001			0	5	00:00:00	00:00:00	☑	☑	☐	☐	☐	☑	☑	Workingday		VL_KAL_3_REGEL
			0	3	00:00:00	00:00:00	☐	☐	☑	☑	☑	☐	☐	Always		
BM_HE1_000	🔒		0	4	12:00:00	15:00:00	☐	☑	☑	☑	☐	☐	☐	Always		
BM_HE3_000			0	7	00:00:00	01:00:00	☐	☐	☐	☐	☑	☑	☑	Always		
			0	3	00:00:00	00:00:00	☑	☐	☐	☐	☐	☐	☐	Always		
			0	2	00:00:00	00:00:00	☐	☑	☐	☐	☐	☐	☐	Always		
			0	2	11:00:00	14:00:00	☐	☐	☑	☑	☐	☐	☐	Always		
			0	5	00:00:00	00:00:00	☑	☑	☐	☐	☐	☑	☑	Workingday		VL_KAL_3_REGEL
BM_HE3_001			0	9	00:00:00	00:00:00	☑	☑	☑	☐	☐	☐	☐	Always		
			0	4	00:00:00	00:00:00	☐	☑	☐	☑	☑	☐	☐	Always		
			0	5	00:00:00	00:00:00	☐	☐	☐	☐	☐	☑	☐	Always		
BM_HE4_000			0	2	00:00:00	00:00:00	☐	☑	☑	☑	☑	☐	☐	Workingday		VL_KAL_3_REGEL
			0	7	00:00:00	00:00:00	☑	☐	☐	☐	☐	☐	☐	Workingday	01	
DE3_800			0	2	00:00:00	00:00:00	☐	☑	☑	☑	☐	☐	☐	Holiday		VL_1_BIS_27
DE5_000			0	4	00:00:00	00:00:00	☑	☑	☐	☐	☐	☐	☐	Always		
			0	5	00:00:00	00:00:00	☐	☐	☑	☑	☐	☐	☐	Workingday	01	
DE5_800			0	9	00:00:00	00:00:00	☑	☑	☑	☑	☐	☐	☐	Workingday		
			0	5	00:00:00	00:00:00	☐	☐	☐	☐	☑	☑	☑	Always		
HB1_100			0	9	00:00:00	00:00:00	☐	☐	☑	☑	☑	☑	☑	Holiday		VL_NUR_SONTAG
			0	2	00:00:00	00:00:00	☑	☐	☐	☐	☐	☐	☐	Always		
			0	2	00:00:00	00:00:00	☐	☑	☐	☐	☐	☐	☐	Always		
SMA_000			0	3	00:00:00	00:00:00	☑	☑	☑	☑	☐	☐	☐	Always		
VL_01		Category	0	3	00:00:00	00:00:00	☑	☑	☑	☑	☑	☑	☑	Always		
VL_02			0	3	00:00:00	00:00:00	☑	☑	☑	☑	☐	☐	☐	Holiday	GG	
VL_03			0	9	00:00:00	00:00:00	☑	☑	☑	☑	☑	☐	☐	Holiday		VL_KAL_3_REGEL
VLADI			0	3	00:00:00	00:00:00	☐	☑	☑	☑	☐	☐	☐	Workingday	01	
VLADI_01			0	2	00:00:00	00:00:00	☐	☑	☑	☐	☐	☐	☐	Always		
VLADI_02			0	2	00:00:00	00:00:00	☐	☑	☑	☐	☐	☐	☐	Always		
VLADI_03			0	4	14:00:00	17:00:00	☑	☑	☑	☑	☑	☐	☐	Always		
VLADI_04			0	2	00:00:00	00:00:00	☑	☐	☐	☐	☐	☐	☐	Always		
			0	3	00:00:00	00:00:00	☐	☑	☑	☐	☐	☐	☐	Always		
			0	6	00:00:00	00:00:00	☐	☐	☐	☑	☑	☐	☐	Always		
			0	9	00:00:00	00:00:00	☐	☐	☐	☐	☐	☑	☑	Holiday		VL_NUR_SONTAG
VLADI_HOC			0	2	00:00:00	00:00:00	☐	☑	☑	☐	☐	☐	☐	Always		

Yellow line color means that the corresponding rule is currently active from the validity time.

Green line color means that the corresponding rule is currently inactive from the validity time.

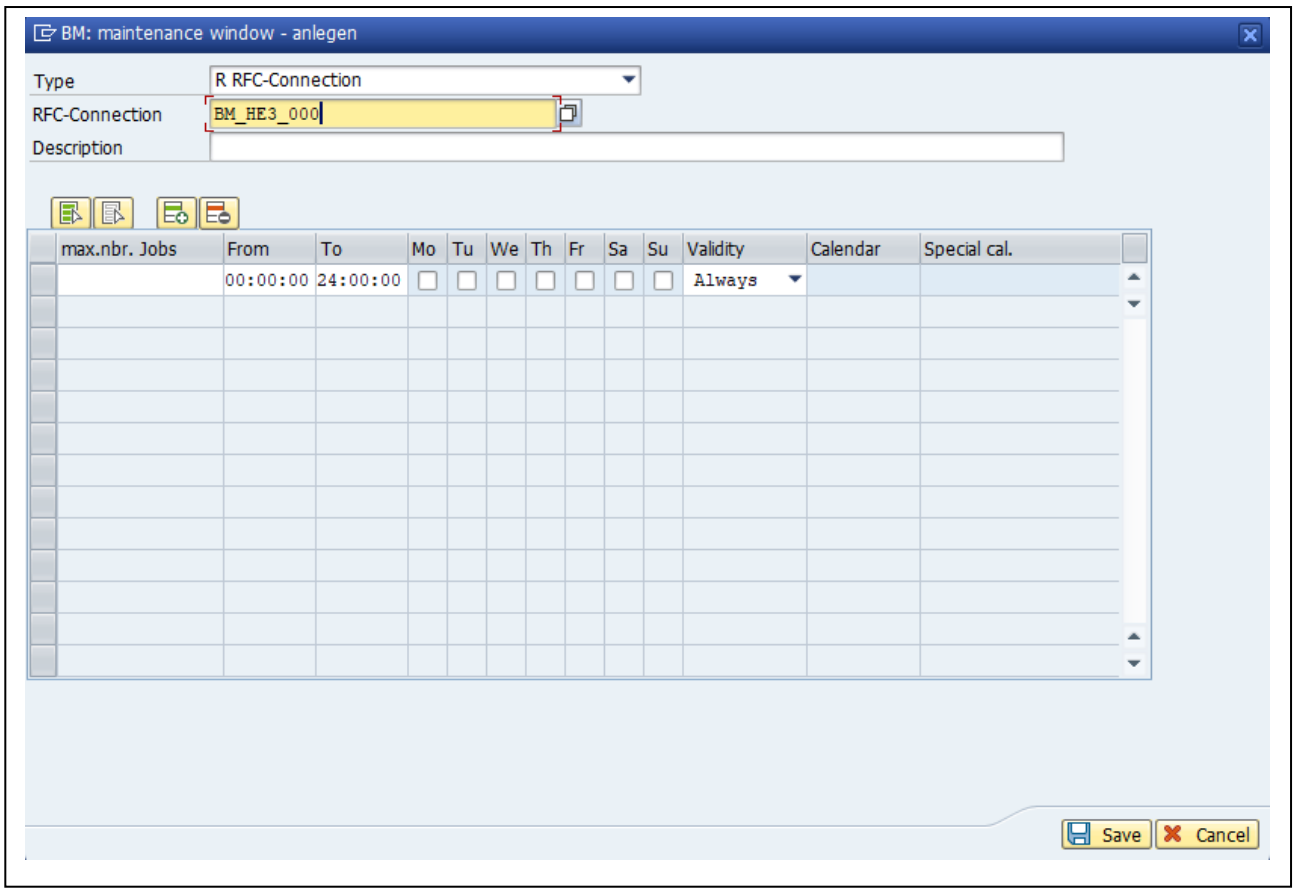
Red line color means that the corresponding rule is locked.

Locking, unlocking a rule can be done via the lock icon. 

If a rule is locked, the system ignores it.

A new rule can be created using the "Create" button. 

To create a rule, the rule type ("RFC Connection" or "Category") must be selected.



Then the period of validity is defined for which this rule should apply and the “max. Number of simultaneously active jobs". Once a rule has been created, it is active immediately - as soon as the validity period has been reached.

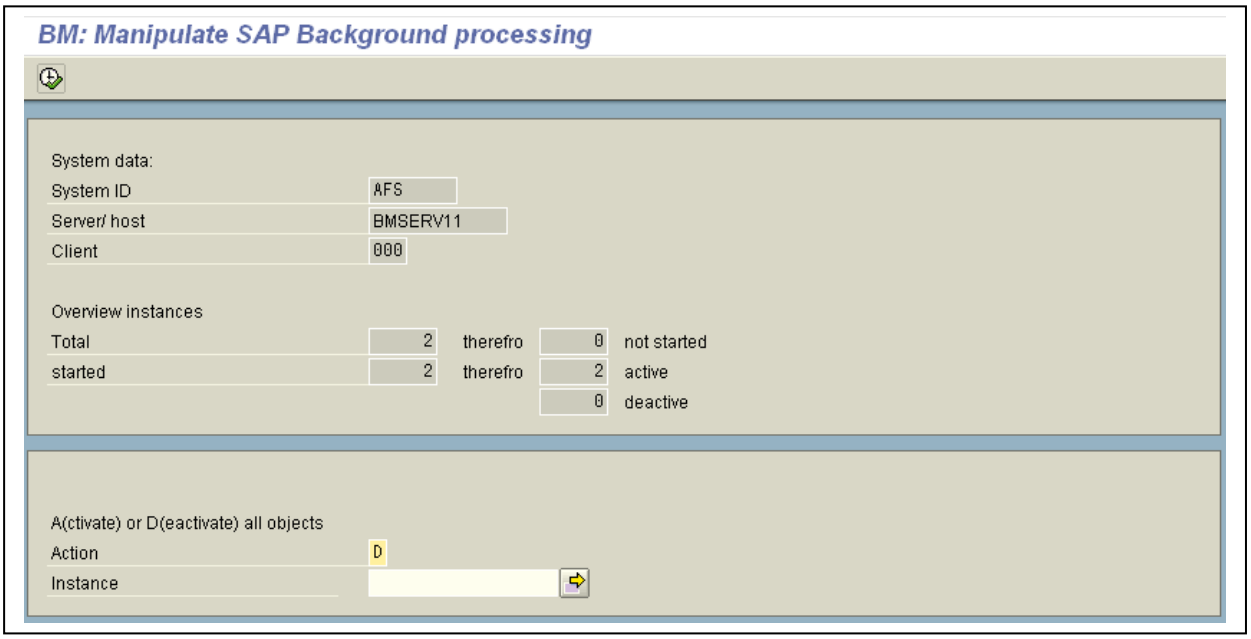
4.7. Scheduler START/STOP

Various actions for starting and stopping of batch activities can be realized through these functions.

4.7.1. Manipulate SAP batch processing

Changed functionality.

Call via *BatchMan Function tree: Monitoring → Scheduler START/STOP → Manipulate SAP batch processing*



The system reads and displays the individual application servers/ instances of the background processing.

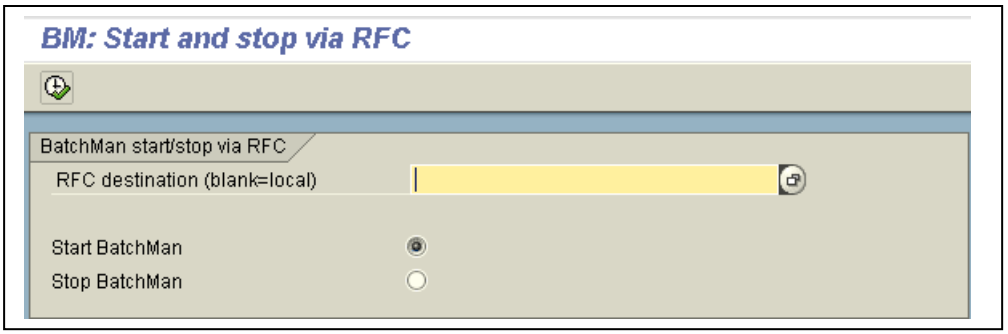
You can set these to the "*inactive*" mode. Conversely, you can also set instances back to the "*active*" mode.

This report gives you the option to influence the instances in the batch operation. You can (and should) execute these activities manually using transaction SM51.

4.7.2. BatchMan START/ STOP via RFC-Connection

With this report (/BTCMAN/M_STO) you can stop and restart BatchMan on remote systems (via RFC Destination). Installation of a BatchMan master system is required.

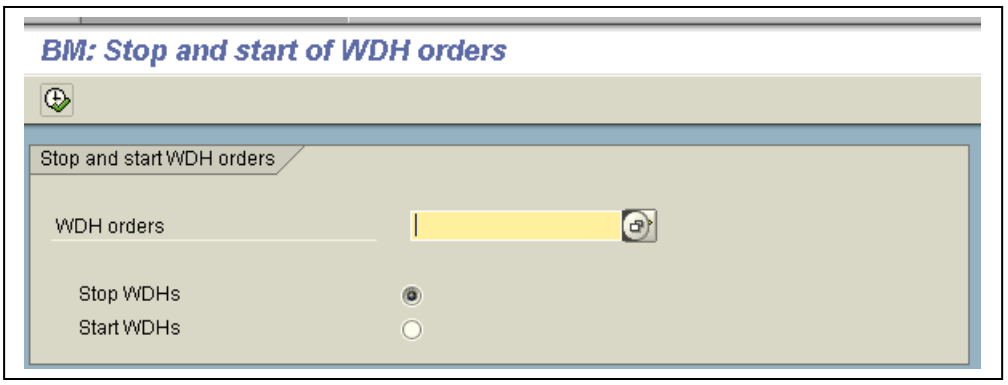
Call via *BatchMan Function tree: Monitoring → Scheduler START/STOP → BatchMan START/STOP via RFC-Connection*



4.7.3. START/STOP WDH Orders

Report (/BTCMAN/M_STW) can stop and start WDH orders via program call. Since this report is batch-capable, it can be included in processing actions.

It can become necessary to stop WDH orders if competing processing is executed/running in another order (e.g. orders). After processing has ended, the WDH is started again.

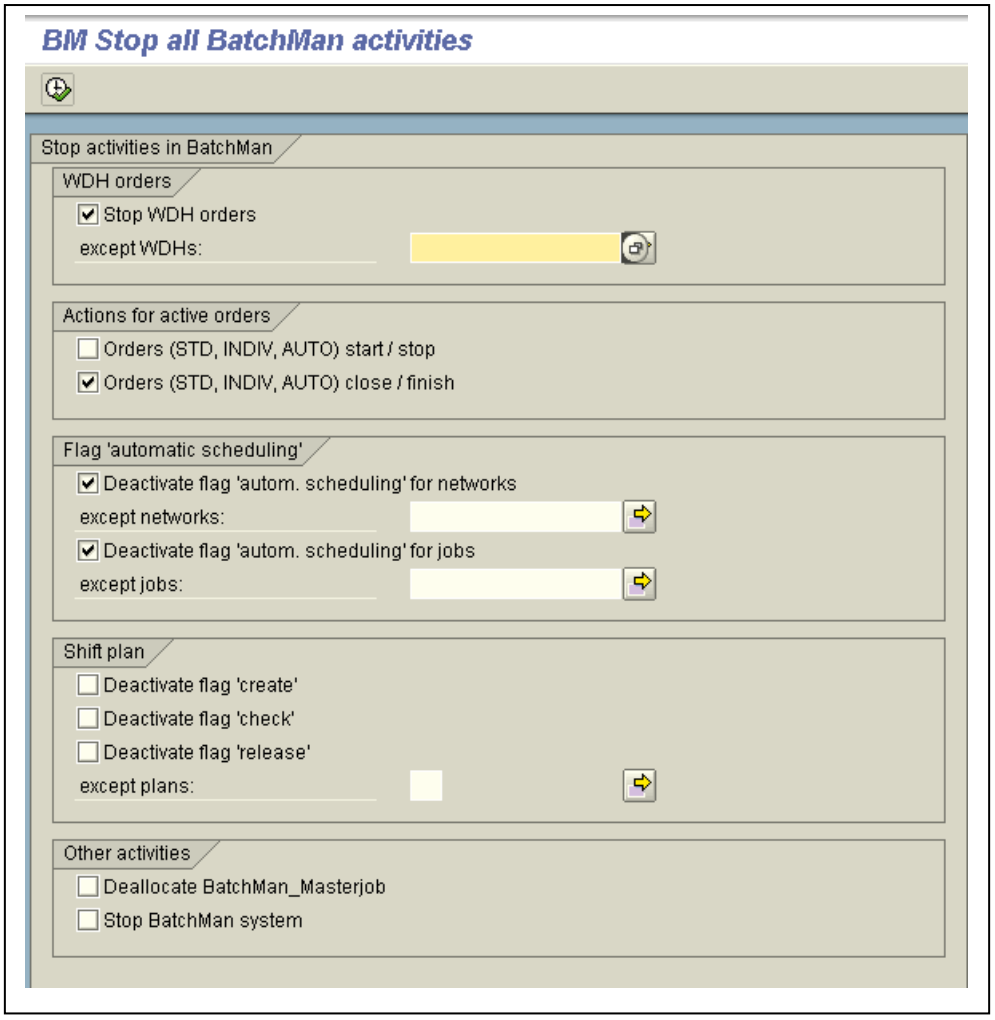


4.7.4. Stopping BatchMan Activities

Program (/BTCMAN/M_SAA) to stop BatchMan activities according to the selected actions. This becomes necessary after copying a productive system to a preceding quality system; if not all BatchMan activities should run there.

1. Repeating orders stop except the WDH selection.
2. Set back the Releasing of WDH orders with option of deletion of these WDH oders
3. Stop or end active orders. Stopped orders remain active and may also continue to run by manual action. Ended orders are closed and cannot be reactivated.

4. Deactivate flag “autom. scheduling” for jobs and networks with the option not to deactivate selected objects.
5. Shift plan: Deactivate flags. If the “Create” flag is to be deactivated, the system also automatically deactivates the flags for “Check” and “Release.” The deactivation of the “Check” flag also deactivates the “Release” flag.
6. Deallocate BATCHMAN_MASTERJOB
7. Stop BatchMan system like transaction J5HI



This report is mainly used after a system copy from the productive system back to a quality system.

A variant is already created on the productive system. After the copying action, this variant is executed in the quality system.

5. Reporting

5.1. Orders

Documentation of reporting functions can be found in the order administration (see Reporting about Orders)

5.2. Nets

Documentation of reporting functions can be found in the master data (see Reporting about Networks)

5.3. Jobs

Documentation of reporting functions can be found in the master data (see Reporting about Jobs)

5.4. Steps

Documentation of reporting functions can be found in the master data (see Reporting about Steps)

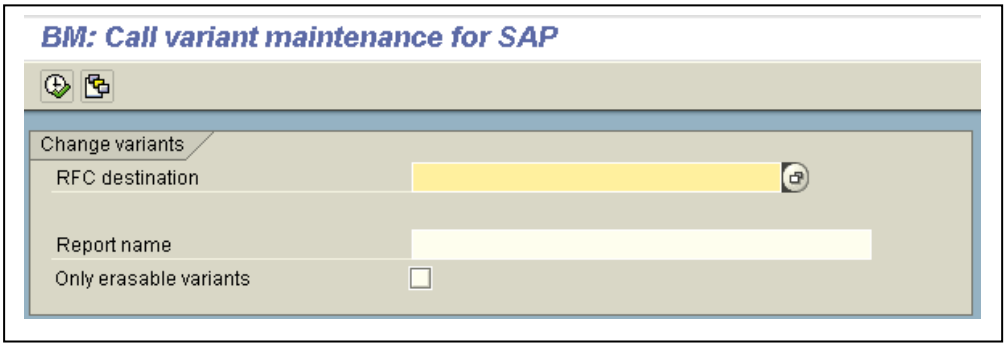
5.5. Variants

In variants, BatchMan uses the SAP variants (client-dependent). Function modules are imported via the master- and client- system transport, which expand the existing SAP functions.

5.5.1. Maintain variants

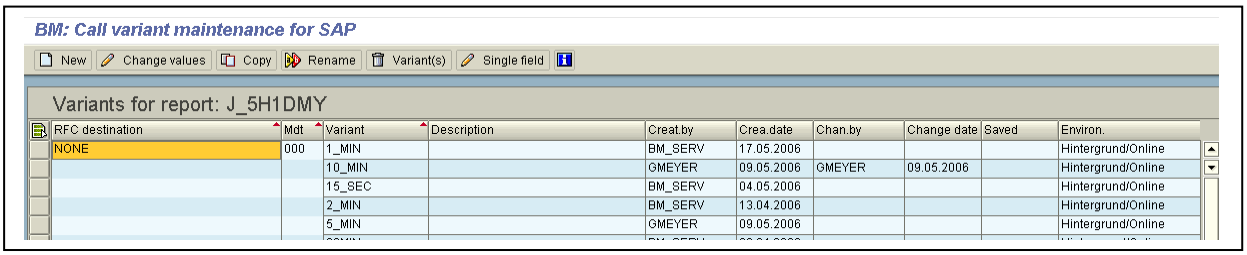
For this program (/BTCMAN/M_VPF), it is possible to create variants in the SAP system and to edit variants that already exist.

***Function tree** BatchMan → Library → Variants → Maintain variants*



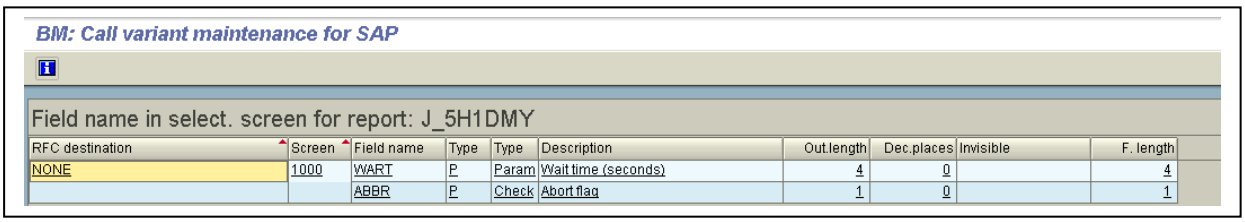
Since BatchMan offers the option to work across systems, the RFC connection is always to be specified. This determines on which system and for which clients the variants are to be maintained.

Variants can always only be maintained for an ABAP/4 program. Also this must have a so-called selection screen.

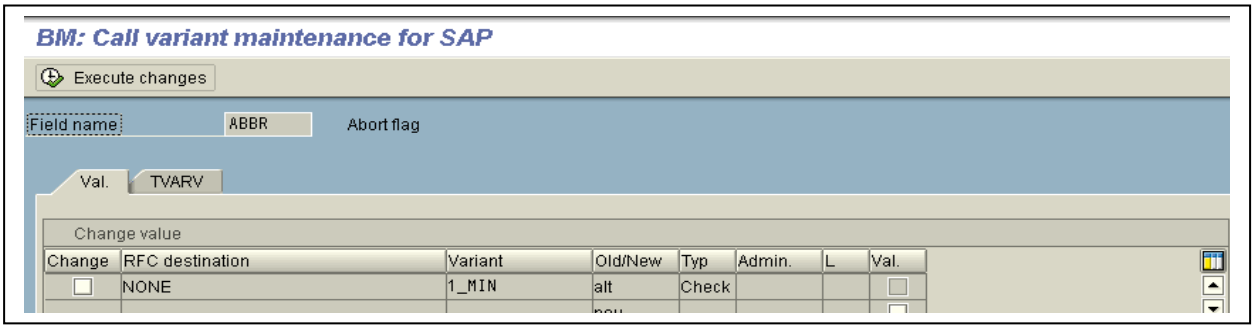


All buttons branch to the variant maintenance of the respective SAP system.

The exception is the “**Single field”** button. After you select one or all variants, the system reads the fields of the selection screen.

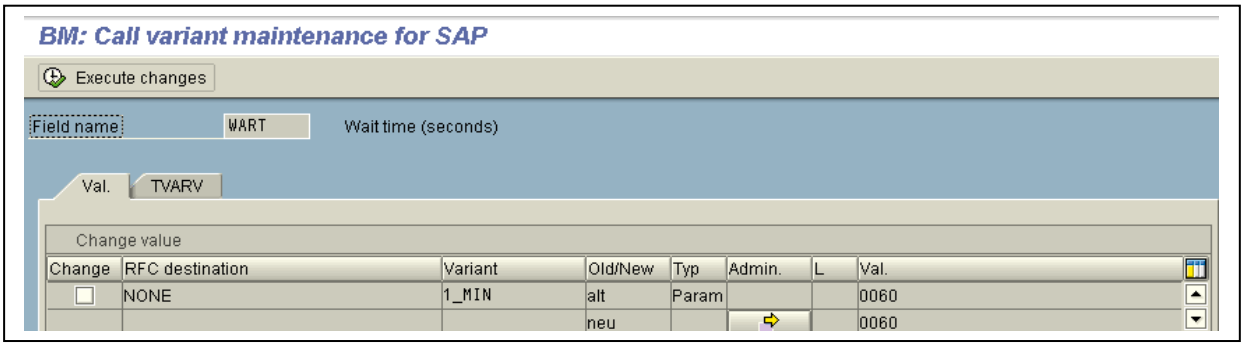


Here, a field name is selected; and the values can be changed via the buttons. Example with the field ‚**ABBR’** and ‚**Value Single Field’**.



Now the flag (because ‚ABBR’ is defined as checkbox in ABAP/4 Program) for all variants can be set.

Example with field name ‚**WART’** and ‚**Value single field’**.

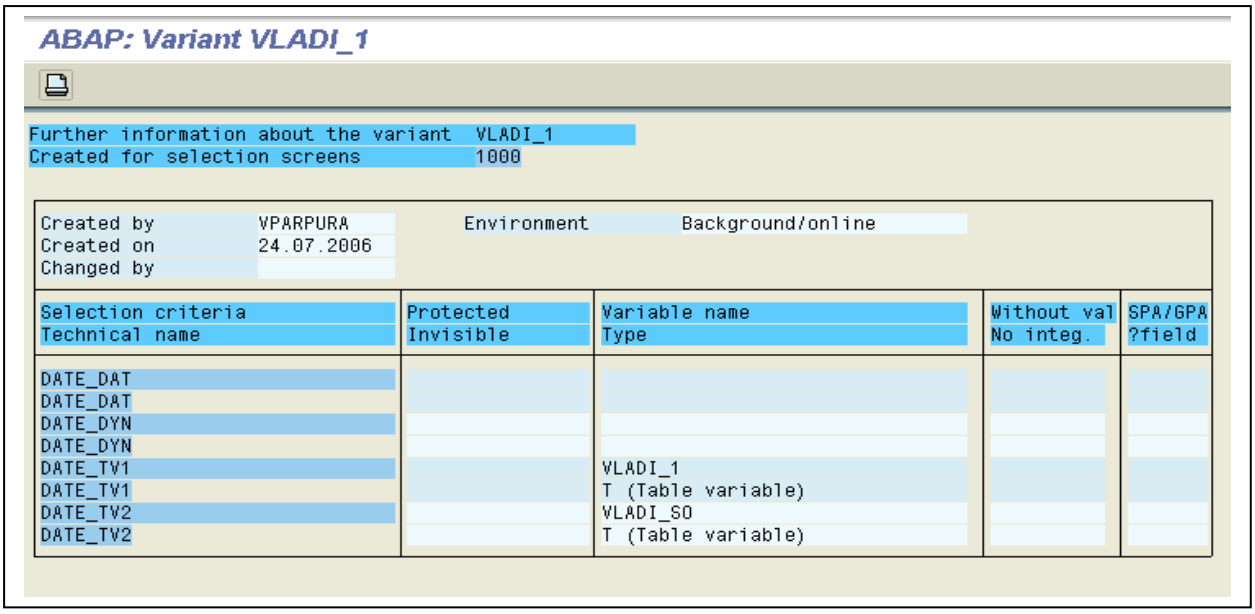


Here maintenance is not possible directly in the list. Using the button, you can call an input window where the values are to be entered. The system checks the values entered as to whether the values conform to the length and field type.

Using the “**Execute changes”** button, you can save/save again the variants with changed values.

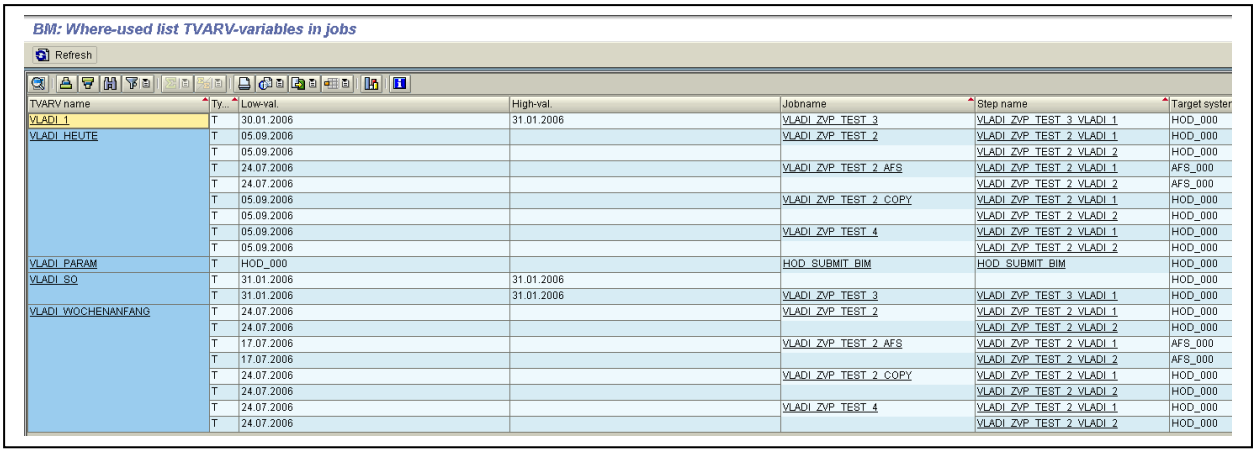
5.6. TVARV(C) – Variables

TVARV Variables are deposited in the characteristics of the variants in the ABAP programs so that variables can be supplied dynamically with values.

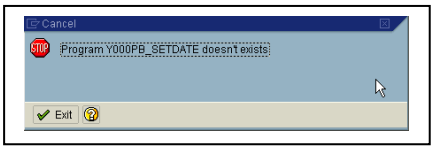


5.6.1. Where-Used-Lists Variables in Jobs

The system displays the variables included in the jobs. Variables may be dynamic date variables (type F) or TVARV variables (type T). The job names may be preselected.



All objects that can be maintained are defined as links. By clicking, you can branch directly to the maintenance of the object.

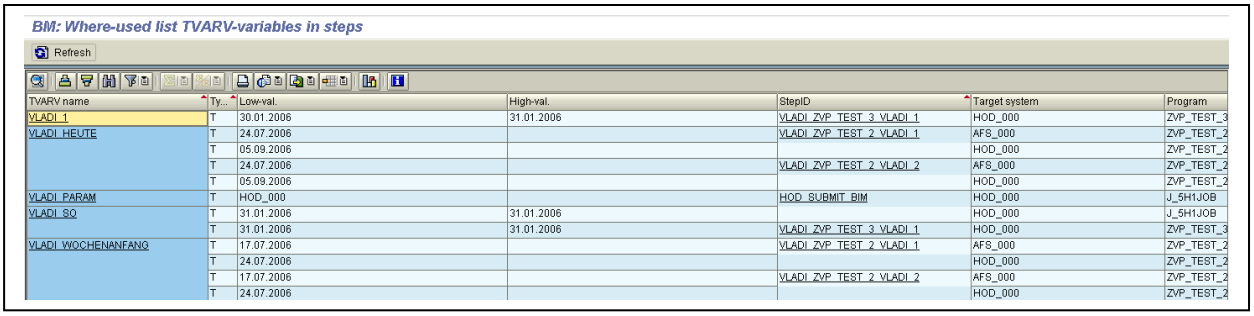


If the system displays this message (text was cut out here), you can continue with “Exit”.

For each program, the system checks whether it exists and has no errors. If not, the system displays these messages. The BatchMan program continues with the next program and does NOT end.

5.6.2. Where-Used List Variables in Steps

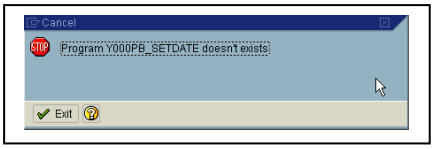
The system displays the variables included in the steps. Variables may be dynamic date variables (type F) or TVARV variables (type T). The step names may be preselected.



TVARV name	Ty...	Low-val.	High-val.	StepID	Target system	Program
VLADI_1	T	30.01.2006	31.01.2006	VLADI_ZVP_TEST_3_VLADI_1	HOD_000	ZVP_TEST_3
VLADI_HEUTE	T	24.07.2006		VLADI_ZVP_TEST_2_VLADI_1	AFS_000	ZVP_TEST_2
	T	05.09.2006			HOD_000	ZVP_TEST_2
	T	24.07.2006		VLADI_ZVP_TEST_2_VLADI_2	AFS_000	ZVP_TEST_2
	T	05.09.2006			HOD_000	ZVP_TEST_2
VLADI_PARAM	T	HOD_000		HOD_SUBMIT_BIM	HOD_000	J_5H1JOB
VLADI_SO	T	31.01.2006	31.01.2006		HOD_000	J_5H1JOB
	T	31.01.2006	31.01.2006	VLADI_ZVP_TEST_3_VLADI_1	HOD_000	ZVP_TEST_3
VLADI_WOCHENANFANG	T	17.07.2006		VLADI_ZVP_TEST_2_VLADI_1	AFS_000	ZVP_TEST_2
	T	24.07.2006			HOD_000	ZVP_TEST_2
	T	17.07.2006		VLADI_ZVP_TEST_2_VLADI_2	AFS_000	ZVP_TEST_2
	T	24.07.2006			HOD_000	ZVP_TEST_2

All objects that can be maintained are defined as links. By clicking, you can branch directly to the maintenance of the object.

In addition to the programs and variants, the system also displays here the field names and the verbal conversion of the field names (language dependent).



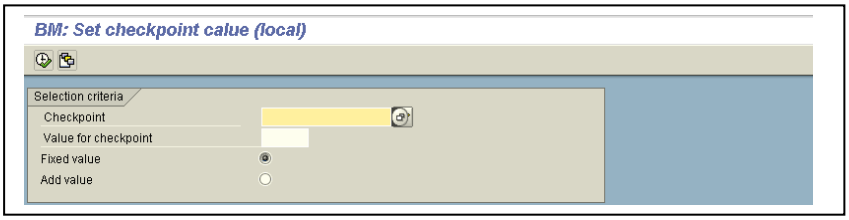
If the system displays this message (text was cut out here), you can continue with “Exit.”

For each program, the system checks whether it exists and has no errors. If not, the system displays these messages. The BatchMan program continues with the next program and does NOT end.

5.7. Checkpoints

5.7.1. Set checkpoints locally

An automatic set of local checkpoint value is realized with the report **/BTCMAN/M_KP2.** More than one checkpoint can be changed here.



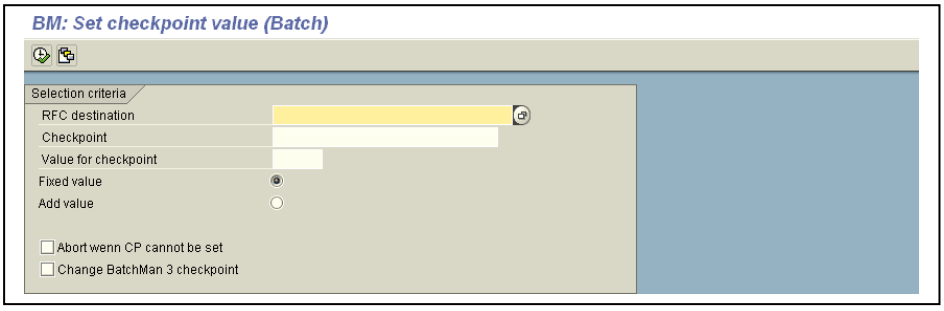
The field for the checkpoints is NOT required. If no checkpoint is entered, the system changes all checkpoints after a confirmation prompt.

5.7.2. Set checkpoints via RFC

An automatic set of checkpoint value is realized with the report **/BTCMAN/C_KP3**. Only one checkpoint can be changed here.

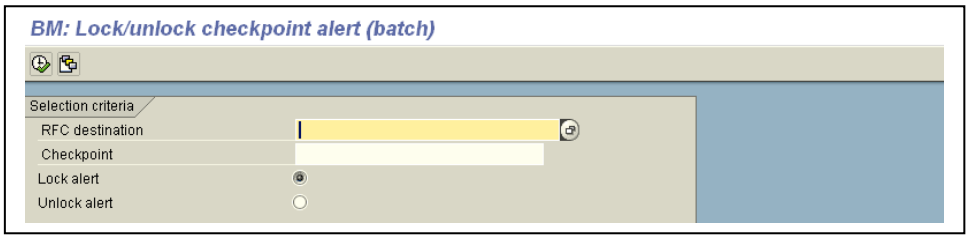
To set a checkpoint in another system, the RFC destination is maintained. For the local system, the RFC destination may also be specified. Or “NONE”’ is used if the field is empty.

Since the field for the checkpoints is required, you have to enter a value. You may enter * Then in the dialog mode, the system again queries (to make certain) whether all checkpoints should really be changed. If the “Add value” flag is activated, the system adds the value entered to the previous value of the checkpoint. If it is not activated, the system sets the value entered as fixed value. This setting is exactly opposite to the start criteria.



The report is then, for example, scheduled in a network as job.

5.7.3. Alarm for Checkpoints



An automatic lock or unlock is implemented with report **/BTCMAN/M_KP4**. Always, only one checkpoint should be changed. The alert at the checkpoint is then defined if the checkpoint is defined as a milestone.

5.7.4. Where-Used List Checkpoints

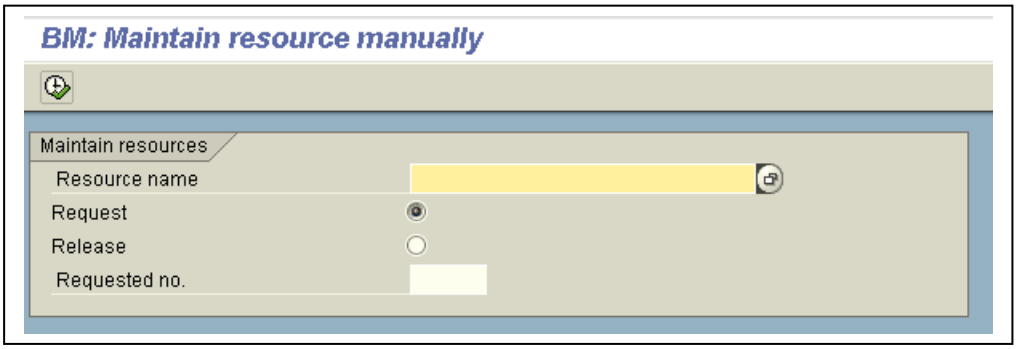
Execute the where-used list for checkpoints. This program is generally called from the related function to maintain checkpoints in Customizing. Executing this report delivers the usage of all checkpoints.

5.8. Ressources

5.8.1. Request/Release Resources per ABAP

Resources are requested manually with the report **/BTCMAN/M_RE1**. This prevents that processes that use the resource, can start running. This is necessary if resources should be blocked permanently, manually or externally over several jobs (except over complete networks).

In the table /BTCMAN/M_RESOBJ, the resources are registered with job name = MANUALLY and empty count. These entries are not released by the master job.



5.8.2. Where-Used List Resources

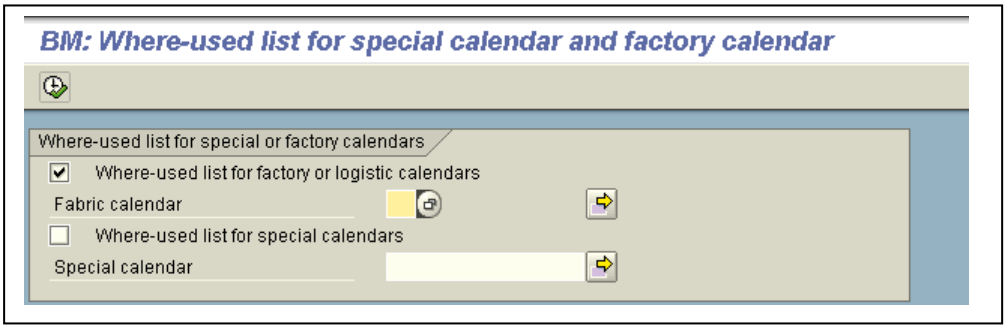
Execute the where-used list for resources. This program is generally called from the related function to maintain resources in Customizing. Executing this report delivers the usage of all resources.

Using the tab page, you can display the usage in jobs and networks. You can also see in which orders (jobs and networks) resources are used.

5.9. Calendar

Executing the where-used list for SAP factory-/ logistics calendars and BatchMan special calendars.

Usage of SAP – calendars in BatchMan objects.

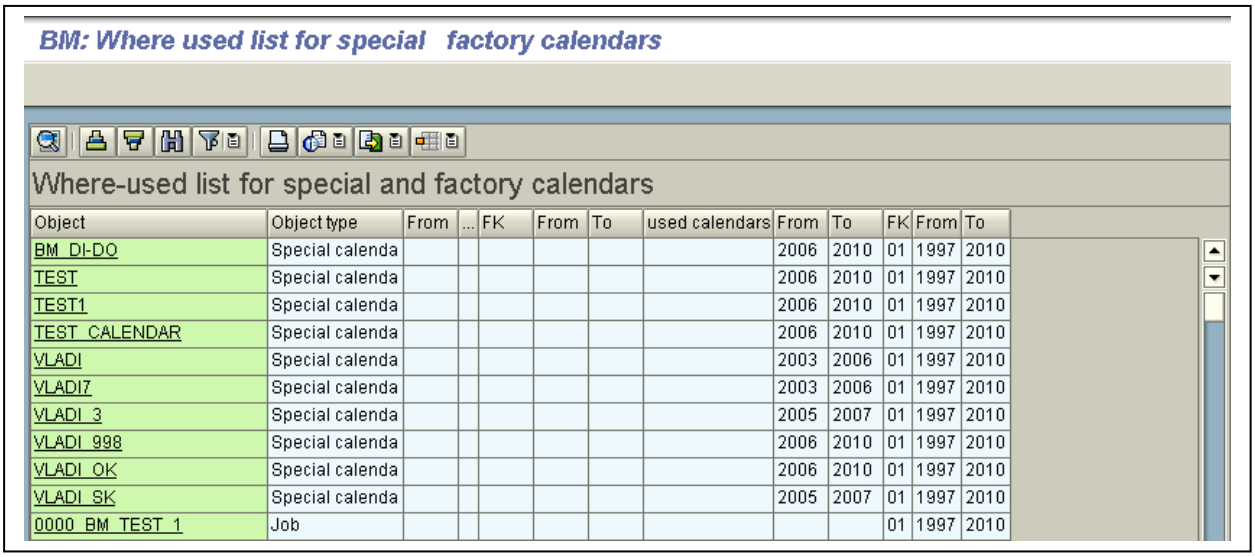


Delivers as result:

For the **BatchMan** special calendar, this program is called from the related functions of the checkpoint maintenance in Customizing Executing this report delivers the usage of all **BatchMan** special calendars.

5.10. Archiving/ Reorganization

The functions Archive orders (/BTCMAN/M_SAV), Restoring orders (/BTCMAN/M_RST) and Delete old archive files (/BTCMAN/M_SFD) are already described in section 3.9.



Object	Object type	From	...	FK	From	To	used calendars	From	To	FK	From	To
BM_DI-DO	Special calenda							2006	2010	01	1997	2010
TEST	Special calenda							2006	2010	01	1997	2010
TEST1	Special calenda							2006	2010	01	1997	2010
TEST_CALENDAR	Special calenda							2006	2010	01	1997	2010
VLADI	Special calenda							2003	2006	01	1997	2010
VLADI7	Special calenda							2003	2006	01	1997	2010
VLADI_3	Special calenda							2005	2007	01	1997	2010
VLADI_998	Special calenda							2006	2010	01	1997	2010
VLADI_OK	Special calenda							2006	2010	01	1997	2010
VLADI_SK	Special calenda							2005	2007	01	1997	2010
0000_BM_TEST_1	Job									01	1997	2010

5.11. Runtime data

5.11.1. Couting table – Job/network schedulings

See reporting about network or job.

5.11.2. Runtime histogram (number of runs)

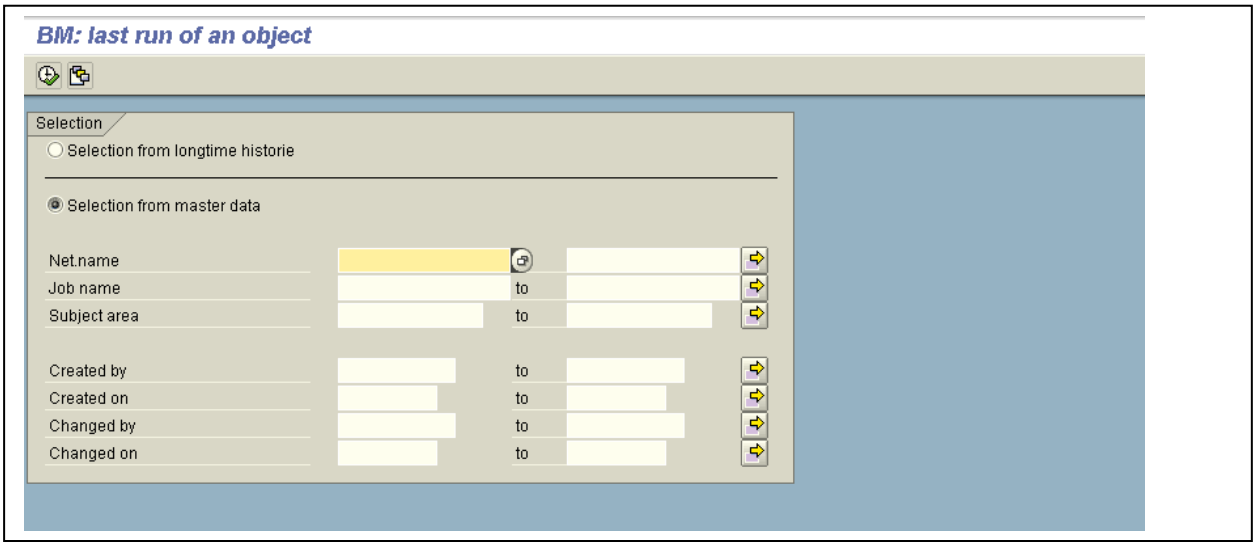
See reporting about network or job.

5.11.3. Runtime jistogram (date)

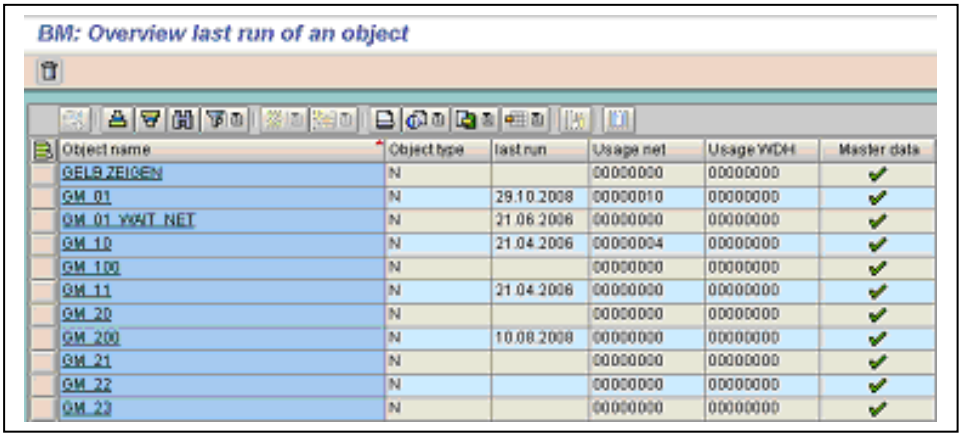
See reporting about network or job.

5.11.4. Display of the last runs

The program delivers a list of the last schedulings (runs) of an object (network or job). You can make the selection via the long term history or the master data.



Results list:

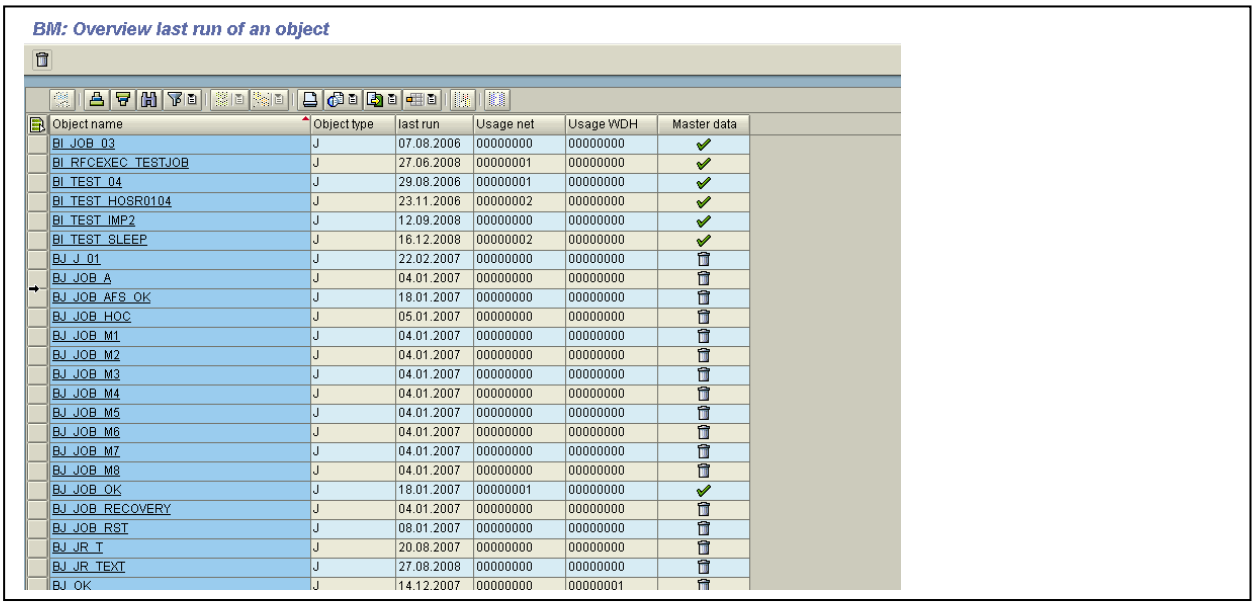


Object name	Object type	last run	Usage net	Usage WDH	Master data
GELB ZEIGEN	N		00000000	00000000	✔
GM_01	N	29.10.2008	00000010	00000000	✔
GM_01_WAIT_NET	N	21.06.2006	00000000	00000000	✔
GM_10	N	21.04.2006	00000004	00000000	✔
GM_100	N		00000000	00000000	✔
GM_11	N	21.04.2006	00000000	00000000	✔
GM_20	N		00000000	00000000	✔
GM_200	N	10.08.2008	00000000	00000000	✔
GM_21	N		00000000	00000000	✔
GM_22	N		00000000	00000000	✔
GM_23	N		00000000	00000000	✔

The green checkmark in the master data column indicates that the master data is available. Since selection was made via the master data, it should always apply.

By marking one or several lines, you can delete the objects directly from the list. Caution: Master data will be deleted.

This is the results list for the selection from the long term history:



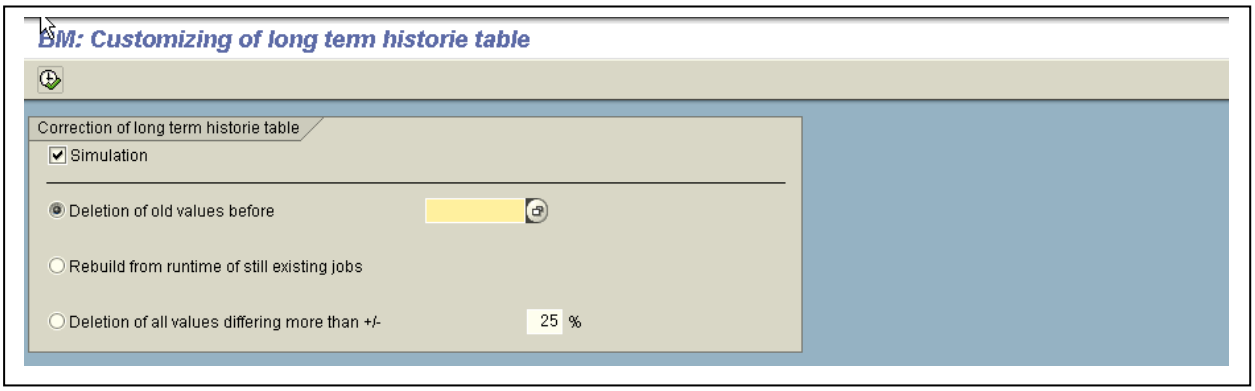
Object name	Object type	last run	Usage net	Usage WDH	Master data
BI_JOB_03	J	07.08.2006	00000000	00000000	✔
BI_RFCEXEC_TESTJOB	J	27.06.2008	00000001	00000000	✔
BI_TEST_04	J	29.08.2006	00000001	00000000	✔
BI_TEST_HOSR0104	J	23.11.2006	00000002	00000000	✔
BI_TEST_IMP2	J	12.09.2008	00000000	00000000	✔
BI_TEST_SLEEP	J	16.12.2008	00000002	00000000	✔
BJ_J_01	J	22.02.2007	00000000	00000000	🗑
BJ_JOB_A	J	04.01.2007	00000000	00000000	🗑
BJ_JOB_AFS_OK	J	18.01.2007	00000000	00000000	🗑
BJ_JOB_HOC	J	05.01.2007	00000000	00000000	🗑
BJ_JOB_M1	J	04.01.2007	00000000	00000000	🗑
BJ_JOB_M2	J	04.01.2007	00000000	00000000	🗑
BJ_JOB_M3	J	04.01.2007	00000000	00000000	🗑
BJ_JOB_M4	J	04.01.2007	00000000	00000000	🗑
BJ_JOB_M5	J	04.01.2007	00000000	00000000	🗑
BJ_JOB_M6	J	04.01.2007	00000000	00000000	🗑
BJ_JOB_M7	J	04.01.2007	00000000	00000000	🗑
BJ_JOB_M8	J	04.01.2007	00000000	00000000	🗑
BJ_JOB_OK	J	18.01.2007	00000001	00000000	✔
BJ_JOB_RECOVERY	J	04.01.2007	00000000	00000000	🗑
BJ_JOB_RST	J	08.01.2007	00000000	00000000	🗑
BJ_JR_T	J	20.08.2007	00000000	00000000	🗑
BJ_JR_TEXT	J	27.08.2008	00000000	00000000	🗑
BJ_OK	J	14.12.2007	00000000	00000001	🗑

The green checkmark in the master data column indicates that the master data is available. If the trashcan is displayed, only data from the history still exists. By marking one or several lines, you can execute the history of an object directly from the list. The master data is retained.

5.11.5. Cleaning up the long term history

In table /BTCMAN/M_RUNCNT, the data for the long term history is saved. Over time, the number of entries saved for a given object grows there, whereby the objects are compressed. For several runs a day, an object is only saved once.

With this program, the table can be reorganized and adjusted.



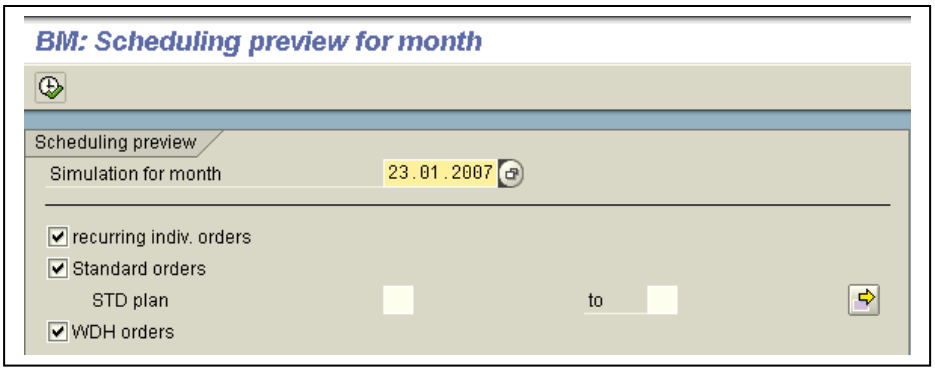
The table only has to be adjusted if major deviations arise due to changes. This may be due to changed processes (program XYZ instead of ABC), an increase of the selections in the variants (five company codes instead of one) or an increase of the data volume.

The execution need not occur daily, weekly or monthly, but may occur as required.

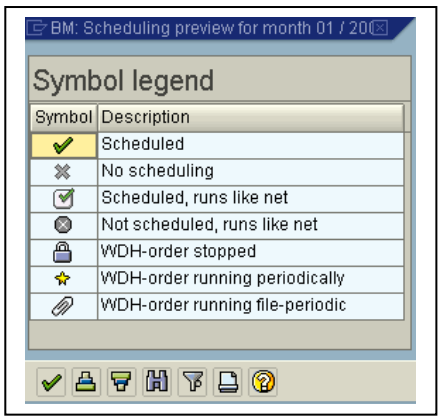
5.12. System Load

5.12.1. Scheduling Overview for one Month

For the month selected, this report (/BTCMAN/M_EP1) determines which networks are to run with which jobs and which individual jobs on which days in which orders. The selection may occur only by the plans for individual order types and especially for the STD orders. The default day is the first of the current month.

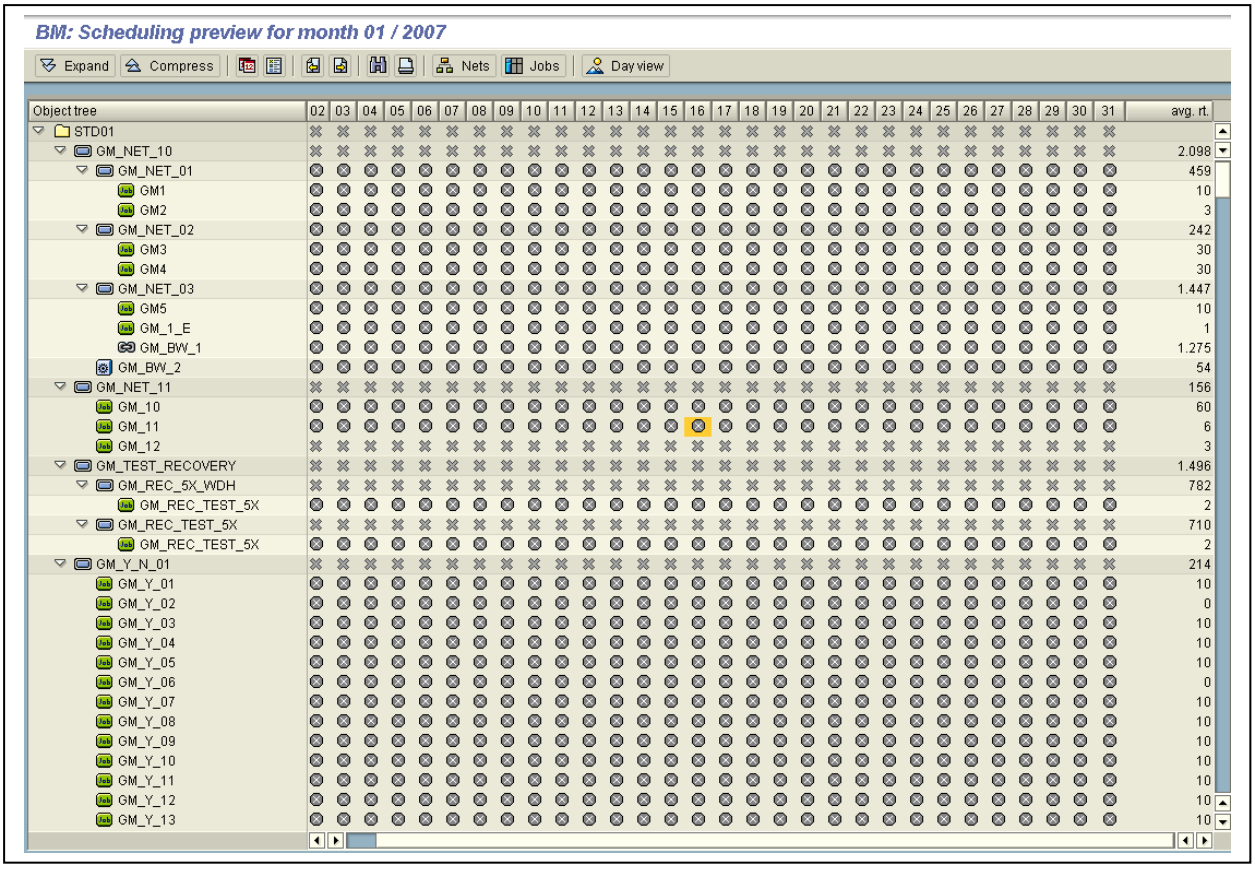


The legend is displayed via **Legend.**

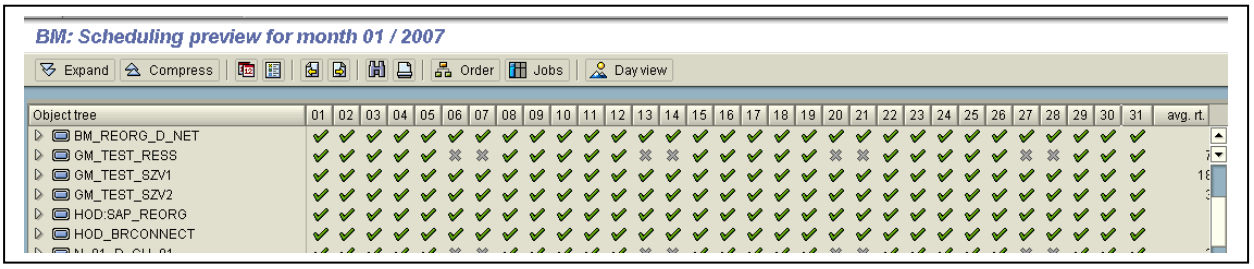


If you place the cursor in a day column, you can use the “**Day view”** button to branch to a more detailed view (scheduling overview for one day) with the start criteria and all steps for the jobs.

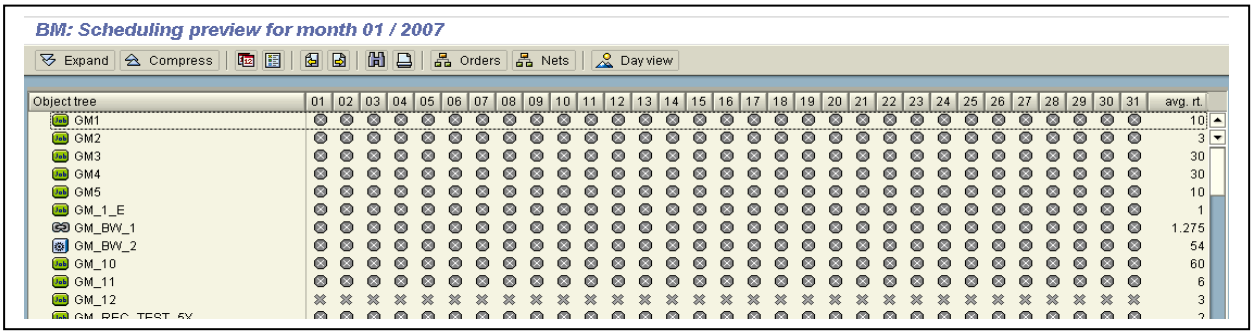
The lists are completely expanded by default. By double-clicking, you can branch to the master data maintenance of the objects.



In the **Network** view, only those networks are displayed that contain the days on which they are scheduled (without orders).



In the **Job** view, only those jobs are displayed that contain the days on which they are scheduled (without orders and networks).



5.12.2. Scheduling Overview for one Day

The report /BTCMAN/M_EP2 produces a detailed overview of all objects that should be executed on the selected day. The selection can occur for individual order types, and especially for the STD orders, according to the plans.

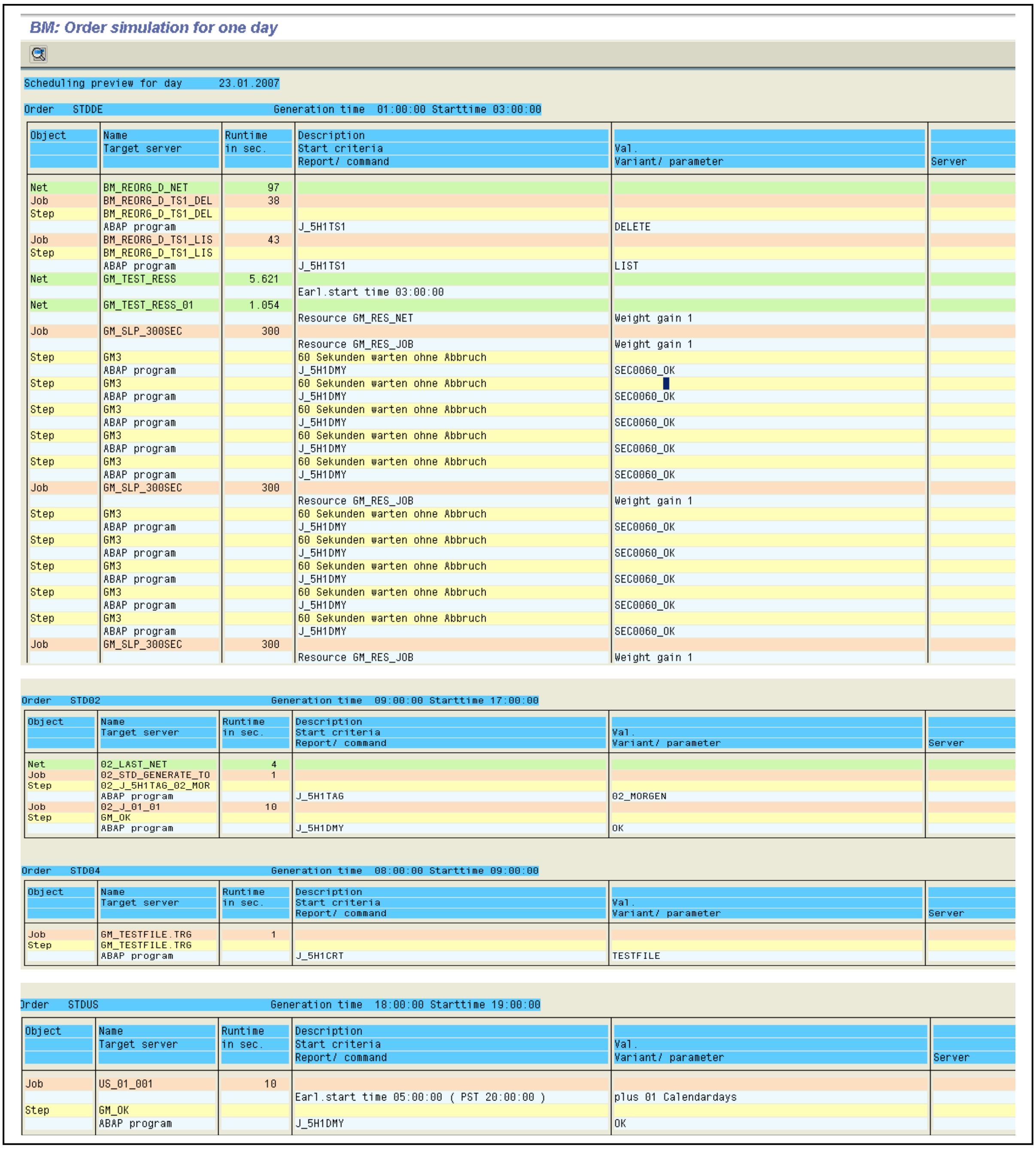
STD – orders Single/individual jobs with steps+variants,

Networks with the contained jobs including steps+variants

Start criteria of networks +jobs

INDIV – orders with the objects

WDH* – orders with the objects and the start criteria



BM: Order simulation for one day

Scheduling preview for day 23.01.2007

Order STDDE Generation time 01:00:00 Starttime 03:00:00

Object	Name Target server	Runtime in sec.	Description Start criteria Report/ command	Val. Variant/ parameter	Server
Net	BM_REORG_D_NET	97			
Job	BM_REORG_D_TS1_DEL	38			
Step	BM_REORG_D_TS1_DEL				
	ABAP program		J_5H1TS1	DELETE	
Job	BM_REORG_D_TS1_LIS	43			
Step	BM_REORG_D_TS1_LIS				
	ABAP program		J_5H1TS1	LIST	
Net	GM_TEST_RESS	5.621			
			Earl.start time 03:00:00		
Net	GM_TEST_RESS_01	1.054			
			Resource GM_RES_NET	Weight gain 1	
Job	GM_SLP_300SEC	300			
			Resource GM_RES_JOB	Weight gain 1	
Step	GM3		60 Sekunden warten ohne Abbruch		
	ABAP program		J_5H1DMY	SEC0060_OK	
Step	GM3		60 Sekunden warten ohne Abbruch		
	ABAP program		J_5H1DMY	SEC0060_OK	
Step	GM3		60 Sekunden warten ohne Abbruch		
	ABAP program		J_5H1DMY	SEC0060_OK	
Step	GM3		60 Sekunden warten ohne Abbruch		
	ABAP program		J_5H1DMY	SEC0060_OK	
Step	GM3		60 Sekunden warten ohne Abbruch		
	ABAP program		J_5H1DMY	SEC0060_OK	
Job	GM_SLP_300SEC	300			
			Resource GM_RES_JOB	Weight gain 1	
Step	GM3		60 Sekunden warten ohne Abbruch		
	ABAP program		J_5H1DMY	SEC0060_OK	
Step	GM3		60 Sekunden warten ohne Abbruch		
	ABAP program		J_5H1DMY	SEC0060_OK	
Step	GM3		60 Sekunden warten ohne Abbruch		
	ABAP program		J_5H1DMY	SEC0060_OK	
Step	GM3		60 Sekunden warten ohne Abbruch		
	ABAP program		J_5H1DMY	SEC0060_OK	
Step	GM3		60 Sekunden warten ohne Abbruch		
	ABAP program		J_5H1DMY	SEC0060_OK	
Job	GM_SLP_300SEC	300			
			Resource GM_RES_JOB	Weight gain 1	

Order STD02 Generation time 09:00:00 Starttime 17:00:00

Object	Name Target server	Runtime in sec.	Description Start criteria Report/ command	Val. Variant/ parameter	Server
Net	02_LAST_NET	4			
Job	02_STD_GENERATE_TO	1			
Step	02_J_5H1TAG_02_MOR				
	ABAP program		J_5H1TAG	02_MORGEN	
Job	02_J_01_01	10			
Step	GM_OK				
	ABAP program		J_5H1DMY	OK	

Order STD04 Generation time 08:00:00 Starttime 09:00:00

Object	Name Target server	Runtime in sec.	Description Start criteria Report/ command	Val. Variant/ parameter	Server
Job	GM_TESTFILE.TRG	1			
Step	GM_TESTFILE.TRG				
	ABAP program		J_5H1CRT	TESTFILE	

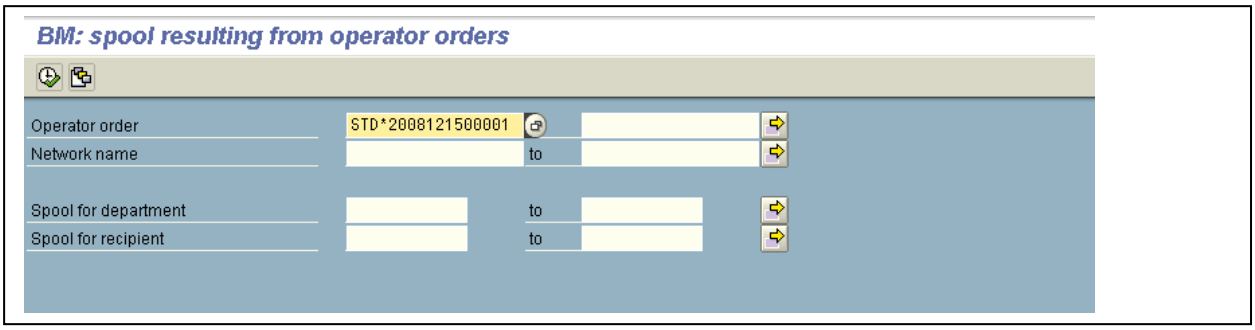
Order STDUS Generation time 18:00:00 Starttime 19:00:00

Object	Name Target server	Runtime in sec.	Description Start criteria Report/ command	Val. Variant/ parameter	Server
Job	US_01_001	10			
			Earl.start time 05:00:00 (PST 20:00:00)	plus 01 Calendardays	
Step	GM_OK				
	ABAP program		J_5H1DMY	OK	

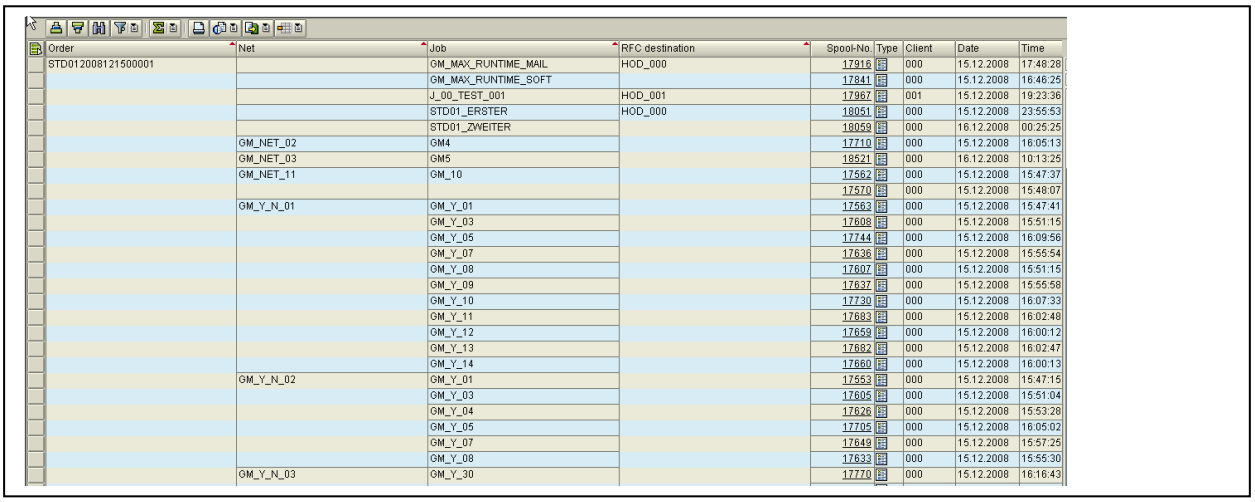
5.13. Print orders

5.13.1. Spool for orders

This program provides an overview of which spool lists were generated in the orders selected. Using the printed attributes of such a spool order, it is simple to continue the administration of these using the standard transactions for spool administration. The spool order number is defined as a link. That makes it possible to branch directly to SP01.



Result list:

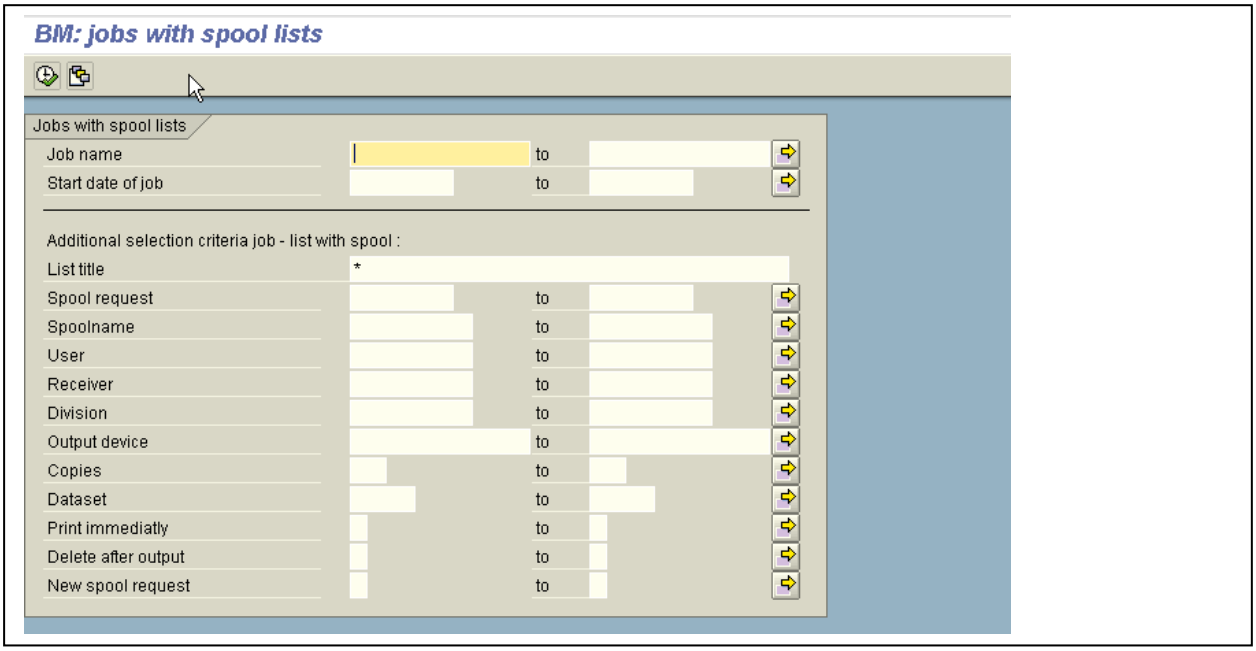


Order	Net	Job	RFC destination	Spool-No.	Type	Client	Date	Time
STD012008121500001		GM_MAX_RUNTIME_MAIL	HOD_000	17916		000	15.12.2008	17:48:28
		GM_MAX_RUNTIME_SOFT		17841		000	15.12.2008	16:46:25
		J_00_TEST_001	HOD_001	17967		001	15.12.2008	19:23:36
		STD01_ERSTER	HOD_000	18051		000	15.12.2008	23:55:53
		STD01_ZWEITER		18059		000	16.12.2008	00:25:25
	GM_NET_02	GM4		17710		000	15.12.2008	16:05:13
	GM_NET_03	GM5		18521		000	16.12.2008	10:13:25
	GM_NET_11	GM_10		17562		000	15.12.2008	15:47:37
				17570		000	15.12.2008	15:48:07
	GM_Y_N_01	GM_Y_01		17563		000	15.12.2008	15:47:41
		GM_Y_03		17608		000	15.12.2008	15:51:15
		GM_Y_05		17744		000	15.12.2008	16:09:56
		GM_Y_07		17636		000	15.12.2008	15:55:54
		GM_Y_08		17607		000	15.12.2008	15:51:15
		GM_Y_09		17637		000	15.12.2008	15:55:58
		GM_Y_10		17730		000	15.12.2008	16:07:33
		GM_Y_11		17683		000	15.12.2008	16:02:48
		GM_Y_12		17659		000	15.12.2008	16:00:12
		GM_Y_13		17682		000	15.12.2008	16:02:47
		GM_Y_14		17660		000	15.12.2008	16:00:13
	GM_Y_N_02	GM_Y_01		17553		000	15.12.2008	15:47:15
		GM_Y_03		17605		000	15.12.2008	15:51:04
		GM_Y_04		17626		000	15.12.2008	15:53:28
		GM_Y_05		17705		000	15.12.2008	16:05:02
		GM_Y_07		17649		000	15.12.2008	15:57:25
		GM_Y_08		17633		000	15.12.2008	15:55:30
	GM_Y_N_03	GM_Y_30		17770		000	15.12.2008	16:16:43

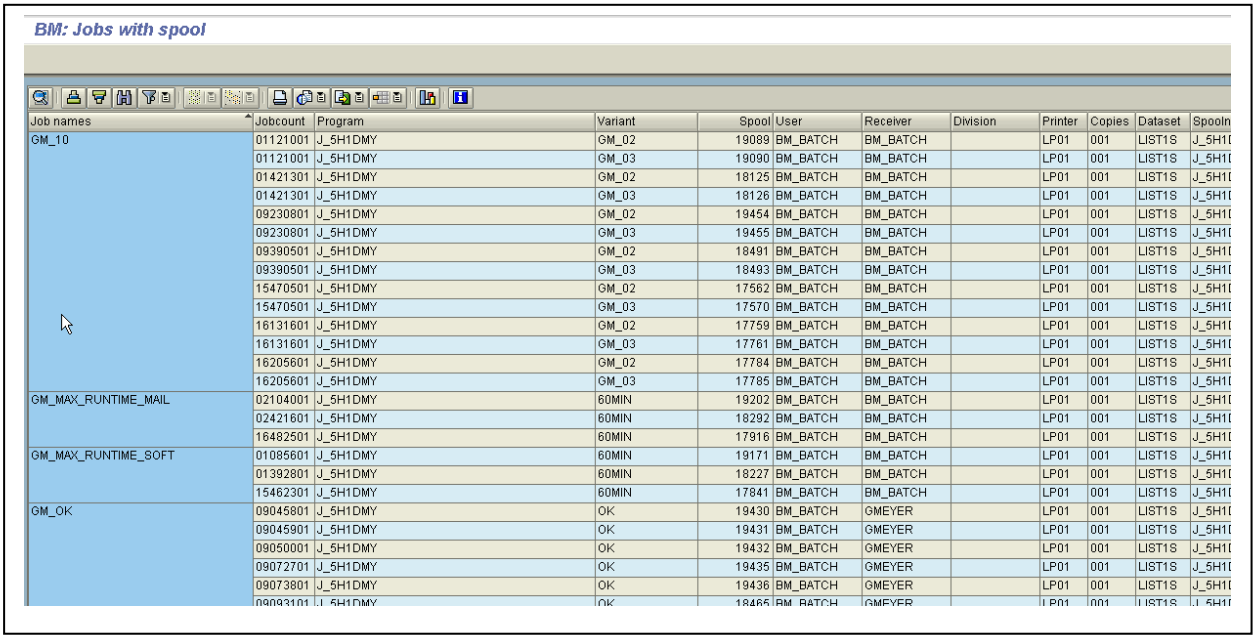
This program is used internally in the order monitor.

5.13.2. Jobs with spool lists

This program provides an overview of which jobs have spool lists in the system. Using the printed attributes of such a spool order, it is simple to continue the administration of these using the standard transactions for spool administration.



Result list:



Job names	Jobcount	Program	Variant	Spool	User	Receiver	Division	Printer	Copies	Dataset	Spooln
GM_10	01121001	J_5H1DMY	GM_02	19089	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	01121001	J_5H1DMY	GM_03	19090	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	01421301	J_5H1DMY	GM_02	18125	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	01421301	J_5H1DMY	GM_03	18126	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	09230801	J_5H1DMY	GM_02	19454	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	09230801	J_5H1DMY	GM_03	19455	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	09390501	J_5H1DMY	GM_02	18491	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	09390501	J_5H1DMY	GM_03	18493	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	15470501	J_5H1DMY	GM_02	17562	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	15470501	J_5H1DMY	GM_03	17570	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	16131601	J_5H1DMY	GM_02	17759	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	16131601	J_5H1DMY	GM_03	17761	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	16205601	J_5H1DMY	GM_02	17784	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	16205601	J_5H1DMY	GM_03	17785	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
GM_MAX_RUNTIME_MAIL	02104001	J_5H1DMY	60MIN	19202	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	02421601	J_5H1DMY	60MIN	18292	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	16482501	J_5H1DMY	60MIN	17916	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
GM_MAX_RUNTIME_SOFT	01085601	J_5H1DMY	60MIN	19171	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	01392801	J_5H1DMY	60MIN	18227	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
	15462301	J_5H1DMY	60MIN	17841	BM_BATCH	BM_BATCH		LP01	001	LIST1S	J_5H1
GM_OK	09045801	J_5H1DMY	OK	19430	BM_BATCH	GMEYER		LP01	001	LIST1S	J_5H1
	09045901	J_5H1DMY	OK	19431	BM_BATCH	GMEYER		LP01	001	LIST1S	J_5H1
	09050001	J_5H1DMY	OK	19432	BM_BATCH	GMEYER		LP01	001	LIST1S	J_5H1
	09072701	J_5H1DMY	OK	19435	BM_BATCH	GMEYER		LP01	001	LIST1S	J_5H1
	09073801	J_5H1DMY	OK	19436	BM_BATCH	GMEYER		LP01	001	LIST1S	J_5H1
	09093101	J_5H1DMY	OK	18466	BM_BATCH	GMEYER		LP01	001	LIST1S	J_5H1

5.14. Consistency Check

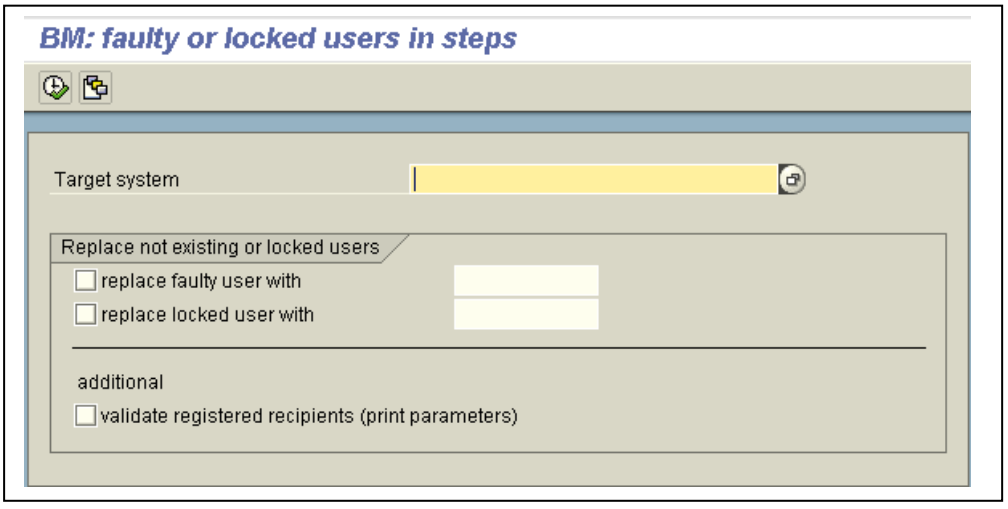
5.14.1. Check Alert Configuration

With this function you can test an installed alert tool configuration. Detailed explanations can be found in BM_e_AlertHB.pdf.

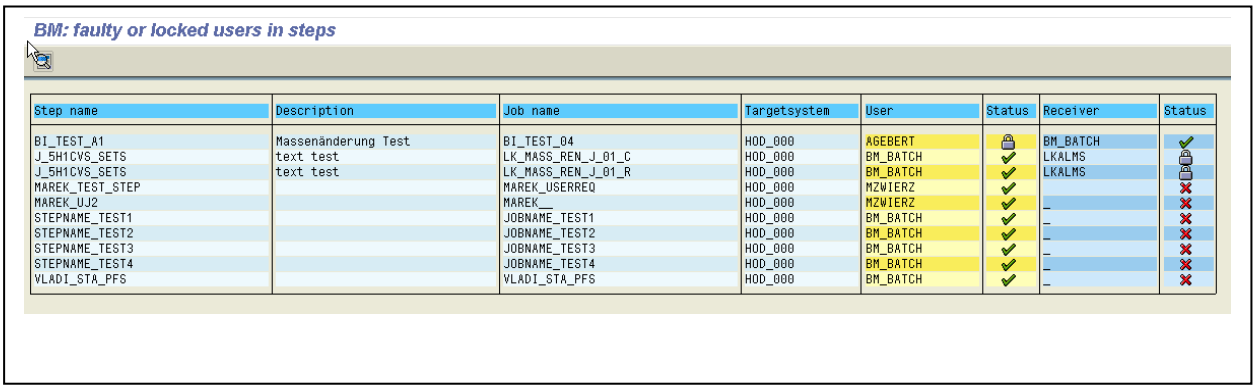
5.14.2. Faulty and Locked User in Steps

This program (/BTCMAN/M_BSJ) checks the batch users and, upon request, the print receivers that are used in the individual steps. The system only checks those steps that are assigned to a job. The user validation occurs remotely for the individual target systems of the relevant job.

If the relevant user IDs are not available or if the user is locked, the system replaces these upon request with the DEFAULT user form the selection screen. The system issues an error log, which enables a direct branch to the individual step or job maintenance dialogs.



The results list displays the users who do not exist as a step user or as receiver in the print parameters of a step.



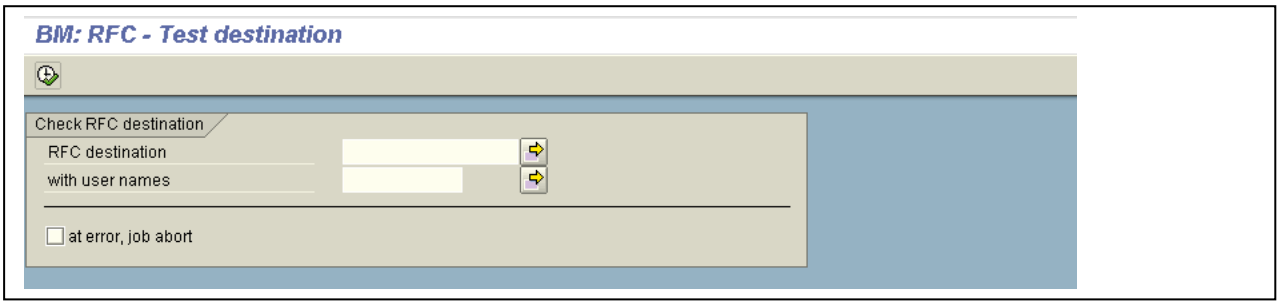
Step name	Description	Job name	Targetsystem	User	Status	Receiver	Status
BI_TEST_A1	Massenänderung Test	BI_TEST_04	HOD_000	AGEBERT		BM_BATCH	
J_5H1CVS_SETS	text test	LK_MASS_REN_J_01_C	HOD_000	BM_BATCH		LKALMS	
J_5H1CVS_SETS	text test	LK_MASS_REN_J_01_R	HOD_000	BM_BATCH		LKALMS	
MAREK_TEST_STEP		MAREK_USERREQ	HOD_000	MZWIERZ			
MAREK_UJ2		MAREK__	HOD_000	MZWIERZ		_	
STEPNAME_TEST1		JOBNAME_TEST1	HOD_000	BM_BATCH		_	
STEPNAME_TEST2		JOBNAME_TEST2	HOD_000	BM_BATCH		_	
STEPNAME_TEST3		JOBNAME_TEST3	HOD_000	BM_BATCH		_	
STEPNAME_TEST4		JOBNAME_TEST4	HOD_000	BM_BATCH		_	
VLADI_STA_PFS		VLADI_STA_PFS	HOD_000	BM_BATCH		_	

Users who do not exist are indicated by an 

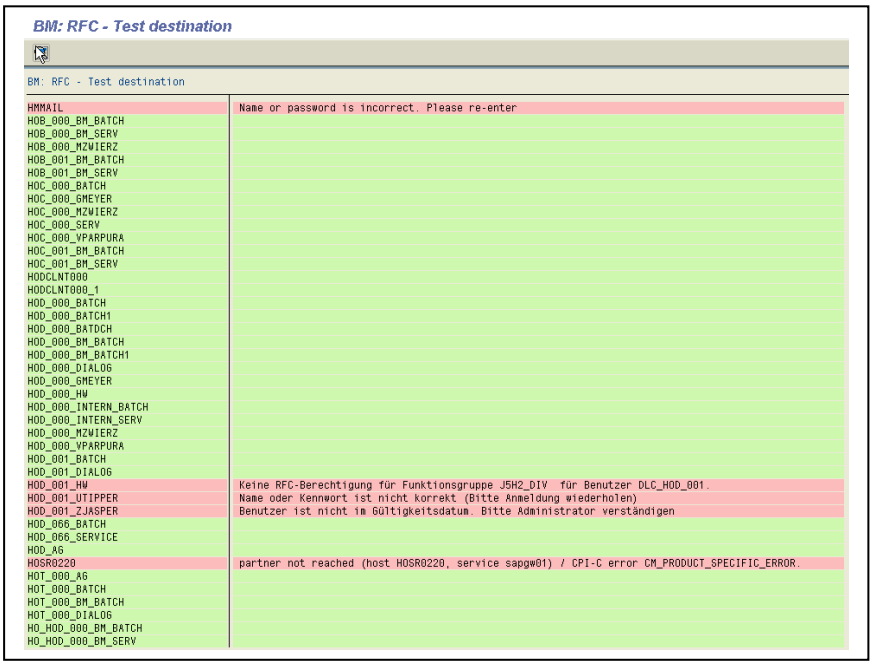
5.14.3. Check RFC connections

The program may be used to check the RFC connections. The system reads all RFC connections of the type “3” (R/3 connections) and “L”’ (logical) from transaction SM59 that a user entered in his/her connection.

For the connections found, a remote login test is executed like for SM59. That way, this function is also available for logical connections.



Result list:



The system also checks connections for transport control and for the system landscape. The check may be restricted via the users, e.g. to the user BM_BATCH used in the installation manual.

In the results list, you can see that not all components of the client transports are installed on separate systems. The language of the output depends on the language defined for the connection.

5.15. Miscellaneous Information

5.15.1. Memory Usage / Sizing

This report (/BTCMAN/M_SIZ) determines the current memory usage of BatchMan tables. Sentence length and sentence number are the basis for the byte count.

The produced values do not correspond to the real values in the data base. These are dependent on the data base organization (e.g. Oracle: Table spaces).

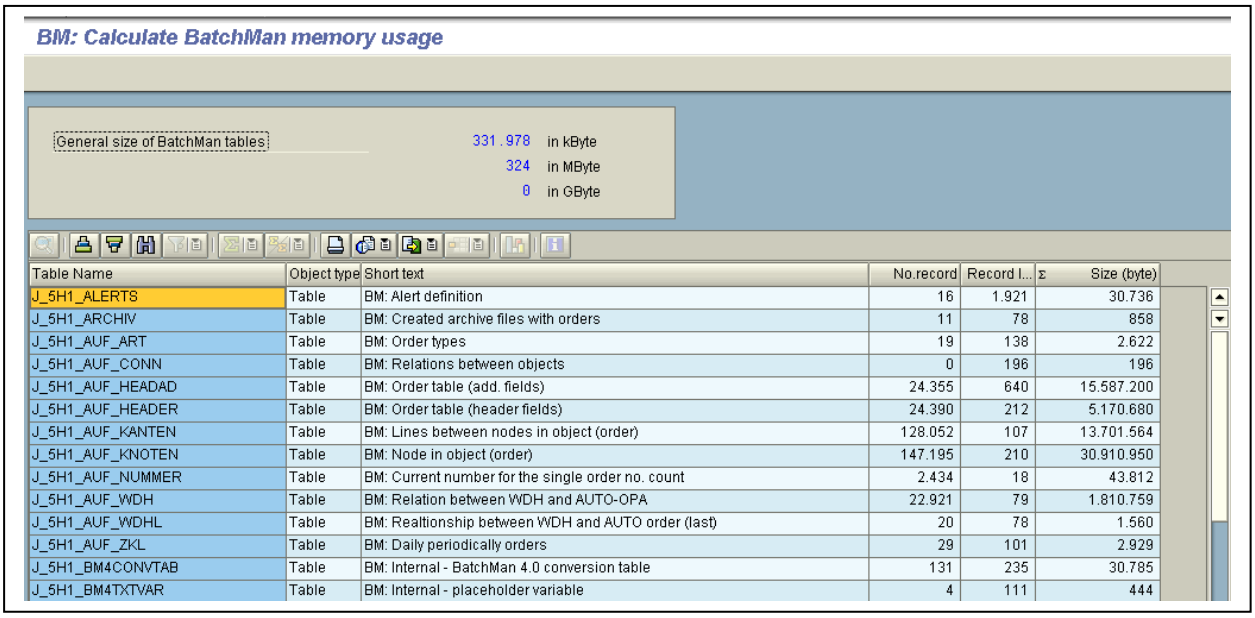
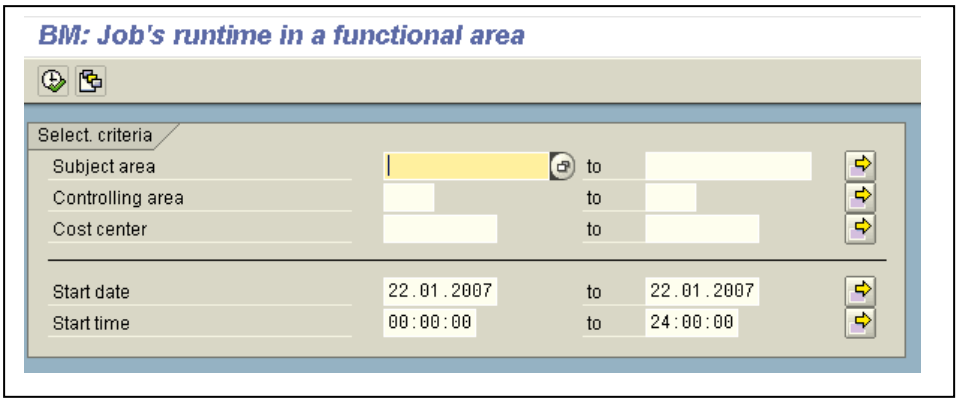


Table Name	Object type	Short text	No.record	Record l...	Σ	Size (byte)
J_5H1_ALERTS	Table	BM: Alert definition	16	1.921		30.736
J_5H1_ARCHIV	Table	BM: Created archive files with orders	11	78		858
J_5H1_AUF_ART	Table	BM: Order types	19	138		2.622
J_5H1_AUF_CONN	Table	BM: Relations between objects	0	196		196
J_5H1_AUF_HEADAD	Table	BM: Order table (add. fields)	24.355	640		15.587.200
J_5H1_AUF_HEADER	Table	BM: Order table (header fields)	24.390	212		5.170.680
J_5H1_AUF_KANTEN	Table	BM: Lines between nodes in object (order)	128.052	107		13.701.564
J_5H1_AUF_KNOTEN	Table	BM: Node in object (order)	147.195	210		30.910.950
J_5H1_AUF_NUMMER	Table	BM: Current number for the single order no. count	2.434	18		43.812
J_5H1_AUF_WDH	Table	BM: Relation between WDH and AUTO-OPA	22.921	79		1.810.759
J_5H1_AUF_WDHL	Table	BM: Realtionship between WDH and AUTO order (last)	20	78		1.560
J_5H1_AUF_ZKL	Table	BM: Daily periodically orders	29	101		2.929
J_5H1_BM4CONVTAB	Table	BM: Internal - BatchMan 4.0 conversion table	131	235		30.785
J_5H1_BM4TXTVAR	Table	BM: Internal - placeholder variable	4	111		444

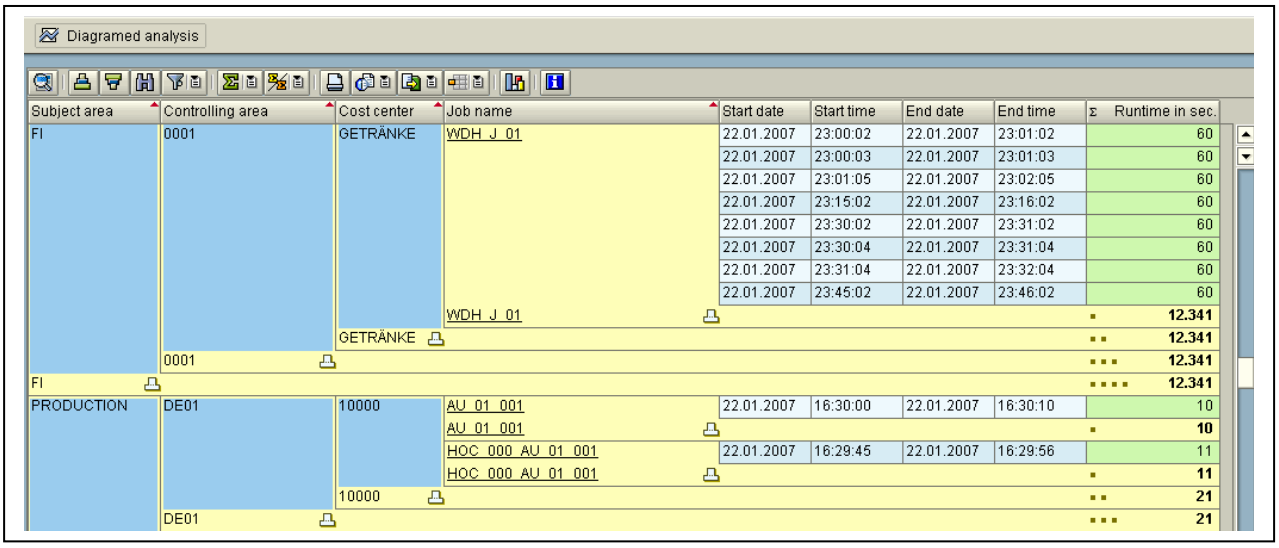
The upper section shows the total bytes. The table below shows the values for each individual BatchMan table.

5.15.2. Allocation Job Run Time to Subject Area/Cost Center

This program (/BTCMAN/M_ACC) shows how easy an accounting report can be produced based on the data collected by BatchMan. Listed are the start- and end times of the jobs and the derived calculated runtime in seconds. A cost center allocation is done based on the subject areas that were maintained in the job master data.

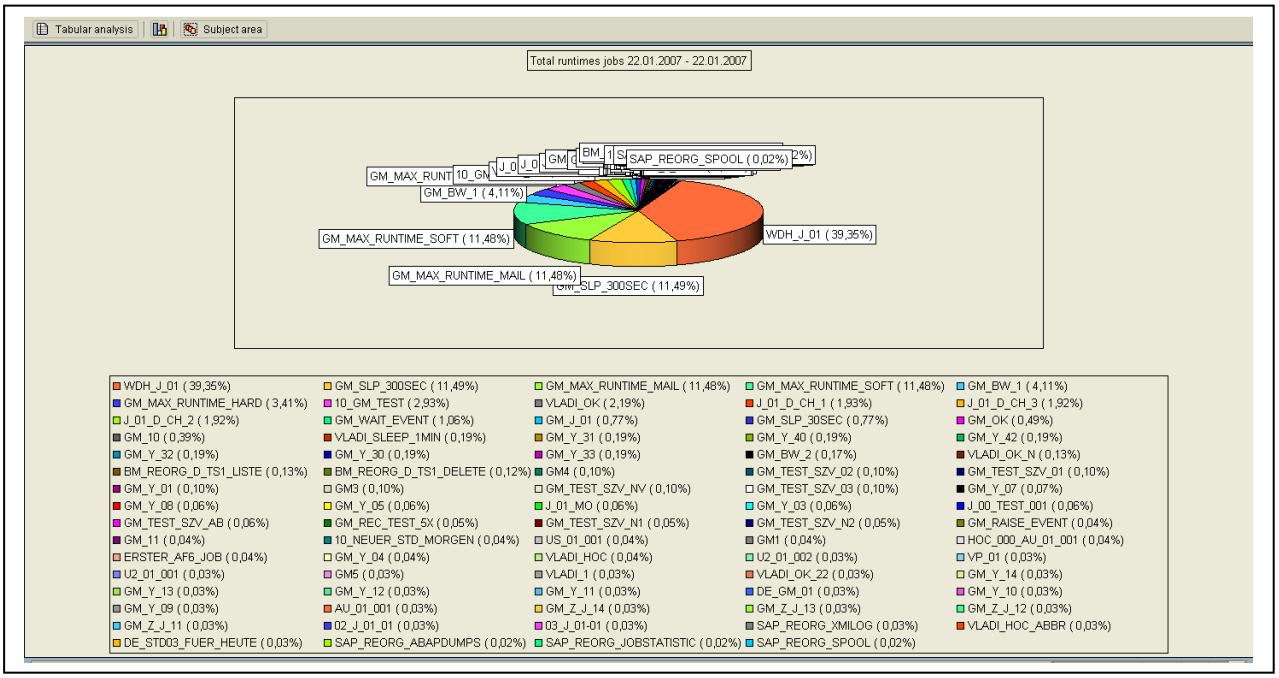


List display



Subject area	Controlling area	Cost center	Job name	Start date	Start time	End date	End time	Σ Runtime in sec.
FI	0001	GETRÄNKE	WDH_J_01	22.01.2007	23:00:02	22.01.2007	23:01:02	60
				22.01.2007	23:00:03	22.01.2007	23:01:03	60
				22.01.2007	23:01:05	22.01.2007	23:02:05	60
				22.01.2007	23:15:02	22.01.2007	23:16:02	60
				22.01.2007	23:30:02	22.01.2007	23:31:02	60
				22.01.2007	23:30:04	22.01.2007	23:31:04	60
				22.01.2007	23:31:04	22.01.2007	23:32:04	60
				22.01.2007	23:45:02	22.01.2007	23:46:02	60
			WDH_J_01					12.341
		GETRÄNKE						12.341
	0001							12.341
FI								12.341
PRODUCTION	DE01	10000	AU_01_001	22.01.2007	16:30:00	22.01.2007	16:30:10	10
			AU_01_001					10
			HOC_000_AU_01_001	22.01.2007	16:29:45	22.01.2007	16:29:56	11
			HOC_000_AU_01_001					11
		10000						21
	DE01							21

Additional displays can be generated via ‚Diagram Analysis'.



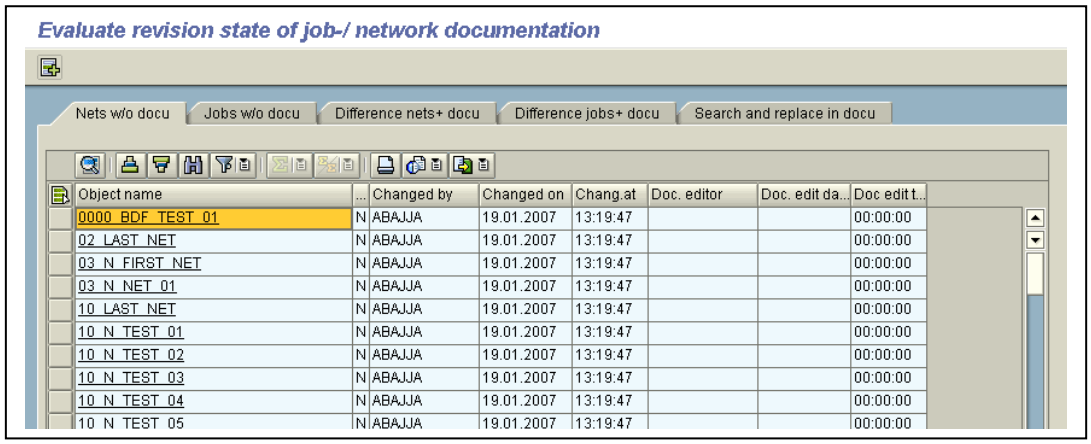
5.16. Documentation

These functions offer the options to check the internal **BatchMan** documentation.

- Jobs without documentation
- Networks without documentation
- Different update times job maintenance against documentation
- Different update times network maintenance against documentation
- Searching and replacing in the documentation

5.16.1. Networks without Documentation

This report shows networks without any or with outdated documentation. Via the tab page, you can branch to the other functions. The system only displays networks for which the documentation is not maintained.



Via this icon you can assign an existing text template to the networks.

5.16.2. Jobs without Documentation

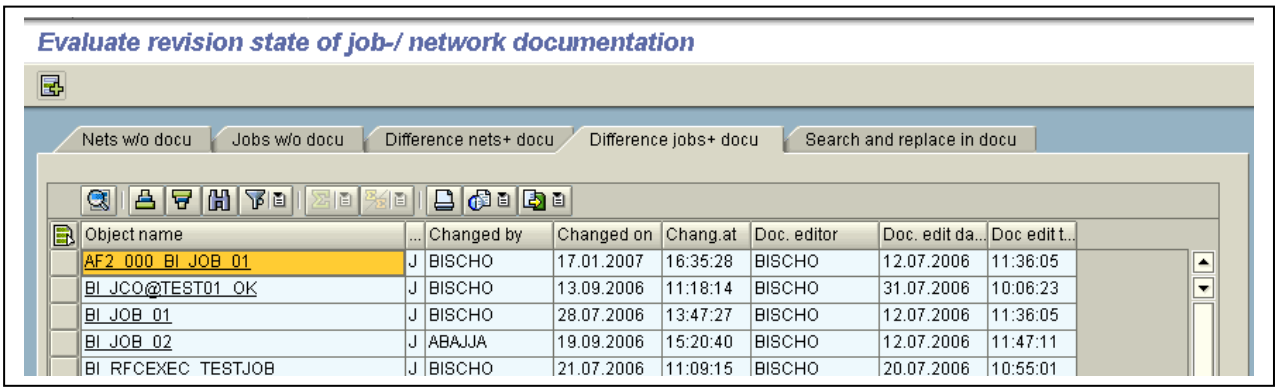
Only those jobs are displayed in which the documentation has not been maintained. Otherwise analog to the networks..

5.16.3. Difference Networks – Documentation

Those networks are listed, in which the change date of the networks deviates from the change date of the documentation.

5.16.4. Difference Job – Documentation

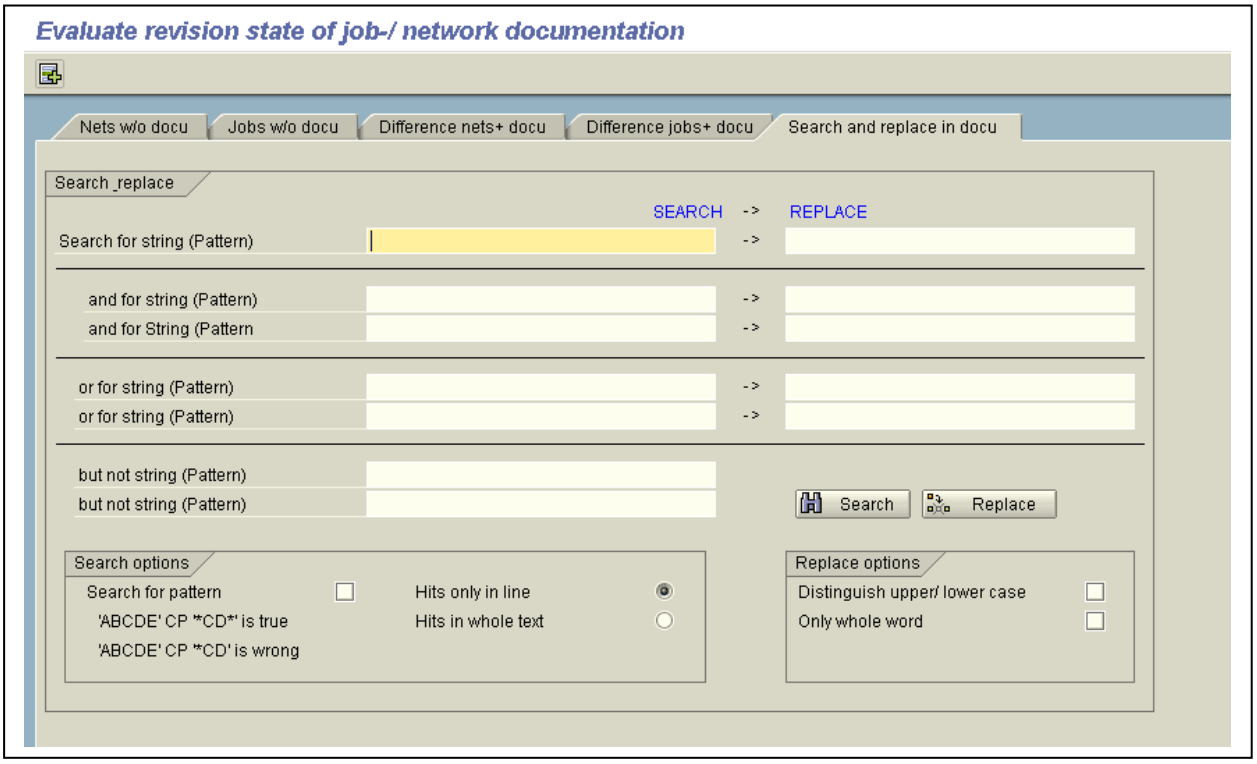
Those jobs are listed in which the change date deviates from the change date of the documentation.



Via this icon you can attach an existing text template to an existing documentation. This can be useful if the previous documentation was created without structure, and now should be formatted.

5.16.5. Searching and Replacing in the Documentation

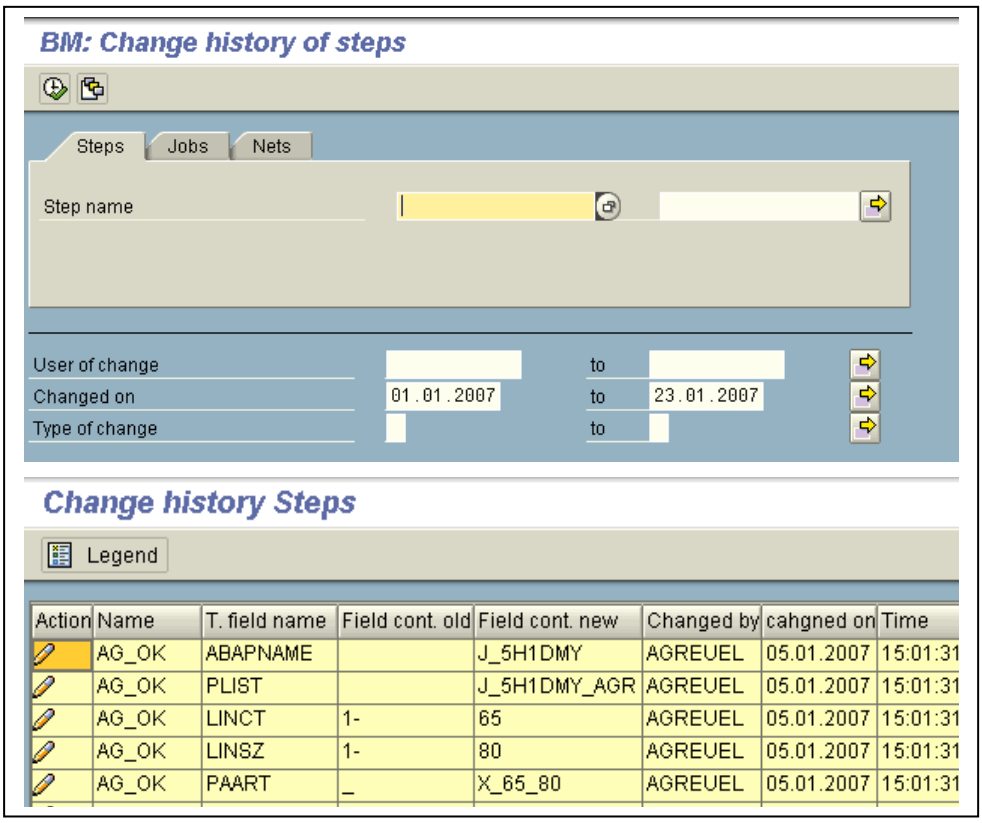
This function is used to search and replace strings within the documentation of jobs and networks. The search terms may be combined freely. You can determine whether the search result is to refer to one line, or whether it applies to the entire documentation. You can differentiate between a string search and a pattern search. The hit list displays an overview of the jobs/networks found.



5.17. Change History

5.17.1. Change History - Steps

The changes are listed in steps depending on the chosen selection. You can select according to step name from the master data and according to user, date and time from the history.



The icons of the actions correspond to the ones of the buttons for master data maintenance.

5.17.2. Change History - Jobs

The changes are listed in jobs depending on the chosen selection. You can select according to subject area and job name from the master data, and additionally according to user, date and time from the history. Analog to steps.

5.17.3. Change History - Networks

The changes are listed in networks depending on the chosen selection. You can select according to subject area and network name from the master data, and additionally according to user, date and time from the history. Analog to steps.

5.18. Where-Used List Customizing

Different programs for the where-used lists of the customizing definitions are collected here.

In principle, these programs are called up exactly for this object from the appropriate functions in the maintenance of the object in customizing. Executing this report makes usage of all objects possible.

6. Customizing

In the Customizing reports is where you can make customer-specific settings. Not all settings are described here in detail. Some functions are special applications, which not only require comments on Customizing, but also on execution, handling or monitoring. For this reason, special documentation is provided.

6.1. Setup Programs

6.1.1. System Information

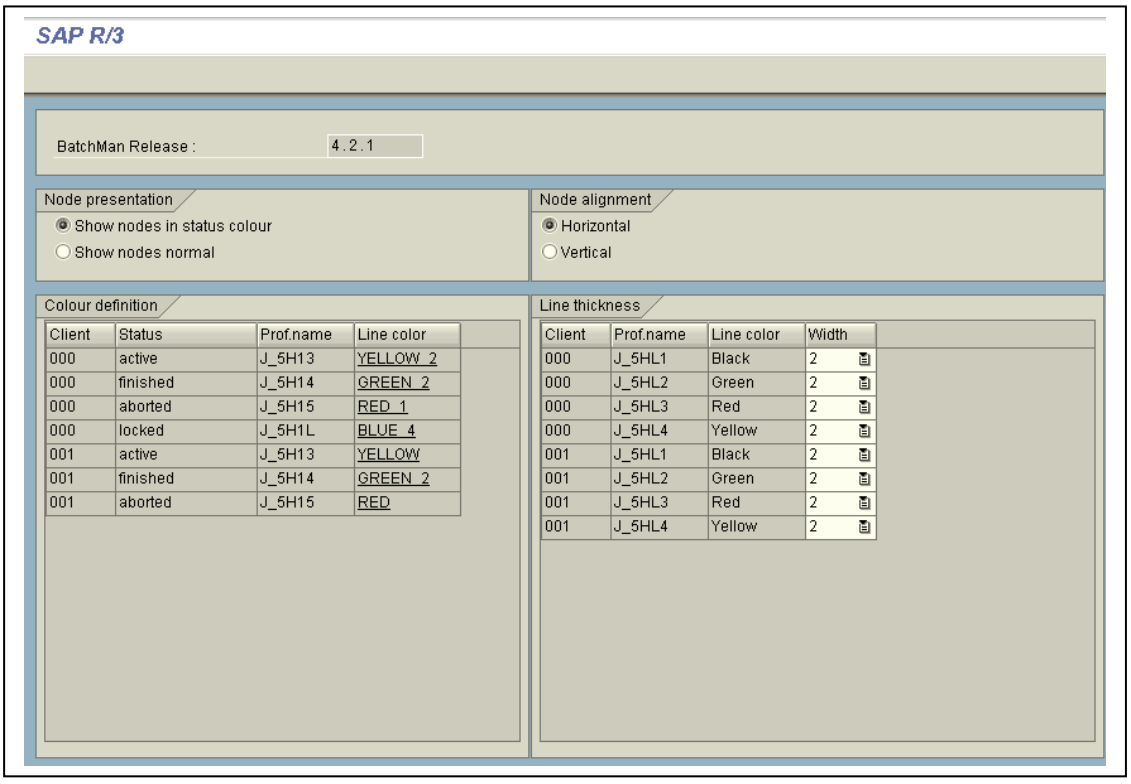
See chapter 1.8 for a detailed information (BatchMan System Information)

6.1.2. Setup Parameter

The installation manual describes the significance of all the setup parameters.

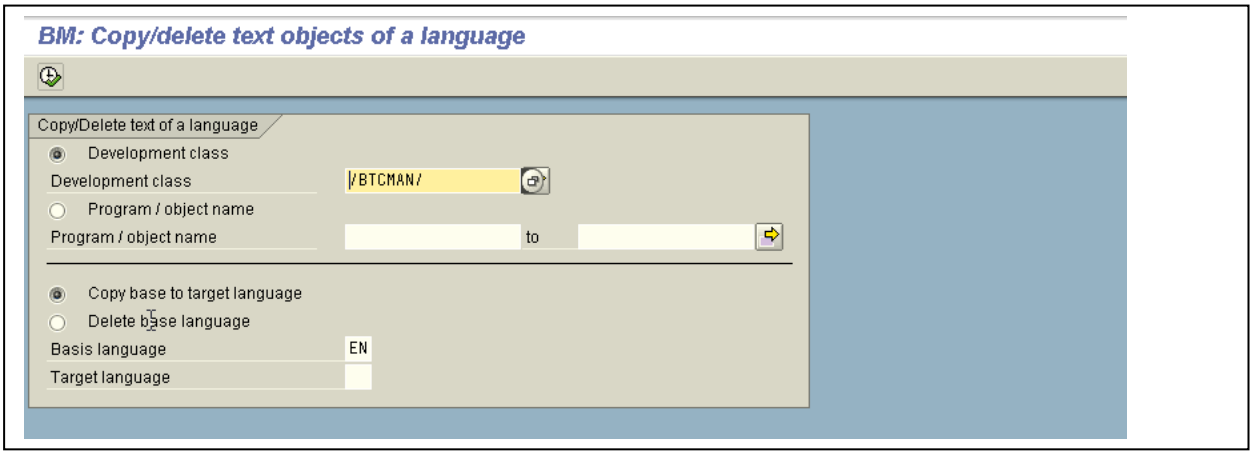
6.1.3. Maintain graphics profiles

With this function, you can individually set the line thickness for the graphics.



You can change the line thickness in the range of 1 (thin) to 4 (thick). The settings of the graphics are described in detail in the installation manual.

6.1.4. Copy Languages

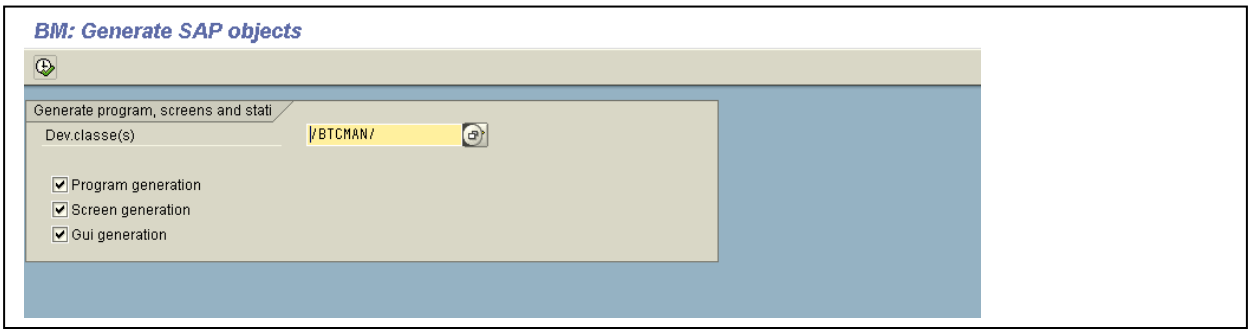


With report **/BTCMAN/M_TXT**, there is a tool available to copy one language to another. It is a language supplementation. You can copy from EN (English) to FR (French), for instance. Then in the French version, all texts are available in English.

The development classes are taken from table /BTCMAN/C_DEVC.

The translation is to be done with an English or German logon. Afterwards, logon in the target language. If the system still displays CUA bars and menus in German, regenerate the report using transaction SE41.

6.1.5. Generate development objects



If after the language supplementation the system still displays CUA bars and menus in German, you can regenerate all BatchMan objects with this report (/BTCMAN/M_GEN).

The development classes are taken from table /BTCMAN/C_DEVC.

To be on the safe side, you should start the report from SA38. Then the system generates all BatchMan reports (except /BTCMAN/M_GEN). You are to generate it manually from SA38. If you start the report from the function tree, the system also does not generate report /BTCMAN/M_REP (this sets up the function tree).

6.1.6. Administration backup system

The backup manual describes the significance of all the setup parameters.

6.2. Technical customizing

6.2.1. Remote Connections / Targets Remote

The settings for the remote connections belong to the settings required in the system. You can find out how to set these up in the installation manual.

6.2.2. Definitions of non-SAP servers

Here is where the servers or HOSTs are stored for the connection of non-SAP functions. The documentation is stored in BM_e_nonSAP_YYMM.pdf

6.2.3. Setup WebService

Storing the WebService definitions as per the documentation BM_d_webserv_JJMM.pdf

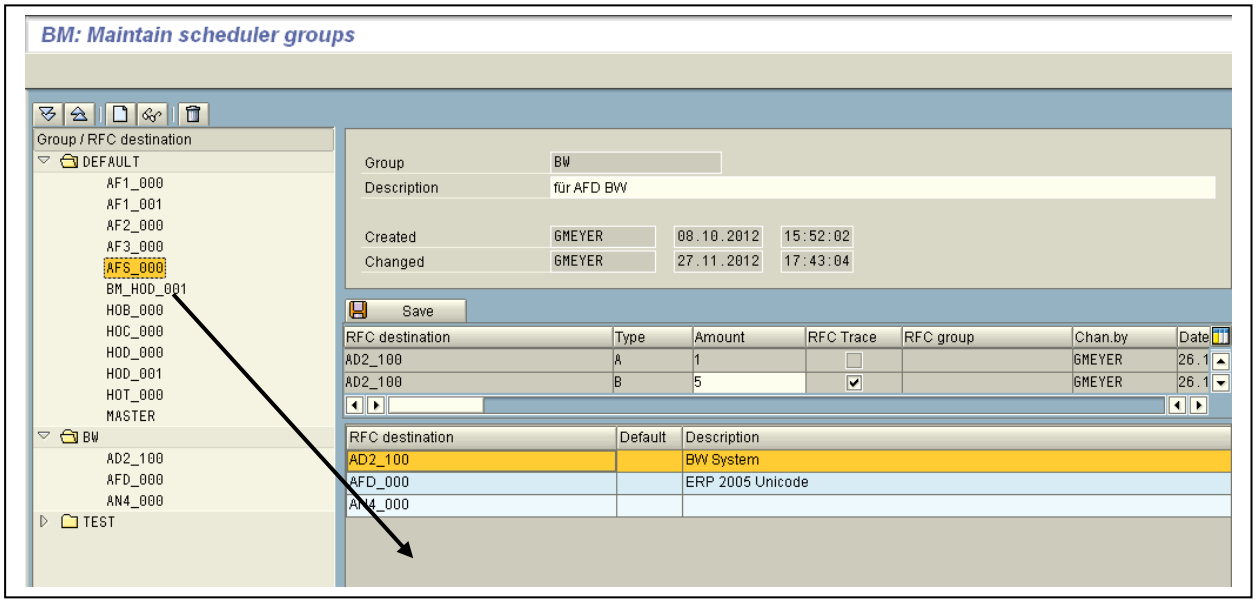
6.2.4. Variant for WebService

Storing the variants for executing the WebServices as per the documentation BM_d_webserv_YYMM.pdf

6.2.5. BatchMan Scheduler groups

In BatchMan, you can parallelize several processes. As a result, you can increase the throughput of the processes processed.

When creating scheduler groups, the system automatically creates a DEFAULT group. In this group, the system first includes all available and also all new remote connections. Then the remote connections can be assigned to another group.



To do this, mark the remote connection on the left side and move it to the right side.

For each group, the system starts separate processing via the BatchMan scheduler job.

Within the scheduler group, you can set another parallel processing. For BW processing (job type B), you can set the number of parallel processes. The maximal number is the number of dialog processes divided by two and rounded up to the next whole number.

6.3. Basis Customizing

6.3.1. Order types

Order types are already described in point 3.2.

6.3.2. STD Plans/ Shift plans

Function tree BatchMan → Customizing → Basis Customizing → STD-Plans / Shift plan

As of BatchMan 3.3, several standard orders (Type STD + plan) can be generated per day. In this way, a distinction can be made between different organizational units like for example countries/states, factories, company/posting codes.

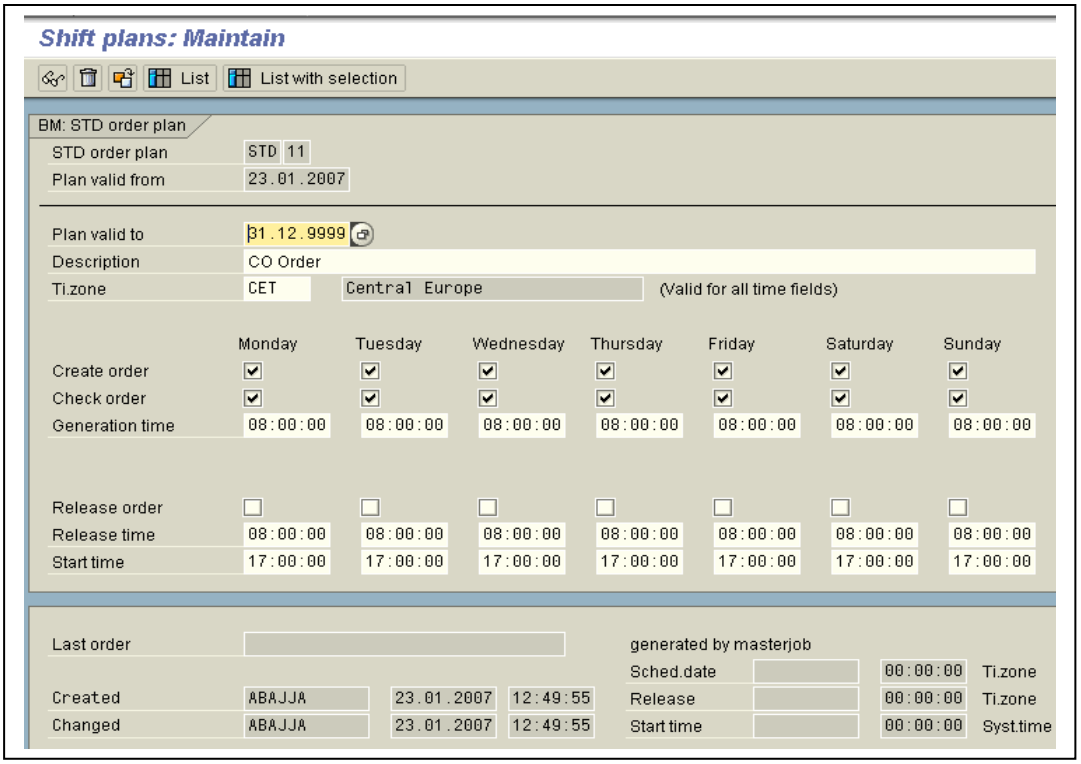
This automatic order creation in the plans is specified for orders of the type STD: the **plan**ident is two-digit and must be alpha-numeric. Hereby the marked weekdays determine on which day a proposal list should be generated. Create; check and release determine which functions should be executed automatically.

The **Generation time** determines the time when the STD should be created and checked; the **release time** determines when the STD is released; and the **Start time** determines when the order will be started.

If no time zone is indicated, the system time zone is valid for all systems. If, however, a time zone is indicated, then all times refer to this specified time zone. This makes is possible to create time zone –related STD orders.

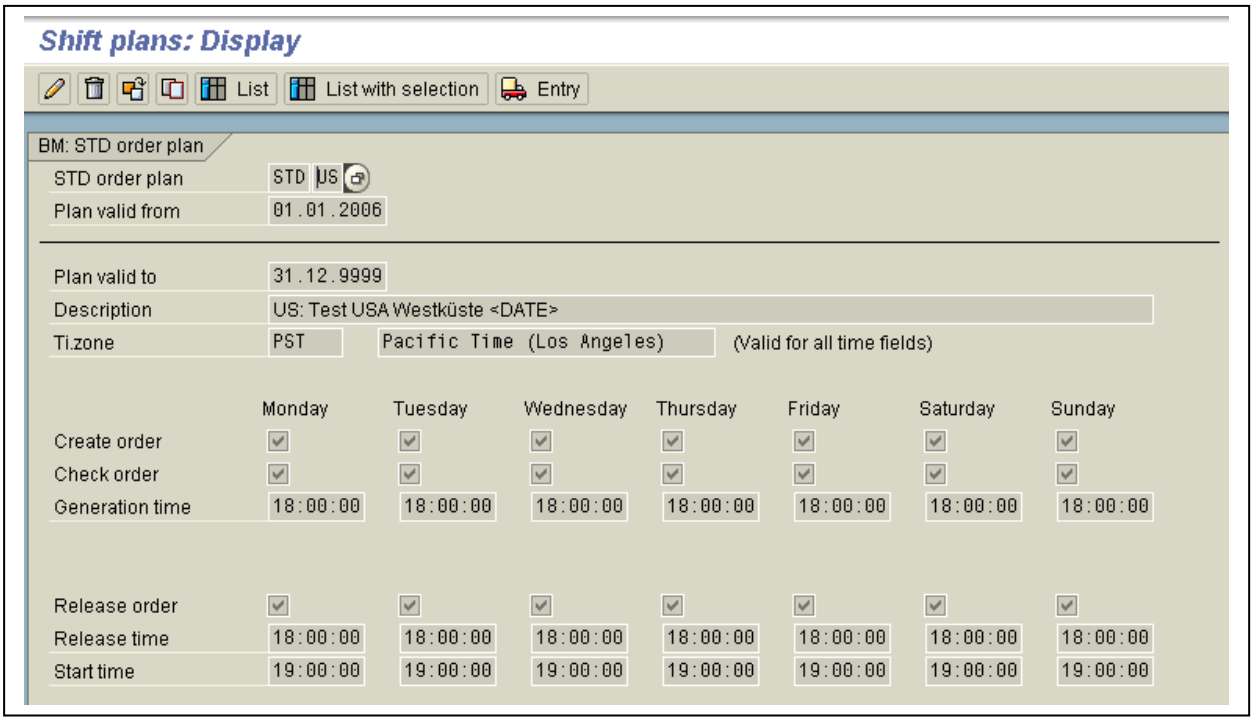
Date: The **from** Date determines the effective date of the shift plan (when it becomes valid); the **to** date until what time it is valid.
In the field **description**, the text is deposited which was given to the order at the time of creation. The parameter **<Date>** is filled with the actual (day) date when the standard order is created.

All STD plans of the system are displayed via the list.



It is possible to automatically generate a STD-order (e.g. at 08:00 a.m.) with all jobs and networks of the day. Between 08:00 a.m. and 17:00 p.m., manual changes (adding of jobs/networks, removing of jobs/networks, changes in the processing/moving of jobs/networks) can be done. At the release time, the STD-order (e.g.. at 17:00 p.m.) is then automatically released.

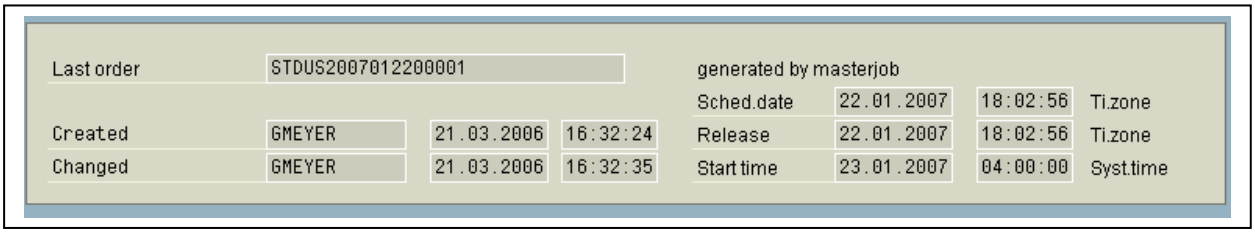
If a time zone is used, you must consider that the generating time is maintained in the time zone.



Since an SAP system only works with one time zone, the system time zone, (here CET), the time zones are converted accordingly. It is most likely (as shown is this example), that the generating-release- and start time will fall already on the next day of the system time.

Analog, if a time zone is specified shortly after the date change (e.g. Australia or Japan), then a STD order will already be created for the next day.

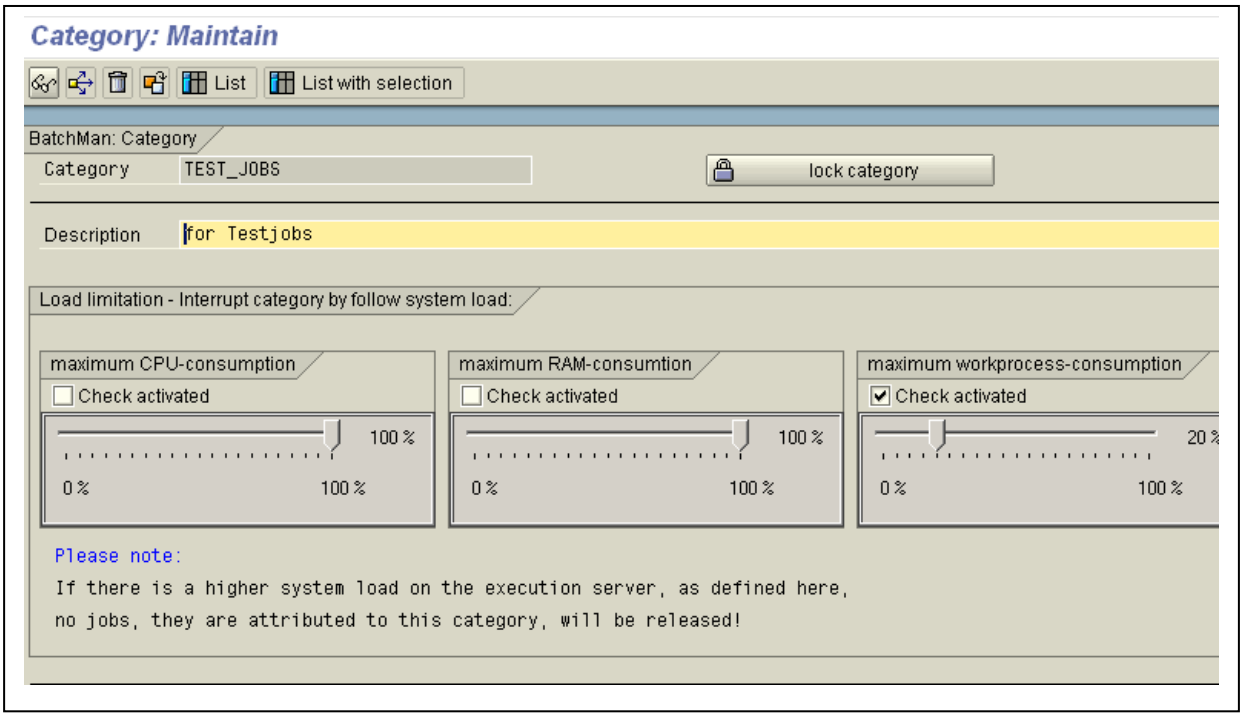
The last order that was created by the master job for the respective plan is shown below (in this case for STD Plan US).



Within **BatchMan,** you can start from two views.

1. The system displays the plans like the calendar days. Scheduling then takes for each calendar day from 12:00 a.m. to 12:00 p.m. Problems arise if the processing in the morning depends on the processing from the day before (evenings).
2. The system displays the plans as complete processing. Since these start only after the end of the dialog time, the plans are defined such that the STD orders start from 6:00 – 7:00 p.m. This is also called end of day processing. For independent jobs and networks, an STD plan like (1) is created.

6.3.3. Categories



You can assign categories both to jobs and to all order types.

Using categories, objects can execute actions belonging to the category, namely actions that belong in the system administration area:

- define maintenance windows,
- Stop and start manually via the objects.

You can use categories to define how objects assigned to those categories are to behave in the event of:

- CPU use
- RAM utilization
- Work process assignment.

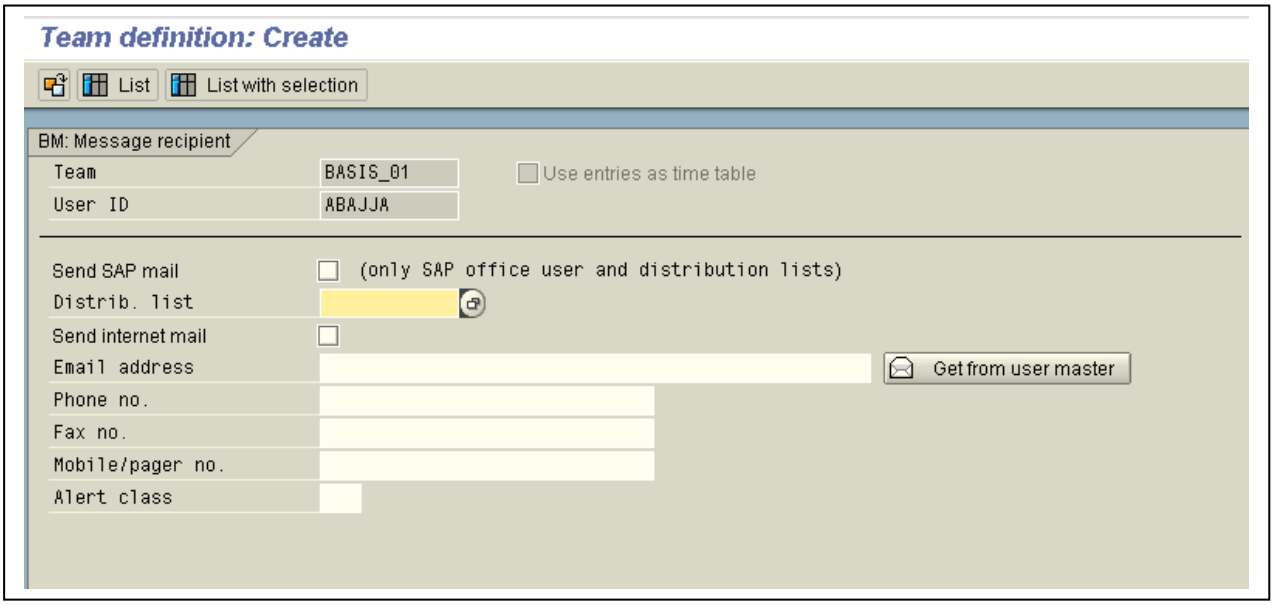
If the amount of resources is exceeded, the system blocks the execution.

6.3.4. Team- / Recipient Definitions

This function allows you to define response groups (**teams**). These teams define who is included in the distribution list if a message is triggered. There is a difference between the definition of a team with arbitrary employees, and a time table.

6.3.4.1. Team Definition

At the initial screen/step for the team maintenance, the flag ‚Use entries as time table' must remain



empty/not activated!

For a group/team, it is possible to assign any number of employees (user names). In this case, the user need not exist in the SAP System. Dummy entries are also possible (user XXX), if for instance an external alert tool is used for notification.

When sending using SAP Office (alert tool), you can select whether this is to occur as pure SAPmail and/or as e-mail. If the user is already assigned an address in the user master record, this can be read using the “Get from user master” pushbutton.

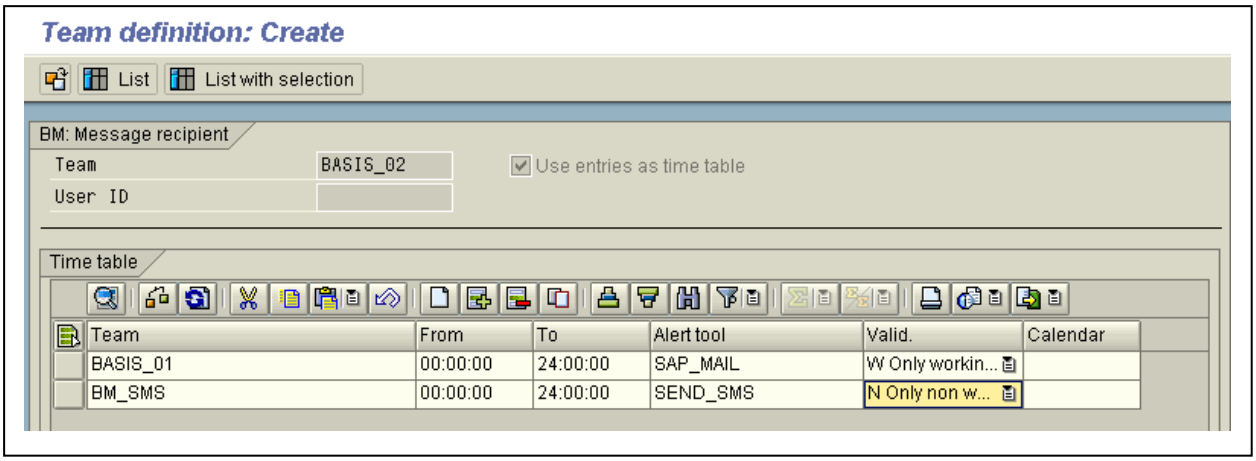
6.3.4.2. Define Time Table

At the initial screen for time table maintenance, the flag ‚Use entries as time table' should be marked / activated. The user name remains EMPTY.

Time tables are used to

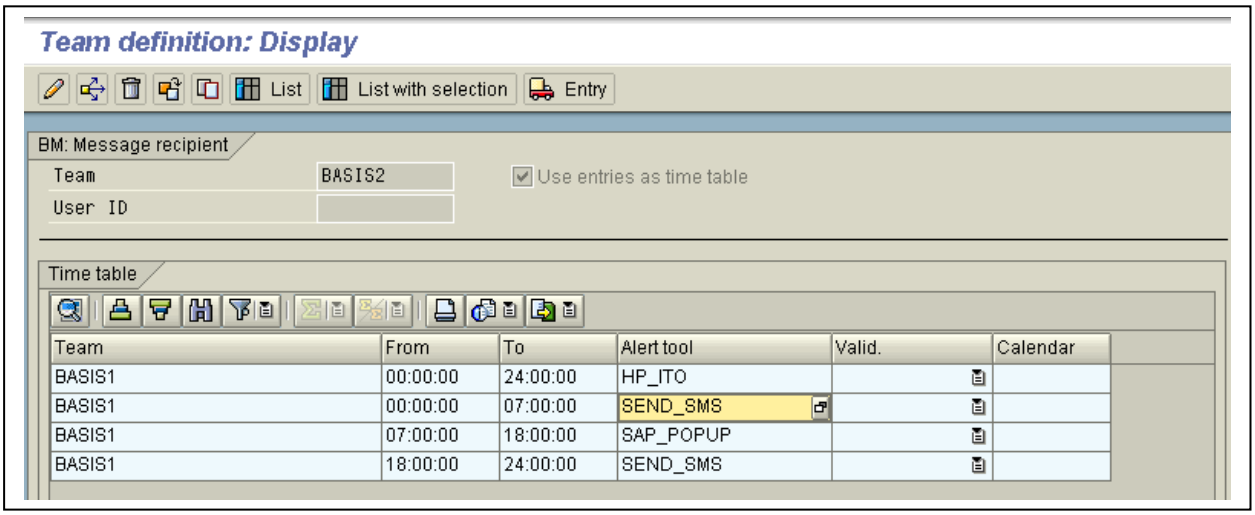
- notify several teams
- notify the same teams in different time windows with different alert tools

You can also combine both functions.



Now you can enter different teams with the time window (from – to) and the desired alert tool in the table. Via the validity entry you can specify when notification should occur.

- Leer Always
- ‚W' Only on working days (calendar must be specified)
- ‚N' Only on non-working days (calendar must be specified)



A message is always transferred to the tool HP_ITO (HP Openview). In principle, the escalation can occur there as well.

In addition, a notification is sent via tool SEND_SMS (send SMS) for the time frame from 00-07 hours and from 18 - 24 hours. An SAP Popup message is sent for the time frame from 07- 18 hours.

6.3.5. Alert tool Configuration

Documentation **BM_e_Alert.pdf** exists fort he configuration of the alert tools. Customizing of the different alert types is described there with examples.

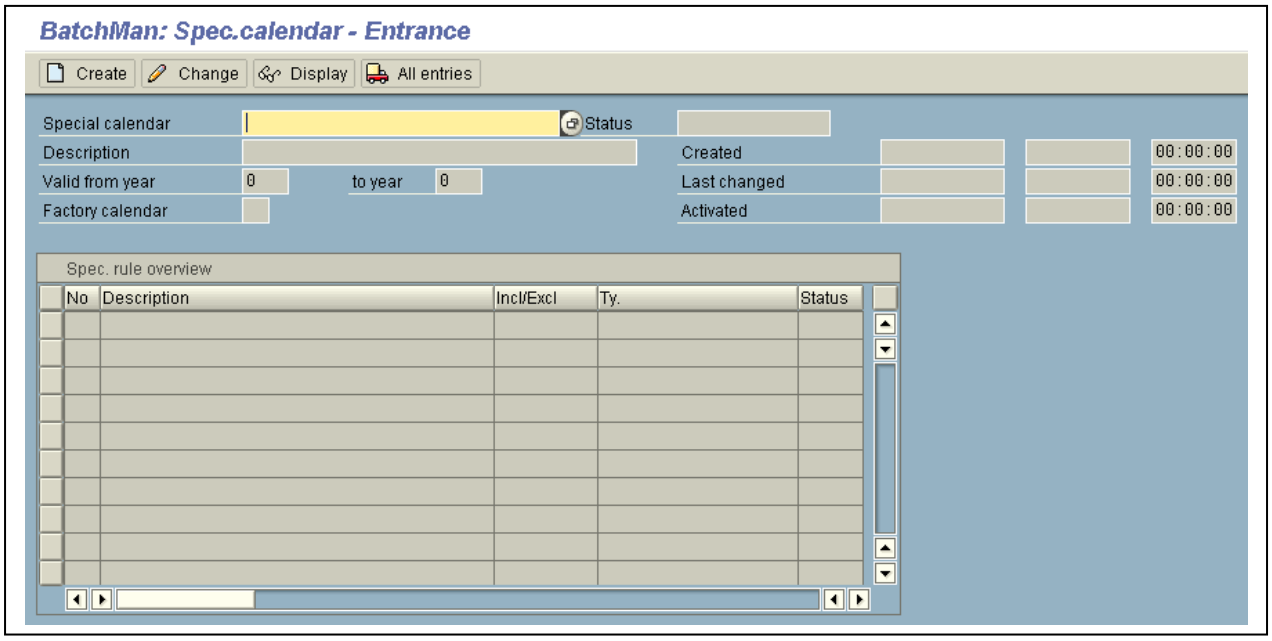
6.3.6. Special Calendars

BatchMan® → Functions tree → Customizing → Basis Customizing → Special Calendar

6.3.6.1. Introduction

The special calendars provide the option to define separate calendars within **BatchMan**. Here, the system calculates a special calendar from its special rules.

If you change a previously saved special calendar, there are two versions of the special calendar. One version is still active and the other is modified. Thus, it is possible to retrieve the old special calendar (rules) and to reactivate it if the changes are not to be used. While a user processes the special calendar, maintenance is blocked for all other users.



Special calendar: Name of the special calendar under which it is addressed in **BatchMan®**.

Status: Active or revised. In **BatchMan**, only active special calendars are used.

Description This is a short description of the special calendar.

Factory calendar: SAP factory calendar to which the special calendar refers for determining the non-working days and holidays. Entry is required.

Valid from year: Year from which the special calendar is valid.

to year: Year to which the special calendar is valid.

Administrative data: Created by, Last changed by and Activated by: User with date and time. The table below lists those special rules defined for the special calendar.

Number: Sequential number of the rule. The sequence of the rules is key for activating the calendar (see activating the special calendar).

Description: This is a short description of the special rules.

Inclusive/ Exclusive: Inclusive (through the rules, days are defined on which the execution occurs)

Exclusive (through the rules, days are defined which are omitted from the execution)

Type: This is the type of table entry (see rule maintenance)

Buttons for the maintenance of special calendars

Create: The special calendar is created after you enter the name of the calendar

Change: Change the special calendar

Display: Display the special calendar

Other object: Init mode in special calendars maintenance

Where-used list: Usage of the special calendar in master data (network and jobs)

Delete: Delete the entire special calendar

Quick selection: Search assistance for special calendar

Copy : Copying the complete special calendar

Create Rule: Create system of rules necessary for the computation of the calendar (1 to n of entries)

All entries: Place all active special calendars of the current SAP system in a transport order. The transport can be imported after releasing it into another system.

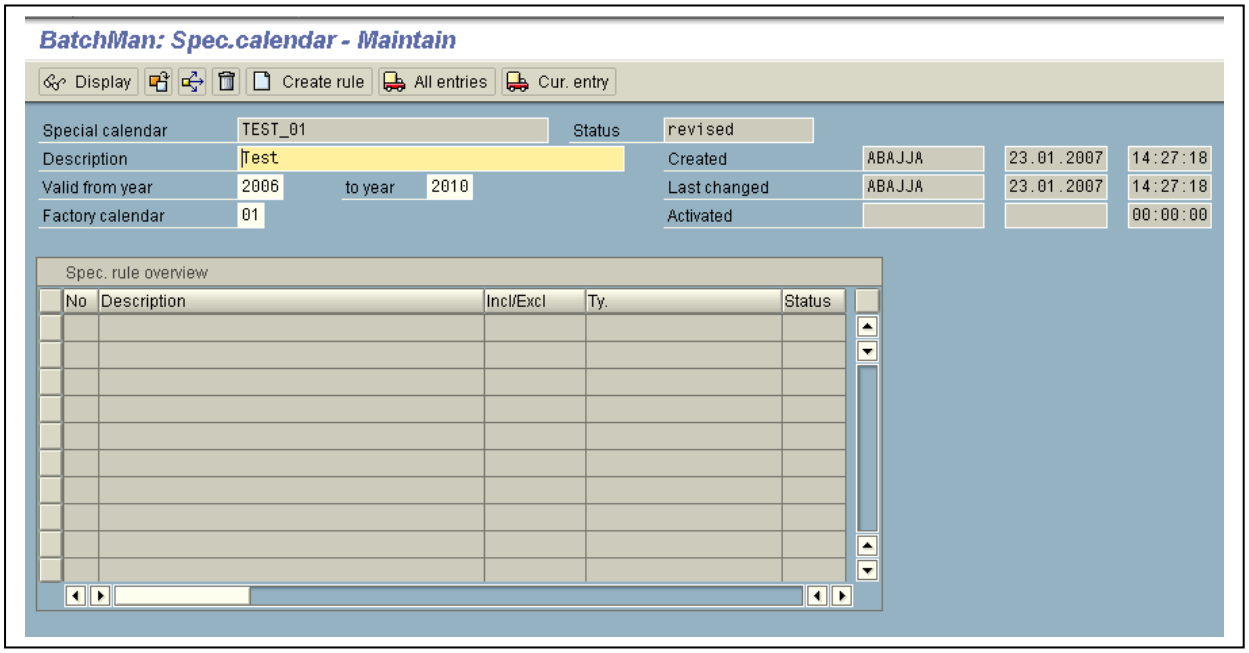
Current entry: Place only the current (active) special calendar in a transport order.

Activate calendar: The changed special calendar is activated. Thus the special calendar is usable in **BatchMan**.

Get active calendar Retrieves the active rules after a non-satisfying change and overwrites/deletes the changed rules.

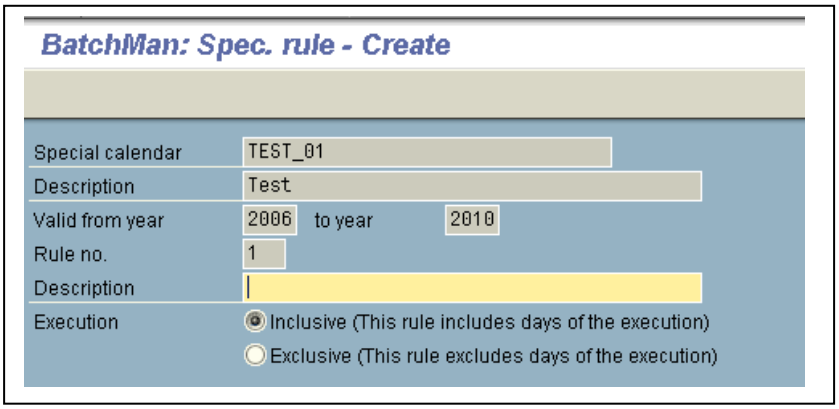
6.3.6.2. Create Special Calendar

After entering the name of the special calendar and pushing the button' Create', the fields become ready for input. Entering the factory calendar and the validity years is mandatory. Afterwards SAVE.



Now you can create the special rules (rules) that are associated with the special calendar by pressing the button ‚Create rule'.

6.3.6.3. Maintain Special Rules (Rules)



Number: Sequential number of the rule. The sequence of the rules is crucial when the calendar is activated (see 'Activating the Special Calendar).

Description: Short description of the special rule

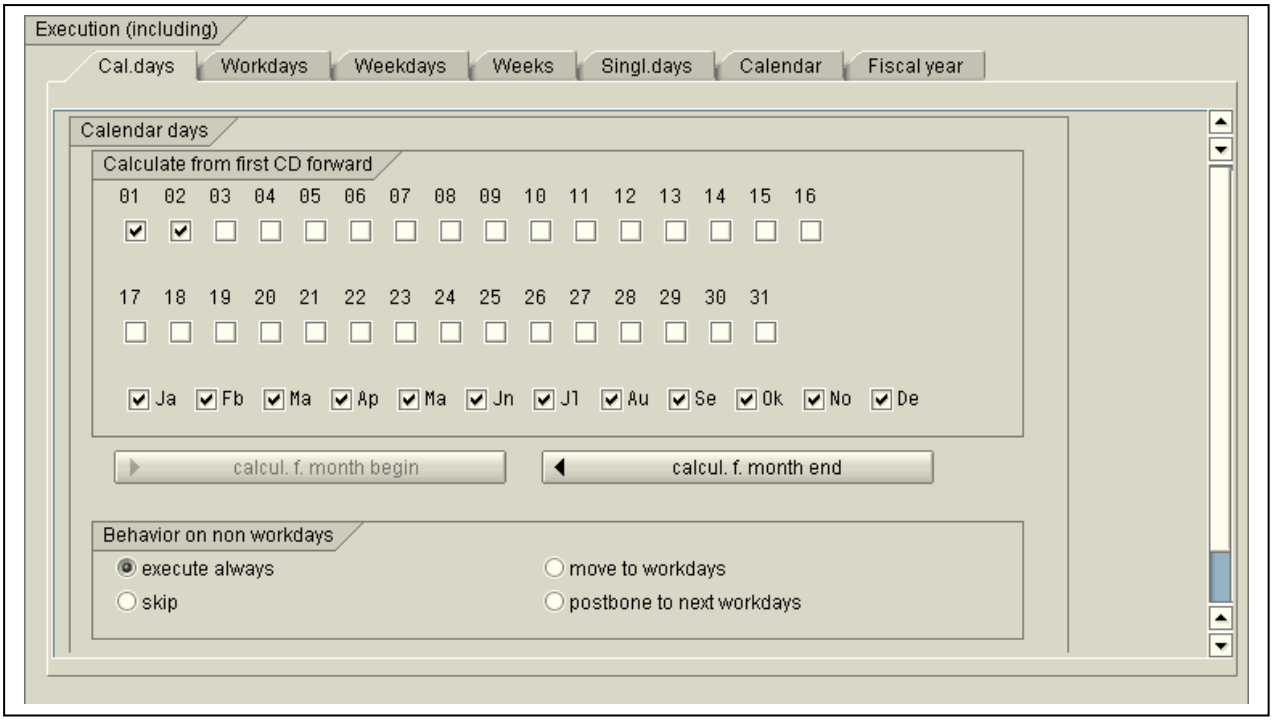
Inclusive Execution: Describes that the days defined by the rules are execution days.

Exclusive Execution: Describes that the days defined by the rules are excluded from the execution.

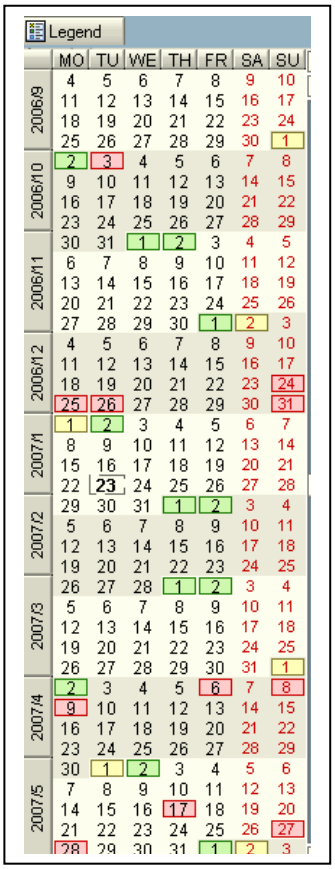
Delete: Deleting the current special rules

In the rule maintenance, rules can be defined for calendar days, working days, week days, weeks and individual days; with reference to SAP factory calendars or **BatchMan** special calendars, or with reference to the fiscal year variants.

Calendar days are active as standard.



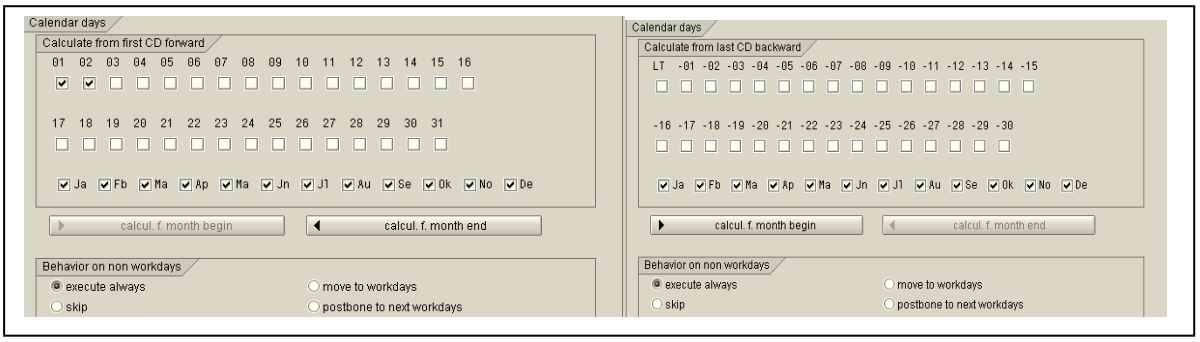
If a new rule is saved, then a calendar with the new rule is shown on the right side. Depending on the rule execution (inclusive or exclusive), the days are marked in color. Legend with interpretation of colors.



Changed values are updated in the calendar only after saving.

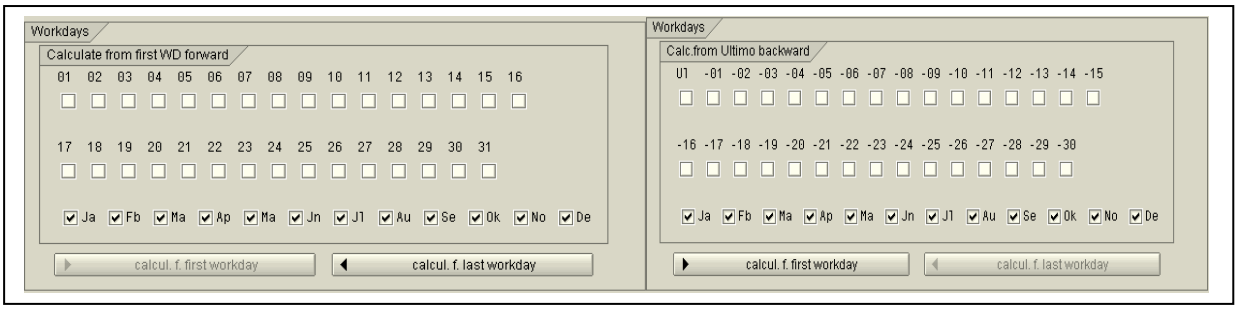
Calendar days

Calendar days define the days of a month. The days can be marked starting with the month beginning or the month end (LD – last day -01 – penultimo/day before last day). Via the month box you can select the month(s). The behavior, if one of the selected calendar days should fall on a non-working day (definition of the special calendar) according to the factory, can also be set up.



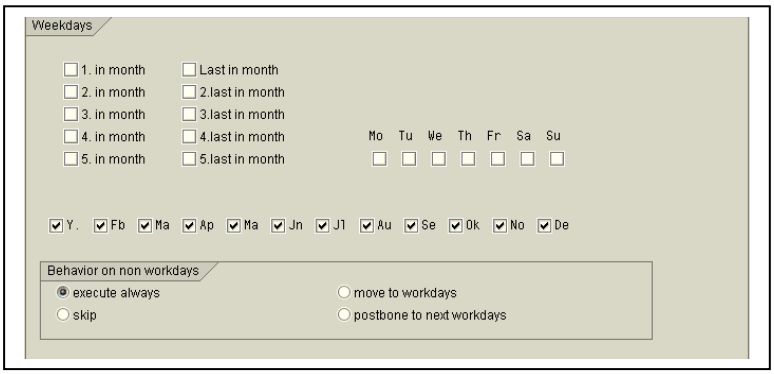
Work days

Work days define the work days of a month, referring to the factory calendar, which was defined when the special calendar was created. The workdays can be marked starting with the month beginning or the month end (UL – Ultimo=last workday; -01 – penultimo/workday before last workday). Via the month box you can select the month(s).



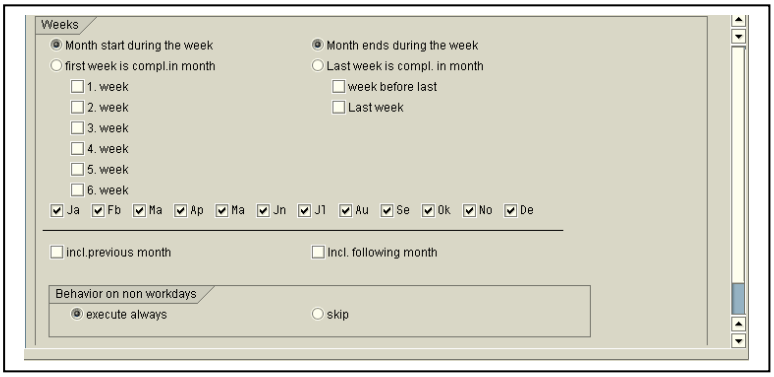
Week days

Hereby you can define weekdays and their 1. to maximally 5 appearance(s) in the month, either in the beginning of the month and/or at the end of the month. Via the month box you can select the month(s). The behavior, if one of the selected calendar days should fall on a non-working day (definition of the special calendar) according to the factory, can be set up.



Weeks

Here individual weeks can be defined within the months. Via the month box, you can select the month(s). The behavior, if one of the selected calendar days should fall on a non-working day (definition of the special calendar) according to the factory, can be set up.



Month starts in the week: The week in which the first calendar day of the month begins is considered the first week. **Incl. previous month** means, that a full week is considered, if the week beginning (Monday) lies still in the previous month.

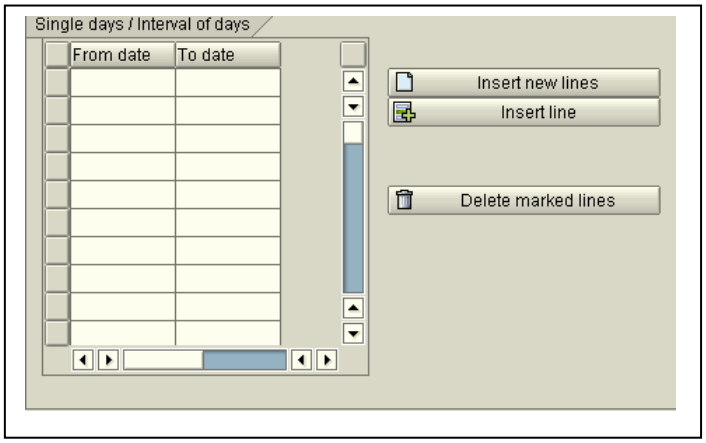
First week is completely in month: The week that lies completely in the month from Monday to Sunday is considered the first week. **Incl. previous month** has no meaning here.

Month ends during the week: The week in which the last calendar day of the month lies is considered the last week. **Incl. following month** means, that a full week considered if the end of the week (Sunday) lies in the subsequent month**.**

Last week is completely in month: The week which lies still fully from Monday to Sunday in the month is considered the last week. **Incl. following month** has no meaning here.

Single days

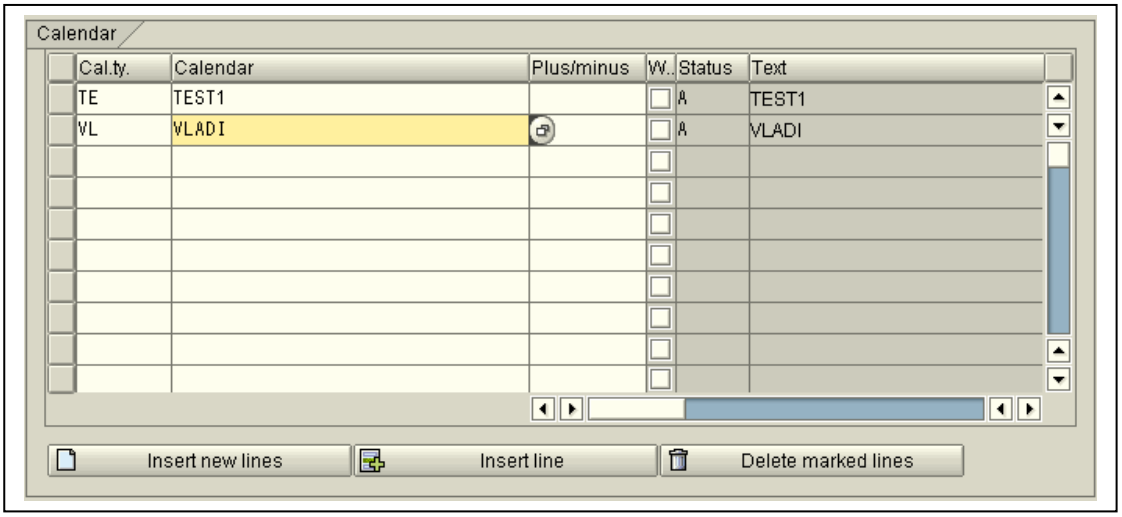
Individual days or a range of days can be defined.



Reference to SAP Factory Calendar or BatchMan Special Calendar

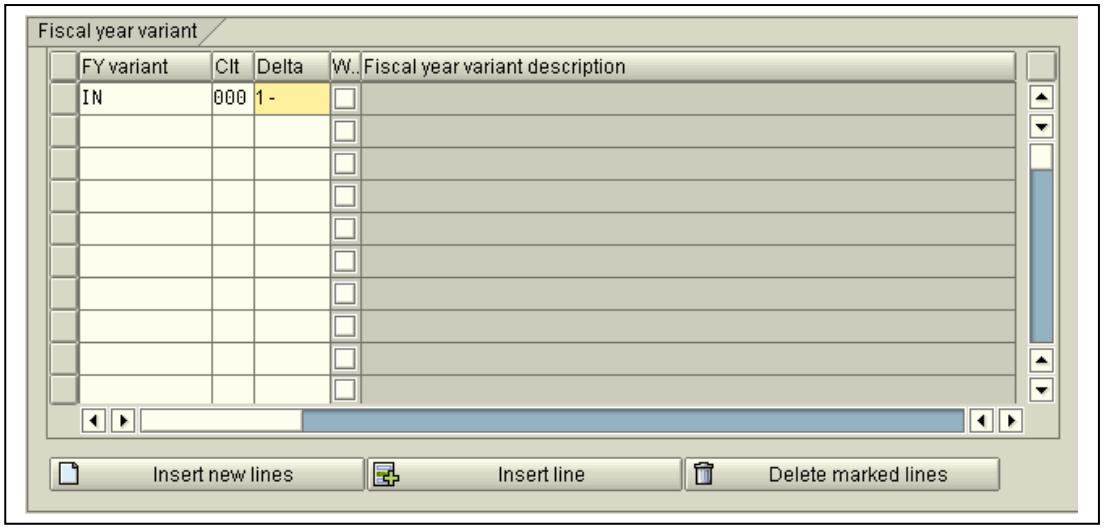
Here you can build up rules which are based on the SAP factory/logistics calendars and/or other BatchMan special calendars.

With the **'Plus/Minus'** and **'W' (workday**) fields you can change the rule. This special calendar can be further refined with additional rules.



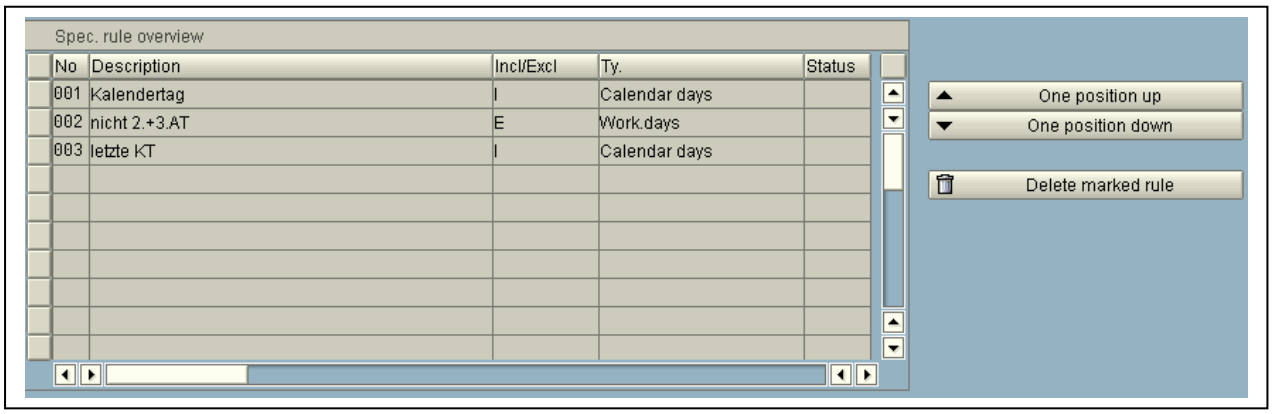
Reference to the Fiscal Year Variants

Here you can build up rules that are based on SAP fiscal year variants. With the **'Delta'** and '**W'(workday**) fields you can change the rules. This special calendar can be further refined with additional rules.



6.3.6.4. Maintain Special Calendar

The rules appear in the initial/entrance screen after maintaining them.



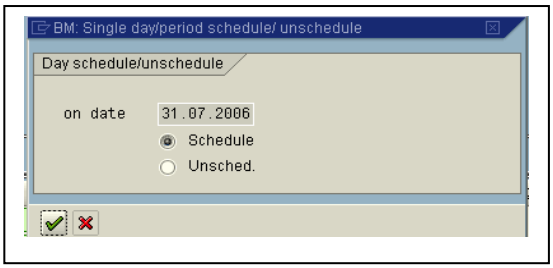
Using the buttons on the right next to the table, you can move the sequence of the rules after marking them. You may insert additional rules. The calendar with the periods incorporated is shown below.

By double-clicking on a day, the system displays this as a popup. Here, you can toggle between scheduling and unscheduling.

The days that deviate from the rules maintained are saved with the numbers 998 (inclusive) and 999 (exclusive). As rules these are individual days and are also maintainable there.

6.3.6.5. Activating the Special Calendar

When you activate, the system takes into account the special rules defined in the sequence in the calendar as they are arranged in the overview. Thus, first are the rules with the number 001, then 002 and so on.



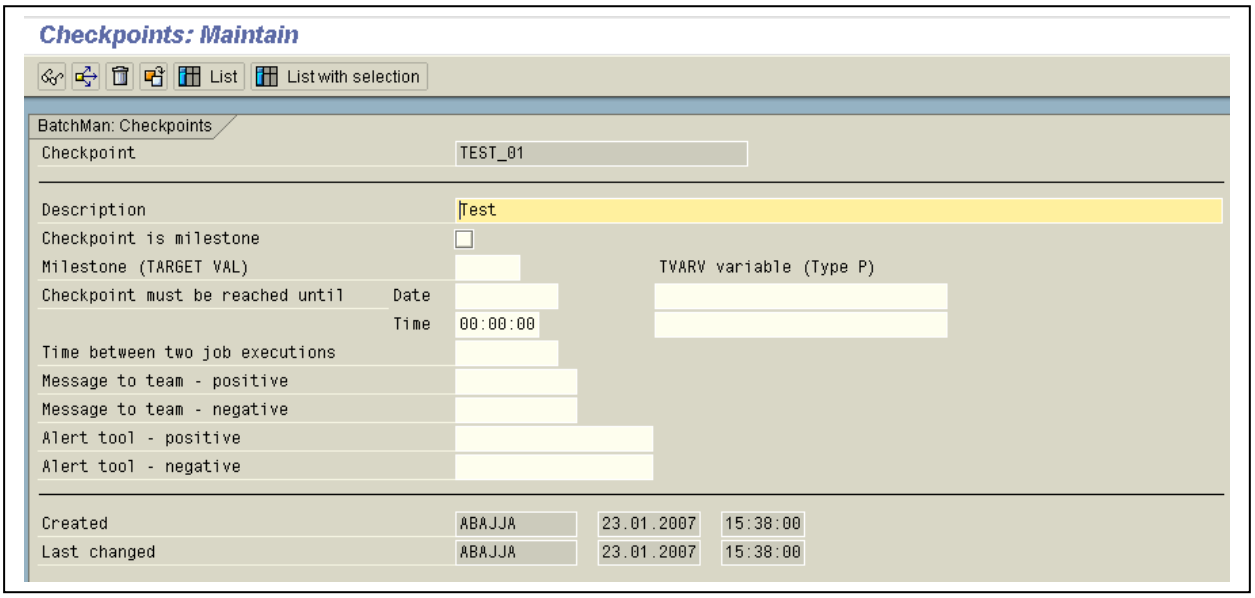
If the display is satisfactory, and hence the definition of the special calendar, it still has to be activated. Call the special calendar display again via the “Activate” button. After you choose the button, the special calendar is active and may be used in the scheduling dates for networks and jobs.

6.3.7. Checkpoints

A checkpoint is a symbolic filed, which is used by BatchMan to determine start characteristics. For the checkpoints, a >= relationship is valid. All jobs or networks, whose checkpoint values are > = (greater as or equal to) the variable, start.

6.3.7.1. Creating Checkpoints

BatchMan Function tree → Customizing → Basis Customizing → Maintain checkpoints



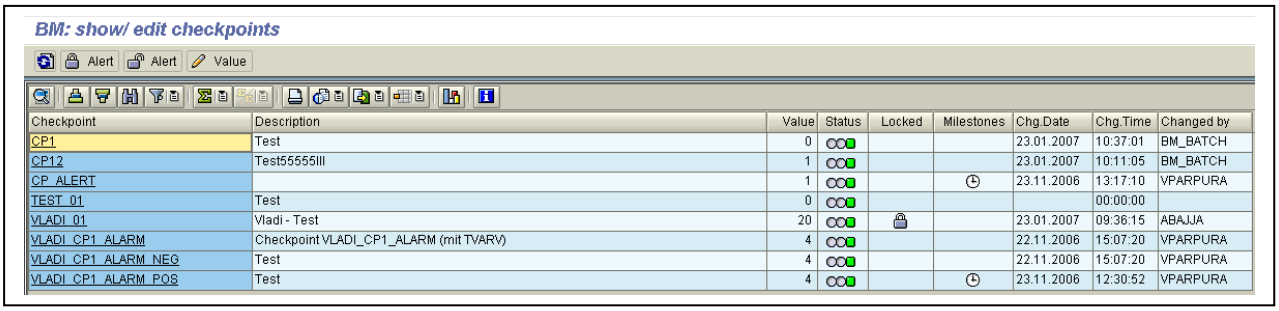
Checkpoints are used mainly to:

- Trigger jobs across networks (also across systems);
- Start an object (job or network) only if several other objects have reached a certain processing state → counter
- Define certain states in the system (use as milestone with the option of alerts and determination of when this milestone is to have reached its target value (dynamic via TVARV))

6.3.7.2. Setting Checkpoints Manually

This report is used for monitoring checkpoints and for changing checkpoint values.

Call via *BatchMan Function tree: → Monitoring → Checkpoints and Resources→ Edit checkpoints*. Selection of the checkpoints can be limited via the selection option.



Checkpoint	Description	Value	Status	Locked	Milestones	Chg.Date	Chg.Time	Changed by
CP1	Test	0				23.01.2007	10:37:01	BM_BATCH
CP12	Test55555III	1				23.01.2007	10:11:05	BM_BATCH
CP_ALERT		1				23.11.2006	13:17:10	VPARPURA
TEST_01	Test	0					00:00:00	
VLADI_01	Vladi - Test	20				23.01.2007	09:36:15	ABAJJA
VLADI_CP1_ALARM	Checkpoint VLADI_CP1_ALARM (mit TVARV)	4				22.11.2006	15:07:20	VPARPURA
VLADI_CP1_ALARM_NEG	Test	4				22.11.2006	15:07:20	VPARPURA
VLADI_CP1_ALARM_POS	Test	4				23.11.2006	12:30:52	VPARPURA

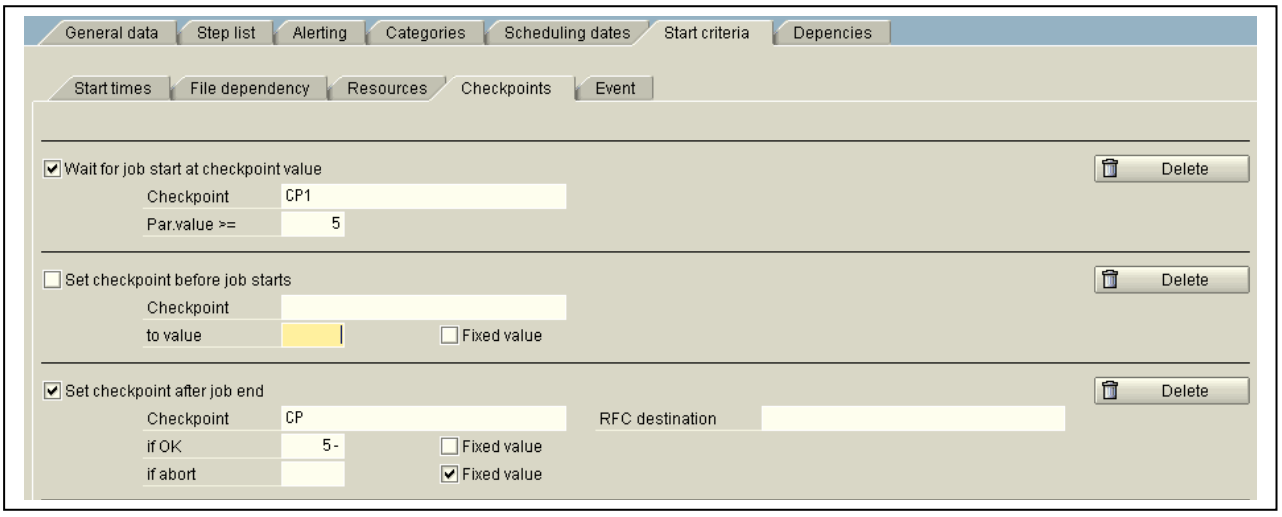
See Checkpoint Maintenance for more information.

6.3.7.3. Using Checkpoints

Checkpoints are scheduled under start criteria in „Define job /network“ (Chapter 2).

6.3.7.4. Setting and Resetting Checkpoints

You can set (reset) a checkpoint under start criteria of a job or network.



Via weight increase **OK** and **Abort** different or the same checkpoints can be set for subsequent jobs.

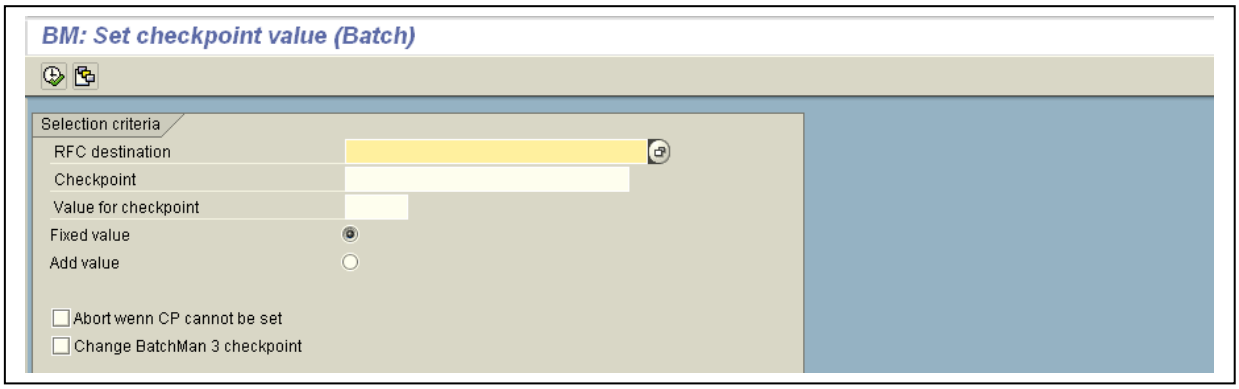
In the above example, the job starts when the checkpoint has reached CP1 >= 5. This is done by predecessor jobs. If the job schedules successfully, the checkpoint is set at a negative value, and thus reset/set back.

An automatic reset at the end of an order is realized with the report **/BTCMAN/C_KP3**. At this point you can change than one checkpoint.

The RFC destination must be maintained in order to maintain a checkpoint in a different system. For the local system, the RFC destination can also be entered, or ‘None’ can be used.

Since the field for the checkpoints is obligatory, a value has to be entered. Entering ‚*’ is possible. Then, in dialogue, an inquiry (for safety reasons/reassurance) occurs again if all checkpoints should be changed.

If the flag ‚Add value' is active, then the entered value is added to the previous value of the checkpoint. If it is not active, then the entered value is set as fixed value. This set up is exactly opposite to the start criteria.



The report is then, for example, scheduled in a network as job.

6.3.8. Resources

Difference Checkpoint vs. Resource

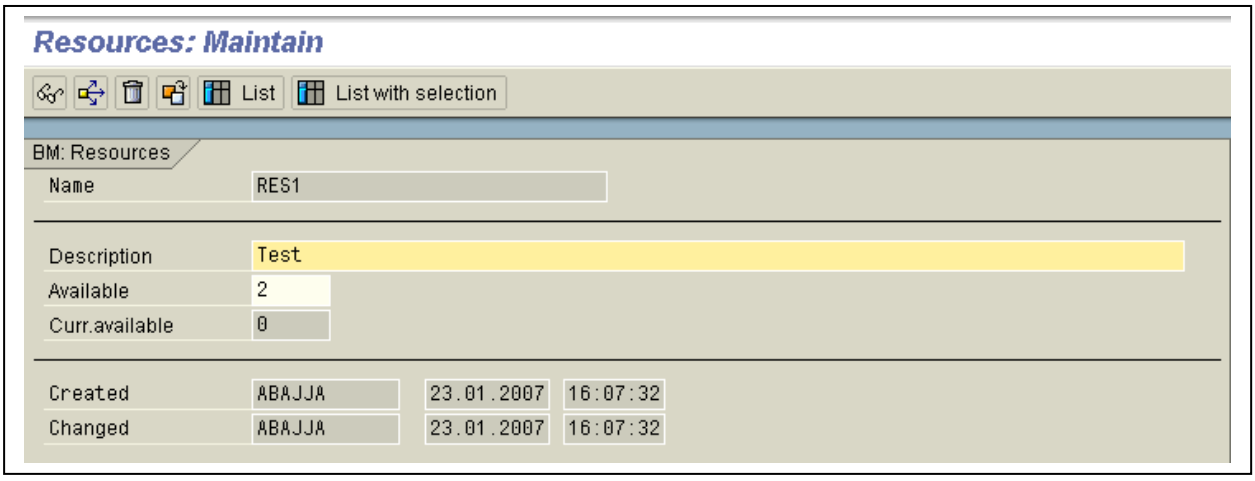
A **Checkpoint** (milestone) is used to provide information about certain conditions of the job control. The value is only changed at the end of a job process.

Resources, contrary to checkpoints, are counted down (counted lower) before the job/network execution. They are released again (counted higher) after the job/network is finished. With this you can control simultaneous access to limited resources like tape devices, batch processes etc.

Whether or not a resource or a checkpoint is available is queried at the start of the job. Like for the checkpoint, a resource must also first be created in a Customizing table.

6.3.8.1. Creating Resources

BatchMan Funktionsbaum → Customizing → Basis Customizing → Ressourcen



A resource is provided with a **maximum available** number of "devices" (tapes, batch processes and the like). Those **currently available** are set by the BatchMan master job. This happens both when a new resource is created and also after a job is terminated (which would not release its

resources otherwise). In contrast, for networks the resource remains filled if a job terminates in the network.

6.3.8.2. Using Resources

Resources are managed in table /BTCMAN/M_RESOBJ.

1. Resources are requested in the start criteria of jobs/networks. The job/network only starts if enough free resources are available. After job/network end, the resources are made available again. In table /BTCMAN/M_RESOBJ, resources are registered with job name and count. If jobs are aborted, the resources will be released by the master job. In networks, however, the resource remains engaged when a job aborts in the network.
2. Resources must be requested manually with the report **/BTCMAN/M_RE1** if resources are to be blocked permanently for several jobs (except via complete networks), manually or externally. In the table /BTCMAN/M_RESOBJ, resources are registered with job name = MANUAL and empty count. These entries are not released by the master job.

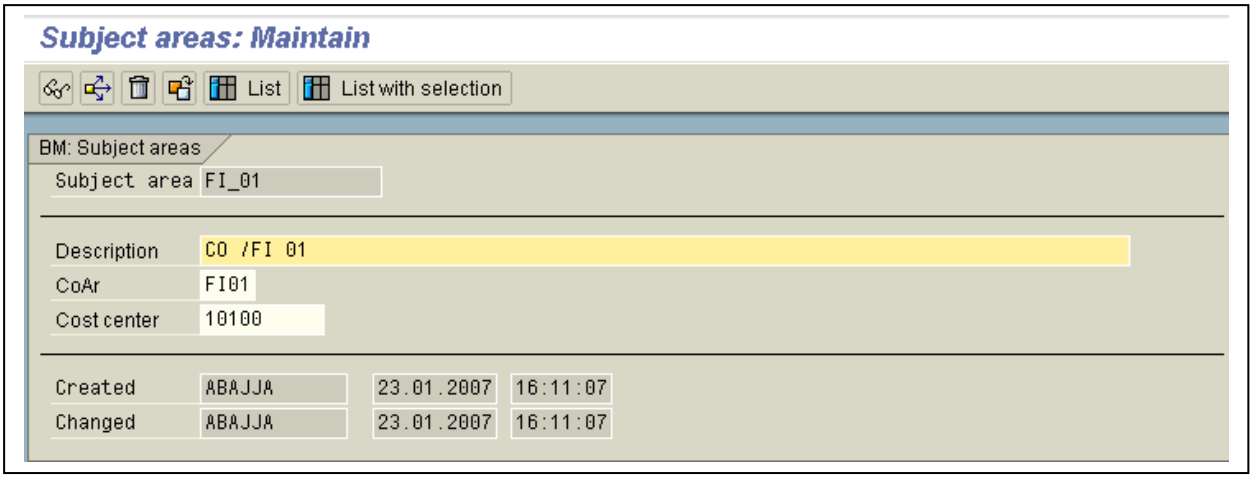
6.4. Extended Customizing

6.4.1. Subject Areas

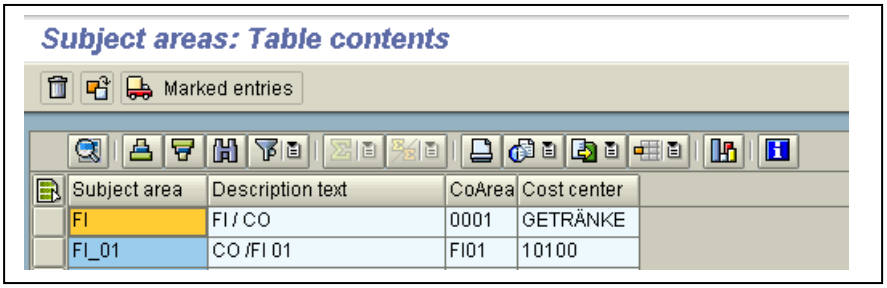
Funktionsbaum BatchMan → Customizing → Basis Customizing → Sachgebiete

The function '**Maintain subject areas'** lets you create and maintain subjects areas. A subject area can be a component of the job- network- data. With the help of subject areas you can group and classify job/networks.

Evaluations can be done on the basis of subject areas. For cost allocation, controlling area and cost centers can be assigned to a subject area. In an ERP system, search requests/help are available for controlling areas and cost centers. These are not available when a non-ERP system is installed (e.g. Solution Manager). The values must be entered manually in this case.



All order types of the system are displayed via the list.



6.4.2. Maintain customer number

Customer numbers may be used in BatchMan to implement a separation of responsibilities in the master data maintenance and in monitoring.

Here, the system does not view customer numbers in such a strict manner that a BatchMan user with customer group 00001 may not see the job names of customer group 00002. Only the displays of contents or changes are prohibited.

The significance of the customer number and maintenance thereof is described in detail in the BM4_d_CustomerNumbers.pdf document.

6.4.3. Change Table RFC target systems

This table is used to define which RFC connection is to be replaced by which RFC connection. This conversion is required after a system copy or after a transport of master data objects (including WDH orders).

The conversion of the target systems occurs with program /BTCMAN/M_RFC and is to be found in *BatchMan function tree (transaction J5HEN) → Mass changes → Convert target systems (WDH orders/ Networks/ Jobs)*.

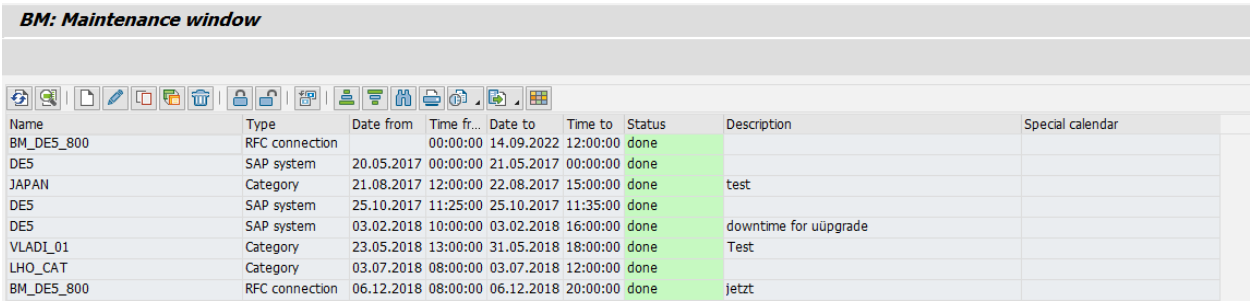
Program /BTCMAN/M_RFC executes the adjustment of the target systems used in the job processes and the RFC connections used in other objects (job, networks and WDH orders).

Also see Switching target systems (repeating orders / networks / jobs).

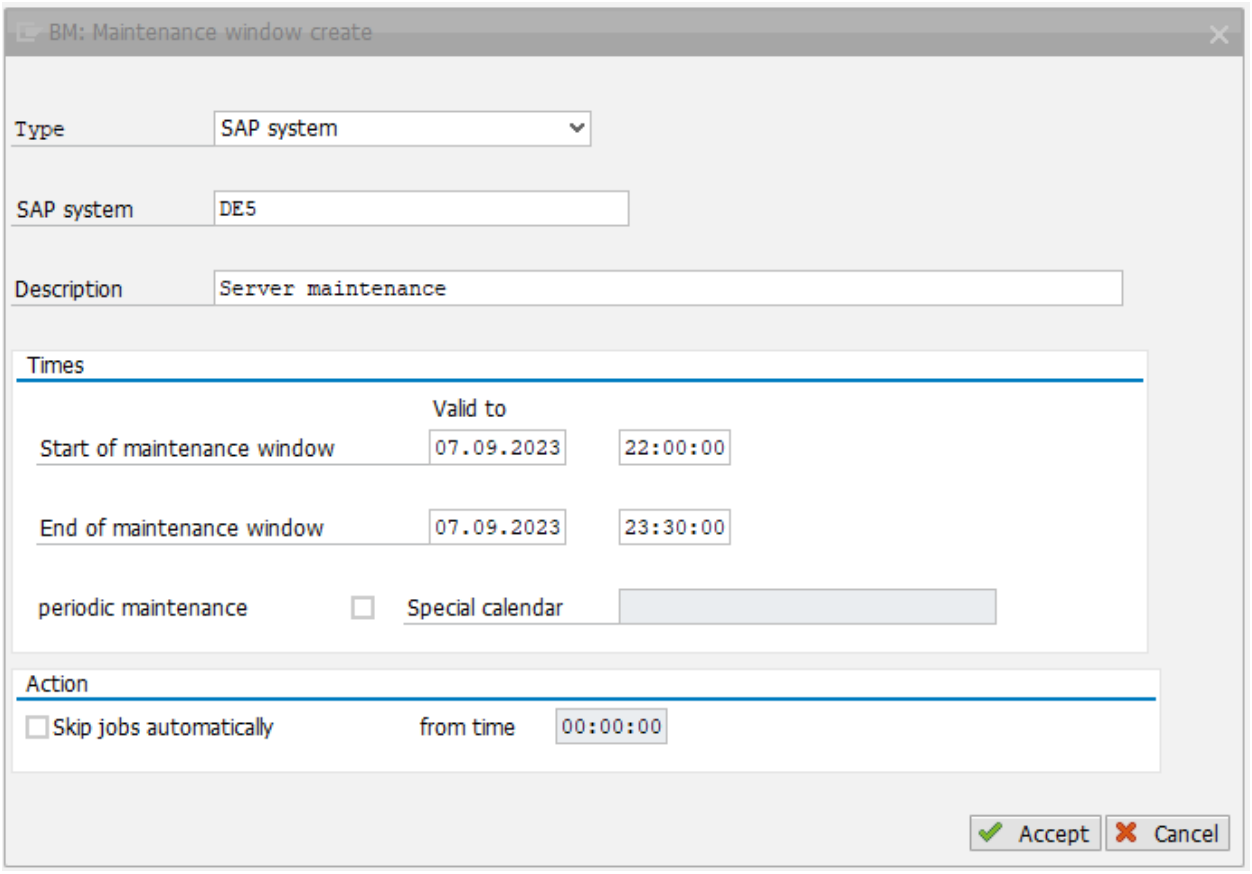
6.4.4. Define Maintenance Time Frames

Since not every system is available all the time (e.g. due to system maintenance), so-called maintenance windows can be defined in BatchMan in transaction /BTCMAN/M_P15.

This allows you to delay or skip the start of Batchman objects.



Name	Type	Date from	Time fr...	Date to	Time to	Status	Description	Special calendar
BM_DE5_800	RFC connection		00:00:00	14.09.2022	12:00:00	done		
DE5	SAP system	20.05.2017	00:00:00	21.05.2017	00:00:00	done		
JAPAN	Category	21.08.2017	12:00:00	22.08.2017	15:00:00	done	test	
DE5	SAP system	25.10.2017	11:25:00	25.10.2017	11:35:00	done		
DE5	SAP system	03.02.2018	10:00:00	03.02.2018	16:00:00	done	downtime for uüpgrade	
VLADI_01	Category	23.05.2018	13:00:00	31.05.2018	18:00:00	done	Test	
LHO_CAT	Category	03.07.2018	08:00:00	03.07.2018	12:00:00	done		
BM_DE5_800	RFC connection	06.12.2018	08:00:00	06.12.2018	20:00:00	done	jetzt	



- Selection of the type: SAP system, RFC connection or category
- Depending on the type, eg.g. the SAP system that is to be maintained
- Times, time range of maintenance. If necessary, periodically with the help of a special calendar
- If necessary, skip jobs

  If this flag is set, jobs that would normally start at the maintenance window time will not run.

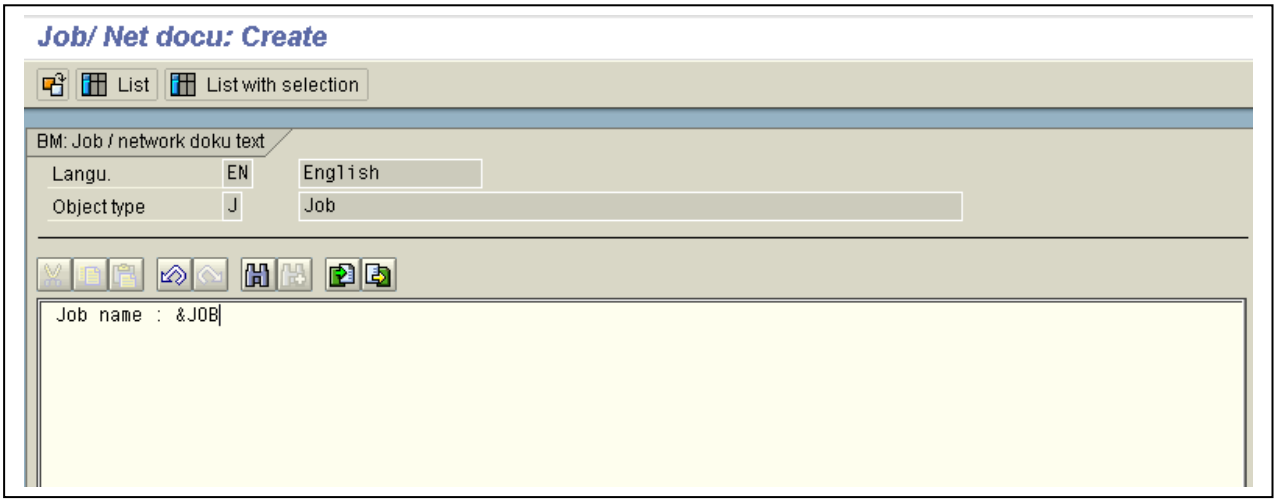
  If the flag is not set, these jobs will start after maintenance window.

6.4.5. Text Templates

Function tree BatchMan → Customizing → Extended Customizing → Text Templates

In the **info text templates** section, you can define the templates that are used for the documentation of library objects (jobs and networks) in a language-dependent and object-dependent manner. In the change mode, the SAP Editor is started.

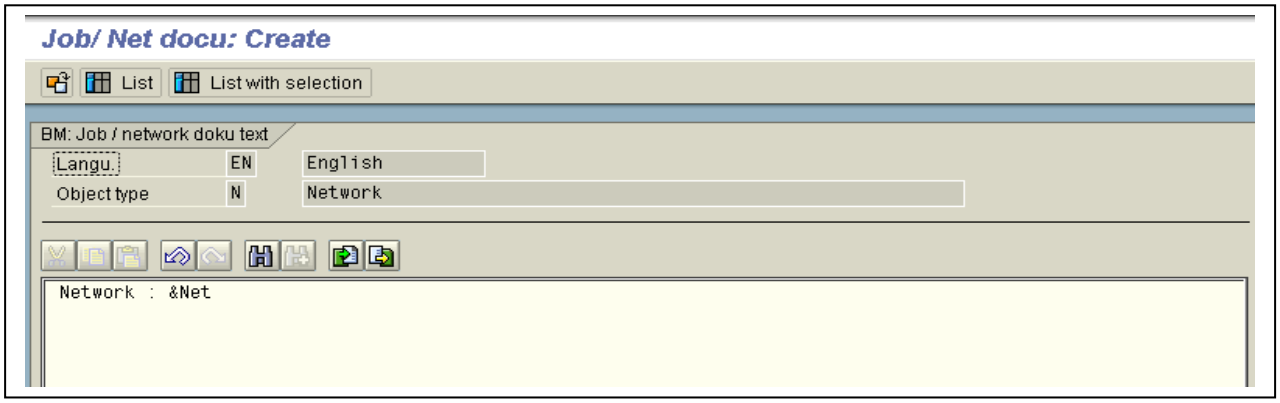
Example for language ‚EN' and object type ‚J' (like job) enter, and Create or Change.



Adjust to own requirements. This template is then used in every first call of the internal job documentation. There are two usable variables:

&JOBNAME	Job name
&USERNAME	User who creates the job

In the editor window, you may transfer these into the text using the context menu (right mouse button) via "Insert placeholder".



Example for language ‚EN' and object type ‚N' (like network) enter, and Create or Change.

Adjust to own requirements. This template is then used in every first call of the internal network documentation. There are two usable variables:

&NETWORK NAME	Network name
&USERNAME	User who creates the object

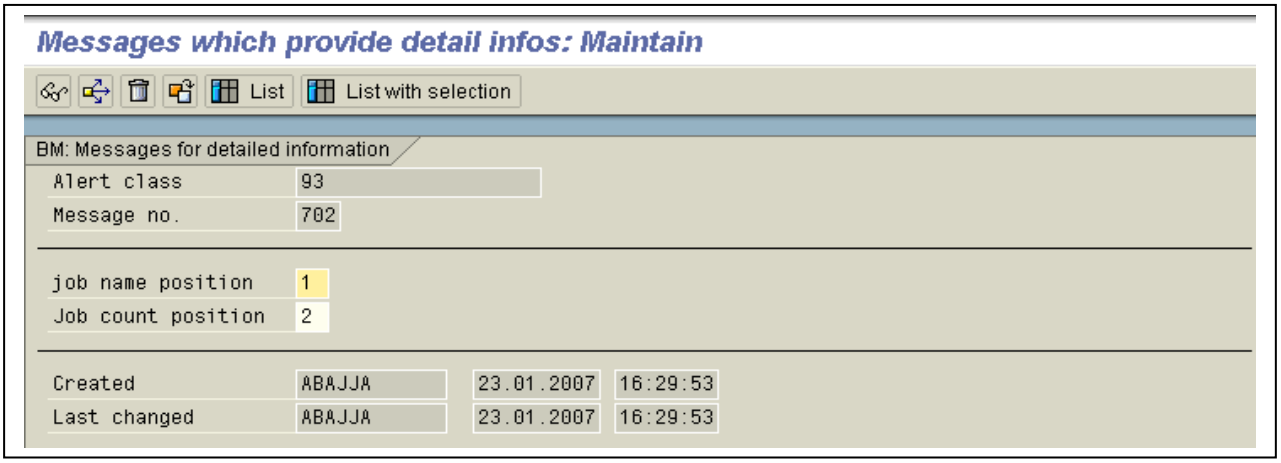
In the editor window, you may transfer these into the text using the context menu (right mouse button) via "Insert placeholder".

6.4.6. Message numbers for detailed informations

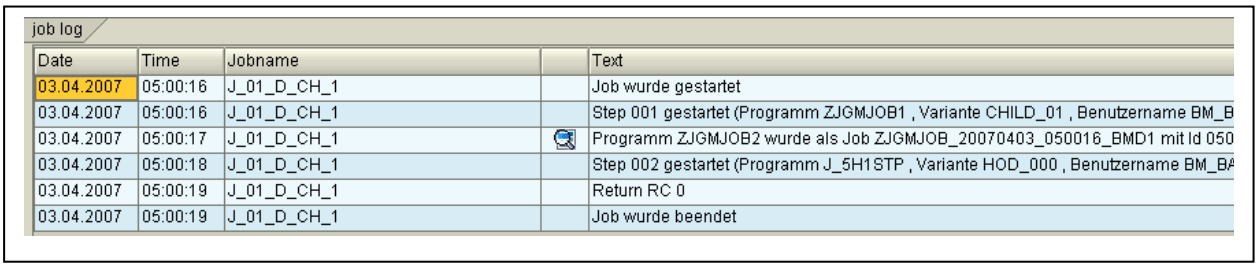
Functiontree BatchMan → Customizing → Exteded Customizing → Message Numbers

Within the SAP system, some ABAP/4 programs exist that schedule jobs (so-called child jobs) in the program. These can be identified in the job log/protocol as entry with message class and message number.

In this table, the message classes and message numbers are entered if you want more information in monitoring.



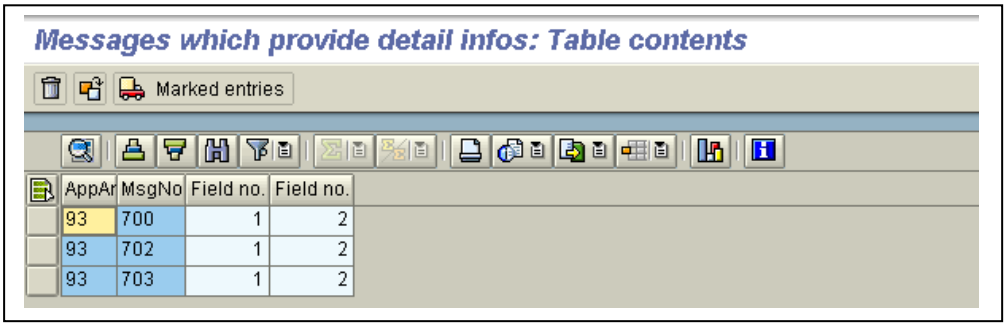
In Monitoring → Joblog, the messages are then displayed with an icon.



This icon is defined as a link. When you click the button, the system branches to the job log of the child job.

All defined messages in the table are displayed via the list.

The messages from message class /BTCMAN/93 are delivered in the transport for the **BatchMan** table content.



6.5. Load Balancing Customizing

6.5.1. Define Interfaces

The definition of an interface is described in the BatchMan interface manual BM4_e_InterfaceHB.pdf.

6.5.2. Authorization Maintenance User jobs

The definition of an interface is described in the BatchMan interface manual.

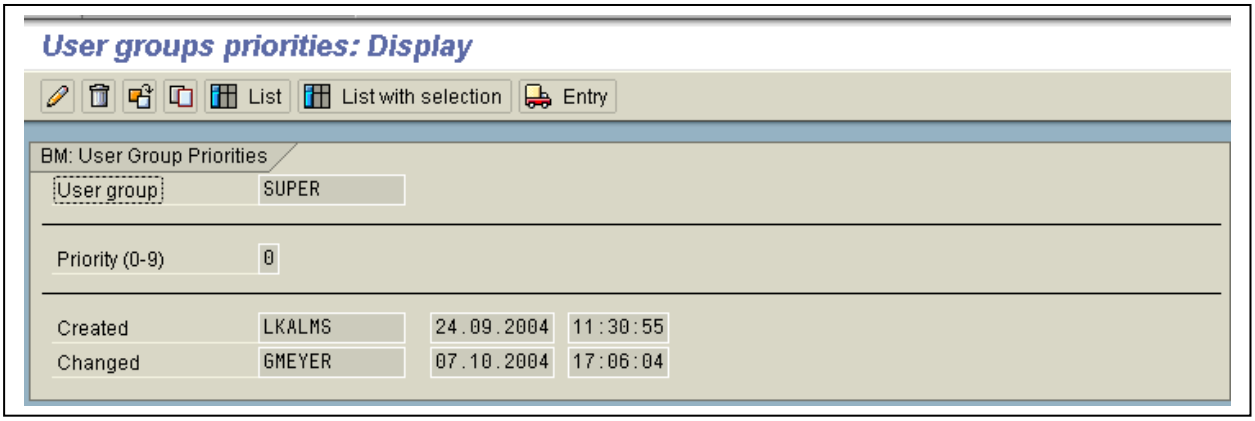
6.5.3. Setup Job Interface

The significance of all setup parameters including the archiving for the interfaces and user jobs is described in the BatchMan interface manual.

6.5.4. Priority of User Groups

Function tree BatchMan → Customizing → Load Balancing Customizing –Priority of user groups

Priorities are assigned to user groups here. ‚0' (zero) is the highest priority, and ‚9' (nine) the lowest priority.

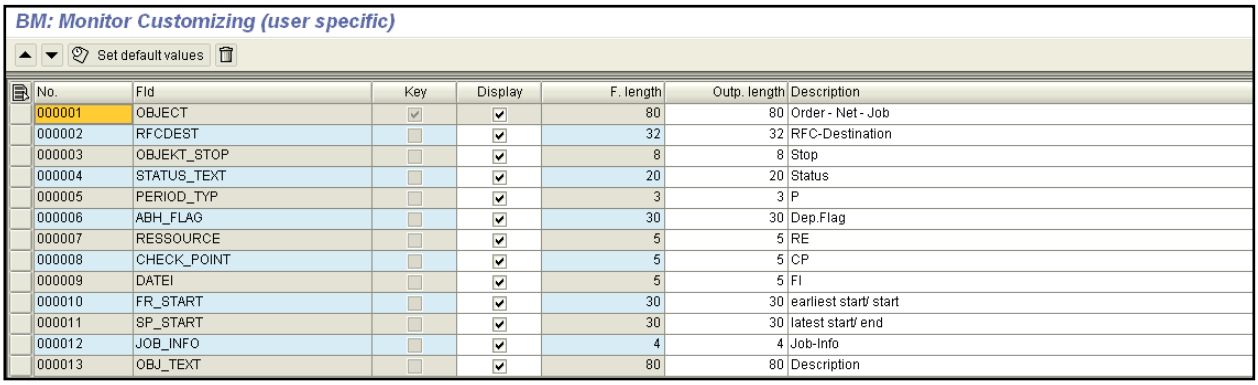


This function is used for the so-called user jobs within BatchMan (also see the BatchMan interface manual).

6.6. User Customizing

6.6.1. Monitor Customizing

Call the maintenance of the user-specific column settings for monitoring.


BM: Monitor Customizing (user specific)

Set default values

No.	Fld	Key	Display	F. length	Outp. length	Description
000001	OBJECT	☑	☑	80	80	Order - Net - Job
000002	RFCDEST	☐	☑	32	32	RFC-Destination
000003	OBJEKT_STOP	☐	☑	8	8	Stop
000004	STATUS_TEXT	☐	☑	20	20	Status
000005	PERIOD_TYP	☐	☑	3	3	P
000006	ABH_FLAG	☐	☑	30	30	Dep.Flag
000007	RESSOURCE	☐	☑	5	5	RE
000008	CHECK_POINT	☐	☑	5	5	CP
000009	DATEI	☐	☑	5	5	FI
000010	FR_START	☐	☑	30	30	earliest start/ start
000011	SP_START	☐	☑	30	30	latest start/ end
000012	JOB_INFO	☐	☑	4	4	Job-Info
000013	OBJ_TEXT	☐	☑	80	80	Description

With this program, you can change the setting of the columns in monitoring (see section 4). The number indicates at which position the system displays the field as column in the order monitor. Number “000001” is the column on the far left.

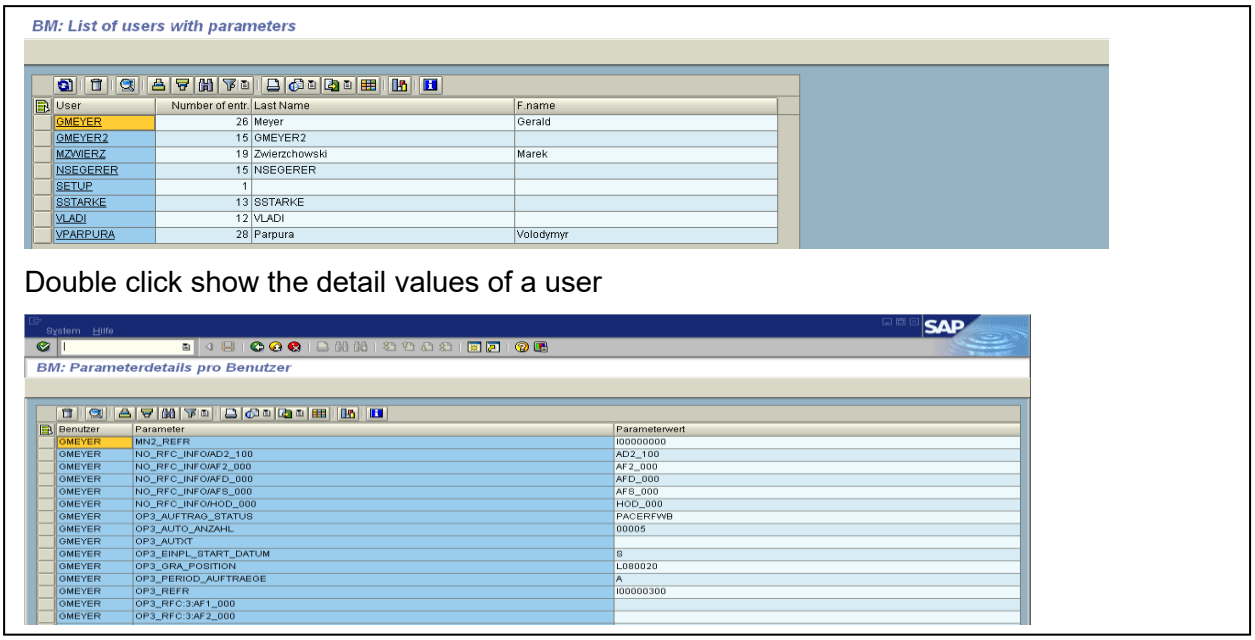
You can **display** or hide fields (and thus the columns). You can modify the **output length** (width) of the columns.

If you mark one of the lines in the tables, you can move its position up or down using the  buttons. That way, the column is moved to the left (the up button) or to the right (the down button).

6.6.2. User parameter

The parameters for other fields are saved in table /BTCMAN/M_USRPAR. This can refer either to a user (e.g. for Monitoring), or be neutral (no user).f

This only shows the selected parameters of the seperate users.


BM: List of users with parameters

User	Number of entr.	Last Name	F.name
GMEYER	26	Meyer	Gerald
GMEYER2	15	GMEYER2	
MZWIERZ	19	Zwierzchowski	Marek
NSEGERER	15	NSEGERER	
SETUP	1		
SSTARKE	13	SSTARKE	
VLADI	12	VLADI	
VPARPURA	28	Parpura	Volodymyr

Double click show the detail values of a user

BM: Parameterdetails pro Benutzer

Benutzer	Parameter	Parameterwert
GMEYER	MN2_REFR	I00000000
GMEYER	NO_RFC_INFO/AD2_100	AD2_100
GMEYER	NO_RFC_INFO/AF2_000	AF2_000
GMEYER	NO_RFC_INFO/AFD_000	AFD_000
GMEYER	NO_RFC_INFO/AFS_000	AFS_000
GMEYER	NO_RFC_INFO/HOD_000	HOD_000
GMEYER	OP3_AUFTRAG_STATUS	PACERFWB
GMEYER	OP3_AUTO_ANZAHL	00005
GMEYER	OP3_AUTXT	
GMEYER	OP3_EINPL_START_DATUM	S
GMEYER	OP3_GRA_POSITION	L080020
GMEYER	OP3_PERIOD_AUFTRAEGE	A
GMEYER	OP3_REFR	I00000300
GMEYER	OP3_RFC:3:AF1_000	
GMEYER	OP3_RFC:3:AF2_000	

7. Tool Box

The tools (programs) listed here are to help make the batch processing more comfortable. They provide functions that are not included in the SAP system or can only be executed in the dialog mode.

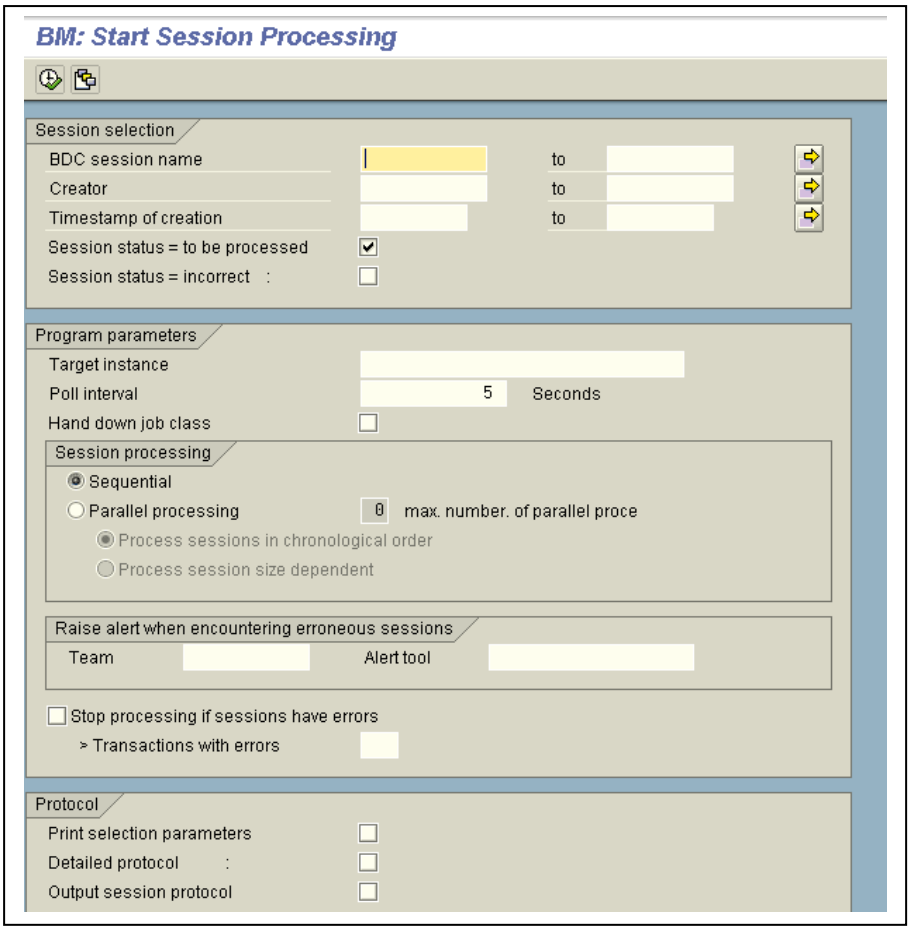
7.1. BatchMan Tools

7.1.1. Batch Input Session Submitter

In the SAP- standard and in other batch systems a job always finishes successfully when it has triggered the session processing. A negative session end does not cause a standstill in the background processing, since this status/condition will never occur!

This additional program generates background jobs (so-called child jobs) for the BTCI- session processing. The program /BTCMAN/C_JOB monitors the job - and session status. This characteristic/feature makes result-dependent control of background processing possible.

Please note: This program must be scheduled in the system in which the batch input session should be processed.



Selection of the sessions can be done **status-dependent** (Status to be processed and/or faulty). **Locked** sessions are ignored. Depending on the number of to-be-processed sessions, an appropriate number of jobs is generated which can be triggered **parallel** or **sequentially (chronologically).**

In sequential processing, the sessions are process according to the FIFO principle.

In parallel processing, the maximum number of the parallel-to-be-started processes can be specified. In addition, you can specify if the sessions should be process according to the FIFO-principle or according to session size (largest session first).

The session jobs can be transferred to a certain server/ certain instance via the **target system**.

The **Poll interval** can be set up on a scale from 1 (one) up to 5 (five) seconds. This is the time interval in which the job checks the current session status (function module RZL_SLEEP).

The job class of the session job is “C” by default. Since report /BTCMAN/C_JOB is scheduled in a higher job class, it is possible to **pass on** its **job class** to the session jobs.

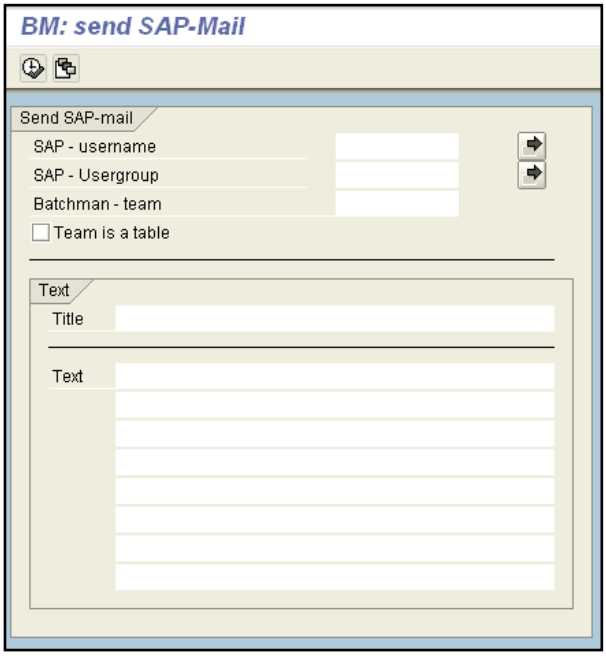
7.1.2. Send SAP Mail – program (/BATMAN_ML3)

Depending on the recipients selected, an SAP e-mail or an Internet e-mail is sent.

SAP user: SAP e-mail

SAP user group: SAP e-mail

BatchMan team: Depends on the Customizing of the teams or time tables.

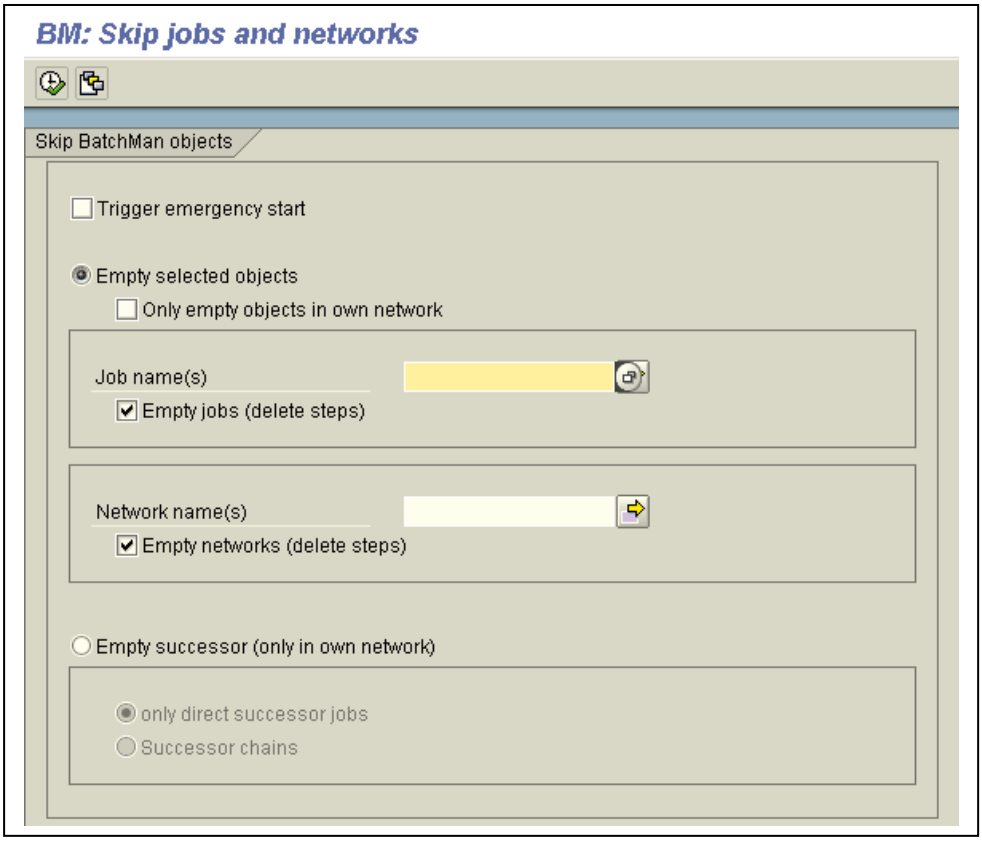


7.1.3. Skip and Start Objects

The program /BTCMAN/M_SKP is used for skipping and starting jobs which can not run on account of the start criteria. It is also possible to schedule the report as recovery job of an aborted job, if, on account of the abort, successor jobs should not be executed anymore.

Skipping means that the status of the jobs is changed to ‚skipped’ within **BatchMan**. If this job is next in line within the processing chain, then the job is skipped. Only the status is set on ‚skip/fin’ within **BatchMan**. Consequently, the job is not executed physically anymore.

The execution of the report /BTCMAN/M_SKP must always be included in the orders (STD; INDIV, WDH), because a skipping action is only possible in a current order.



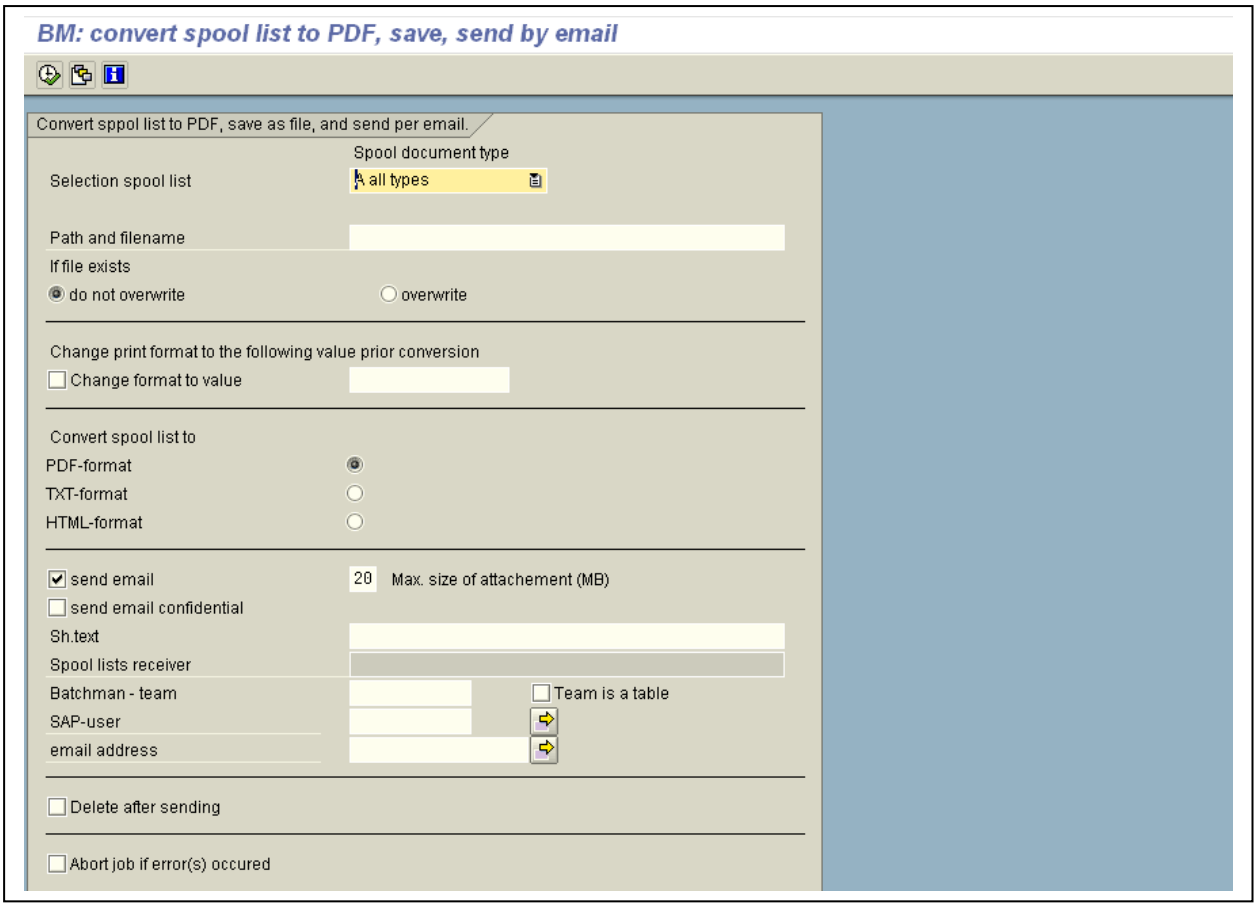
The program can be started with two main options:

- Empty selected objects. The objects specified in the job names and network names are skipped.
  - **Empty jobs/networks:** The system skips all objects that correspond to the name specified within the order.
  - **Only empty objects in own network:** The system skips only jobs/networks that correspond to the name specified within the network in which report /BTCMAN/M_SKP is running. Otherwise it skips those within the entire order.
- Empty successor: You can only execute this function within the same network in which report /BTCMAN/M_SKP is running.
  - **Only direct successors:** The system only skips the objects (jobs/networks) that directly follow the terminated job for which report /BTCMAN/M_SKP was scheduled as recovery job.
  - **Successor chains:** The system skips all objects that follow the terminated job (also the successors of the skipped objects etc.) for which report /BTCMAN/M_SKP was scheduled as recovery job.
- Trigger events: Should remain empty so that the objects in the processing chains are started at the right time. If the flag is set, the system starts the job immediately after skipping. Subsequent objects that are not empty would be started afterwards. If these are used for subsequent processing, an incorrect processing sequence arises.

This may be used to start jobs that cannot yet run due to start criteria (corresponds to the emergency start in the order monitor)

7.1.4. Convert Spool List for Job

The program converts spool list(s) of the type ABAP into the PDF format or HTML or TXT, saves these and/or sends the converted documents via e-mail.



The program must run after the steps that generate the spool orders, and it only runs in the background. A spool list is **converted** depending on the selection. This may occur in the formats TXT, PDF or HTM. The **formatting** may be adjusted.

The converted spool list is generally saved as file. The directory must be reachable by the SAP system. For the path entry, reserved words may also be used; such as:

- <HOST> - Processor name as per system field SY-HOST
- <DATE> - Date as per system field SY-DATUM
- <L= > - Logical file name
- <A= > - Logical path, etc.

One reserved word that can only be used in this report is <SPOOL_LIST_TITLE>. This way, the spool title entered for the maintenance of the print parameters is used as the file name. Analog with the reserved word <SPOOL_ID>, the ident of the spool job.
We recommend you use the reserved words (and perhaps <DATE> and / or <TIME>) in the path and file so that existing files are not overwritten.

The file may be sent as an e-mail attachment to the recipients entered. E-mail recipients may be:

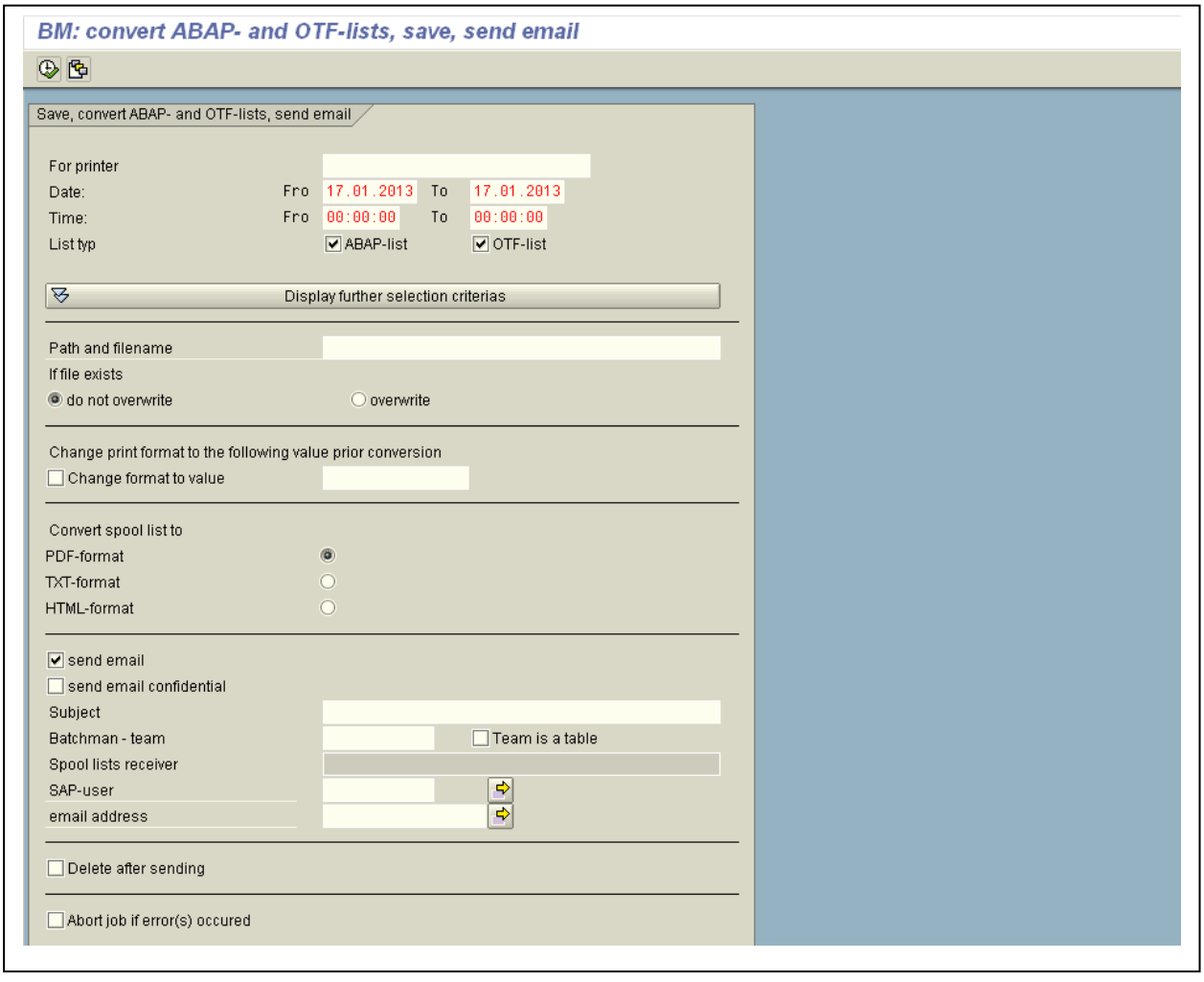
- SAP users
- E-mail addresses and/or
- a BatchMan team

If problems arise:

Use transaction SOPE to check whether or not there is an entry for the related file name extension and it is not locked. For TXT, deactivate there the flag for ASCII. If logical file names or logical paths are to be used there, these must be maintained with transaction FILE.

7.1.5. Convert Spool list for Time

The program converts spool list(s) of the type ABAP and OTF (SAP script) into the PDF format or HTML or TXT, saves these and/or send the converted documents via e-mail.



The program selects the spool orders using the time stamp from – to. Thus, it need NOT run after the steps that the spool orders generate. It also functions only in the background.

All other parameters are identical to the previous program.

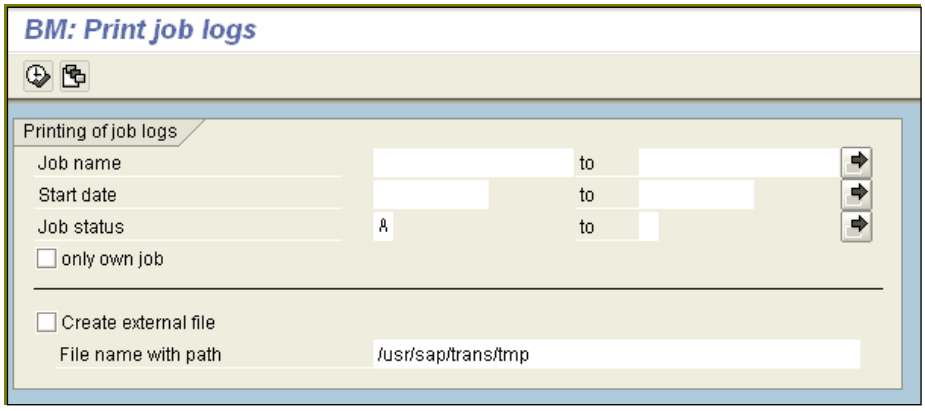
7.1.6. Output Job Logs as Spool List / File

Here you can save the job logs and print them out, if required. Prerequisite is that they are not in the TBTCO table. If you have selected “Create external file,” the file is created on the server that is logged on.

In the “File name” field, you may also use a logical file name and path. The old values with absolute specifications continue to function.

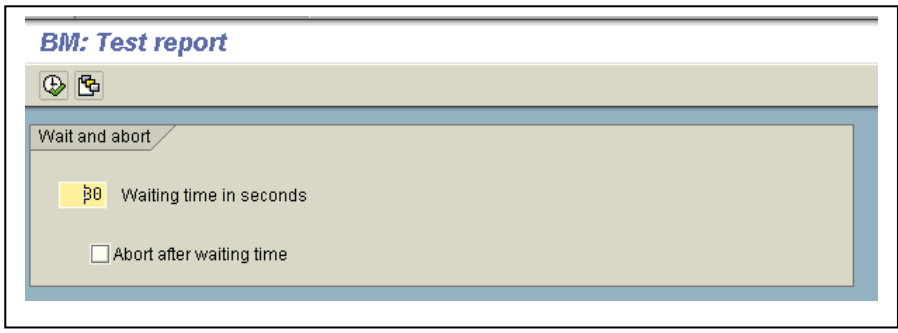
- <L= logical file> defines a logical file name (transaction FILE)
- <A=logical path> defines a logical path (transaction FILE)

Example: <A=SAVE_PATH>file.txt



You can also attach it as additional step in the job. For this, activate the “Only own job” flag. The system only outputs the log if the job does not terminate in a step prior to this report.

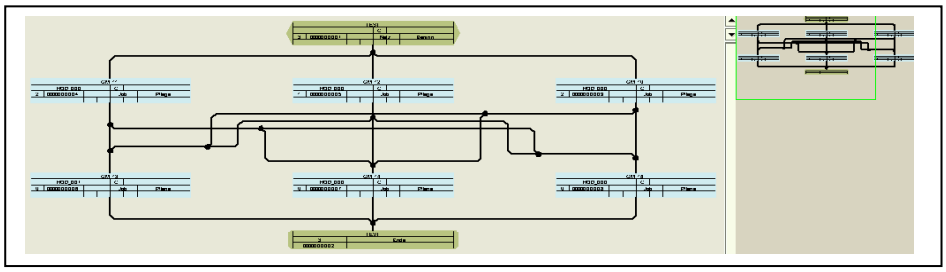
7.1.7. Dummy Report for Wait and Test



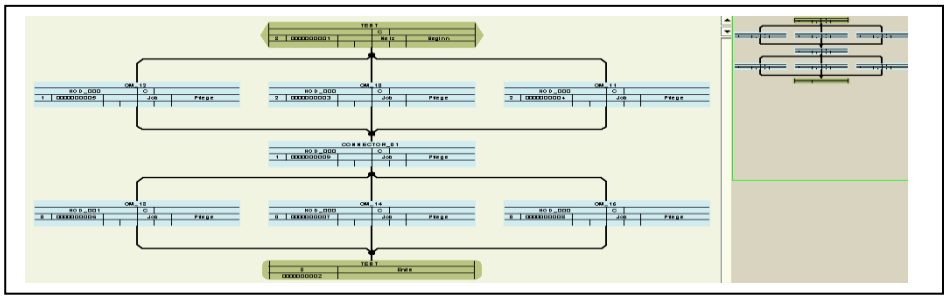
This is a small test program (/BTCMAN/C_DMY) to simulate wait times and terminations. We use this program frequently to test.

The report can be used to create jobs that are used as connectors in the graphics. This way, the graphics are formed more clearly.

Without Connector:



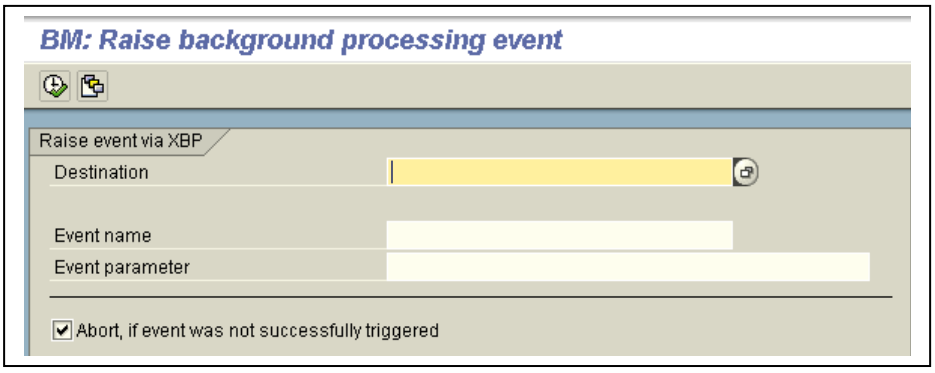
With Connector



7.2. SAP Tools I

7.2.1. Trigger Event

An event for possible subsequent/future processes can be triggered with this report (/BTCMAN/M_EVT). The RFC destination determines on which system the event is triggered; ‚NONE ‘is the local system.



7.2.2. Unschedule all Jobs

This program (/BTCMAN/C_AUS) makes unscheduling of jobs that are *released* in SAP Status possible. 'Unschedule/deallocate all jobs' will set the SAP status of these jobs to *btc_put_active* = **unscheduling/ suspend**

Job that start with ‚RDDIMPDP', will not be unscheduled/deallocated.

Please note: This program has to be scheduled on the system where the activity should be executed.

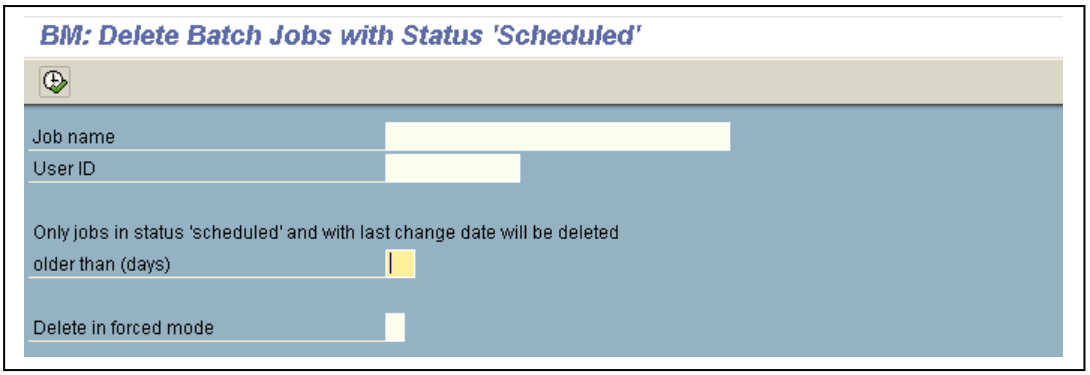
7.2.3. Schedule all Jobs

This program (/BTCMAN/C_PLA) makes scheduling of jobs possible that are in status *put_active*. „Schedule all jobs" will set these jobs to the status *released* = **Scheduling.**

Please note: This program has to be scheduled on the system where the activity should be executed.

7.2.4. Delete jobs in status ‚Scheduled'

Deleting jobs from the current client which are in status 'scheduled' in the SAP table TBTCOwith program /BTCMAN/C_DEL.



This program is based on the SAP-Program RSBTCDEL. In RSBTCDEL, the selection options for deleting jobs in the status 'scheduled' are not differentiated enough. That is the reason why this program targets especially the 'scheduled' jobs.

Selection: xx days older than the CHANGE DATE (LASTCHDATE) of the jobs in status 'scheduled'.

Please note: For the job selection in SAP standard, not the last change date is used but the scheduled (and real) execution date.

Pay attention/Note (from BP_JOB_DELETE): The forced mode in deleting will cause that the deleting of the jobs will nevertheless be continued in the case of the following errors:

- Error in deleting the step table (Exception CANT_DELETE_STEPS)
- Error in deleting the job protocol/log (Exception CANT_DELETE_JOBLOG)
- Error in deleting the scheduler entry (Exception CANT_DELETE_TIME_ENTRY and/or CANT_DELETE_EVENT_ENTRY)

All other errors must lead to an abort of the deleting process; otherwise inconsistencies will occur within the job administration.

Please note: This program has to be scheduled on the system where the activity should be executed.

7.2.5. Search in Spool Text / Reprint Spool Request

This program (/BTCMAN/C_SP4) must run in batch mode. The flag "Delete after output" may not be set in the original print parameter. Otherwise this report can not identify a spool order. Parameter description:

Parameter description:

Find termination text in spool: In the spool list, the system searches for a text string. If one is found, the program is ended with an E message.

Send spool lists to printer: The system sends the spool lists to the printers of your choice.

Termination text to be found: This is the text string the system is to find. For the “Print spool lists” variant, the text string is not required.

Program name: This is the name of the ABAP program that the spool list generated.

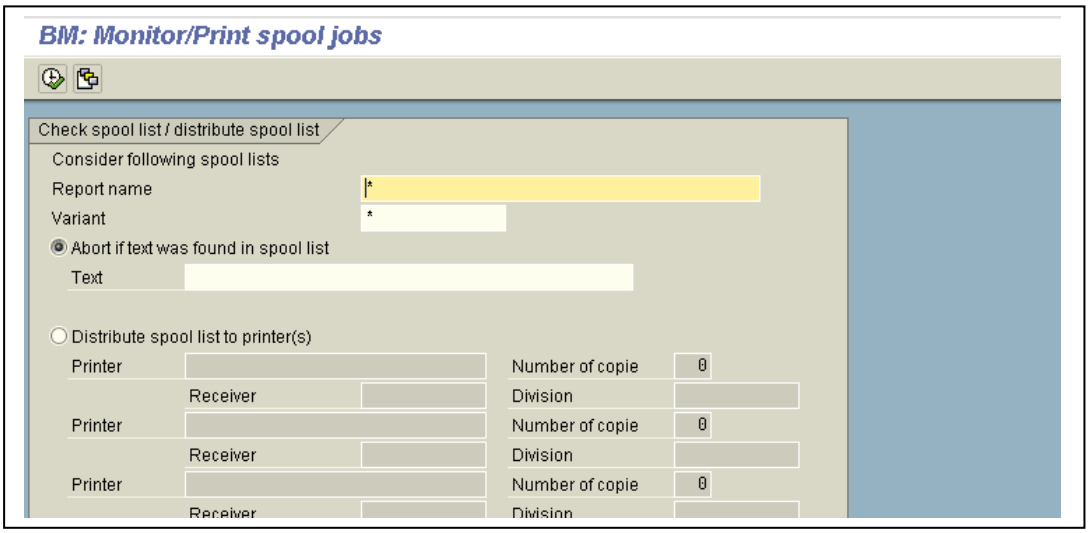
Variant of the ABAP: Name of the variant with which ABAP was started.

Name for printers 01 to 10: This is the name of a printer in the system to which the spool list is to be sent. For the variant to find termination text, these entries are not required (as well as all subsequent entries).

Number of copies for printers 01 to 10: This is the number of the spool list printouts for the individual printers.

Recipient for printers 01 to 10: This is the recipient name for the spool list printout of the individual printers.

Division name for printers 01 to 10: This is the name of the recipient division of the individual printers.



Please note: This program has to be scheduled on the system where the activity should be executed.

7.2.6. Suche Text in Joblog/Syslog/Spoolliste und Aktion

With the report /BTCMAN/C_TXS you search for messages or texts in

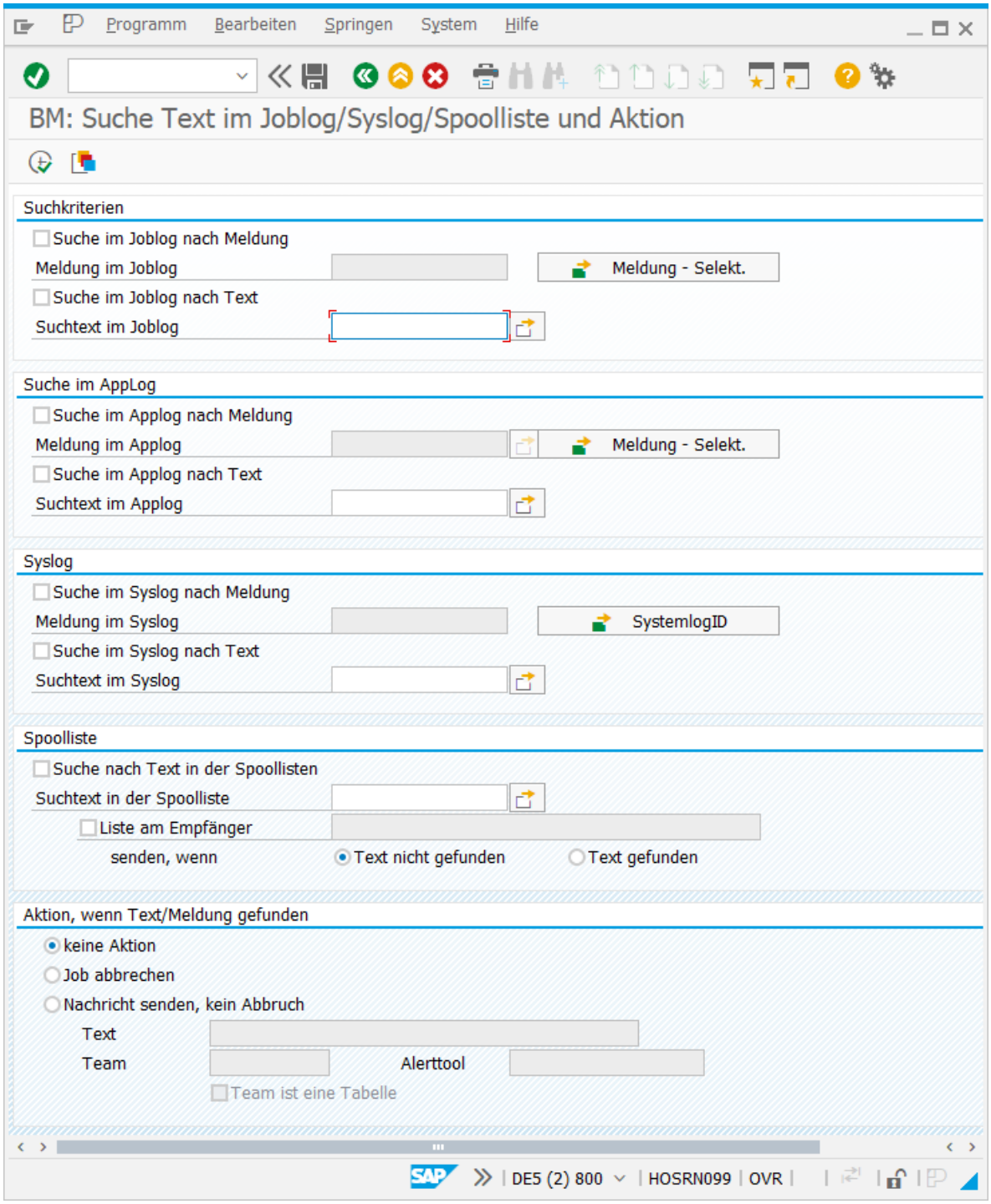
- Job logs,
- Application logs,
- System logs,
- Spool lists.

In the event of a find or non-find, you can

- send an email,
- quit the job or
- do not perform any action

The report /BTCMAN/C_TXS only runs in the background. Steps that are not to be searched are placed after the report step.

Since BatchMan 5.1.2, the report can be used as a separate job and then searches the predecessor. This is important, for example, if a job of a different type was previously running, e.g. an FDM job.


Programm
Bearbeiten
Springen
System
Hilfe
BM: Suche Text im Joblog/Syslog/Spoolliste und Aktion
Suchkriterien
Suche im Joblog nach Meldung
Meldung im Joblog
Meldung - Selekt.
Suche im Joblog nach Text
Suchtext im Joblog
Suche im AppLog
Suche im Applog nach Meldung
Meldung im Applog
Meldung - Selekt.
Suche im Applog nach Text
Suchtext im Applog
Syslog
Suche im Syslog nach Meldung
Meldung im Syslog
SystemlogID
Suche im Syslog nach Text
Suchtext im Syslog
Spoolliste
Suche nach Text in der Spoollisten
Suchtext in der Spoolliste
Liste am Empfänger
senden, wenn
Text nicht gefunden
Text gefunden
Aktion, wenn Text/Meldung gefunden
keine Aktion
Job abbrechen
Nachricht senden, kein Abbruch
Text
Team
Alerttool
Team ist eine Tabelle
DE5 (2) 800
HOSRN099
OVR

7.2.7. Delete Joblogs without TemSe-entry

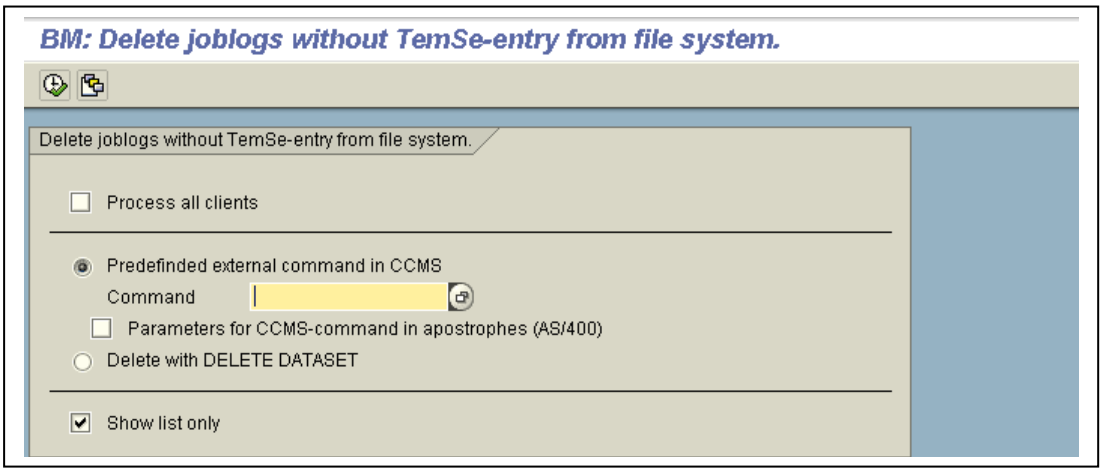
Through a client copy, restoring a data saving, data base copy or deleting a client, inconsistencies can occur in the TemSe.

Possible inconsistencies:

- Files are missing (File system) for the entries in the TemSe: CORRECTION: The program **RSTS0020** (TemSe consistency checks) deletes the orphaned/neglected TemSe objects
- Files exist that do not have a TemSe entry: CORRECTION: Execute **/BTCMAN/C_TS1**.

  This program (/BTCMAN/C_TS1) reads client-wide all job logs on the file system level and checks if each job log has the corresponding TemSe entry and job (TBTCO). Orphaned job logs are deleted on file system level.

  Please note: This program only processes job logs. SAP deposits printing orders in DB tables, and not on file system level.



Process all clients: inactive → only the current client is cleaned up/adjusted

active → all clients are processed

Display only list: active → only the list is displayed

inactive → processing is triggered

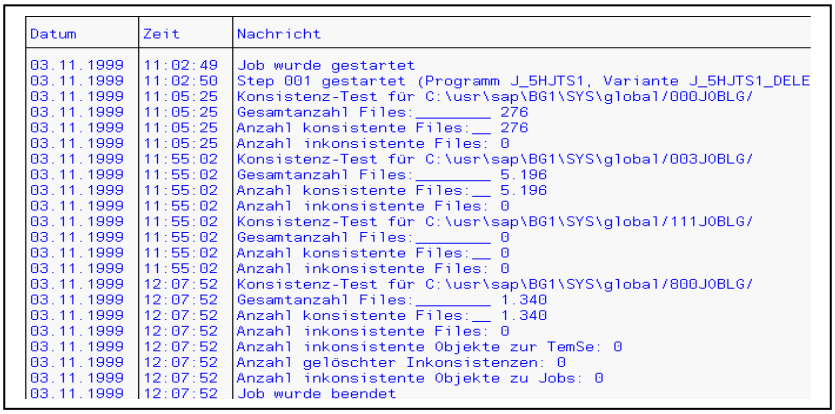
Deleting can be done in two alternative ways:

- With acommand defind in CCMS (transaction SM69) or
- With a DELETE DATASET command

The program runs a bit longer and therefore should only run in batch.

TEST: all clients ➔ Joblog

Please note: This program has to be scheduled on the system where the activity should be executed.



Datum	Zeit	Nachricht
03.11.1999	11:02:49	Job wurde gestartet
03.11.1999	11:02:50	Step 001 gestartet (Programm J_5HJTS1, Variante J_5HJTS1_DELE
03.11.1999	11:05:25	Konsistenz-Test für C:\usr\sap\BG1\SYS\global/000J0BLG/
03.11.1999	11:05:25	Gesamtanzahl Files:_____ 276
03.11.1999	11:05:25	Anzahl konsistente Files:__ 276
03.11.1999	11:05:25	Anzahl inkonsistente Files: 0
03.11.1999	11:55:02	Konsistenz-Test für C:\usr\sap\BG1\SYS\global/003J0BLG/
03.11.1999	11:55:02	Gesamtanzahl Files:_____ 5.196
03.11.1999	11:55:02	Anzahl konsistente Files:__ 5.196
03.11.1999	11:55:02	Anzahl inkonsistente Files: 0
03.11.1999	11:55:02	Konsistenz-Test für C:\usr\sap\BG1\SYS\global/111J0BLG/
03.11.1999	11:55:02	Gesamtanzahl Files:_____ 0
03.11.1999	11:55:02	Anzahl konsistente Files:__ 0
03.11.1999	11:55:02	Anzahl inkonsistente Files: 0
03.11.1999	12:07:52	Konsistenz-Test für C:\usr\sap\BG1\SYS\global/800J0BLG/
03.11.1999	12:07:52	Gesamtanzahl Files:_____ 1.340
03.11.1999	12:07:52	Anzahl konsistente Files:__ 1.340
03.11.1999	12:07:52	Anzahl inkonsistente Files: 0
03.11.1999	12:07:52	Anzahl inkonsistente Objekte zur TemSe: 0
03.11.1999	12:07:52	Anzahl gelöschter Inkonsistenzen: 0
03.11.1999	12:07:52	Anzahl inkonsistente Objekte zu Jobs: 0
03.11.1999	12:07:52	Job wurde beendet

7.3. SAP Tools II

7.3.1. Monitor Update Entries

The program (/BTCMAN/C_ONI) monitors update orders which were created before program start. For the selection, a time window can be specified and/or updated orders of the certain reports or of a certain transaction can be considered. This way an update-dependent scheduling is made possible.

Depending on the parameter, the update monitor monitors one or several update task (UT) in a time period. The update monitor waits until the selected update tasks are processed or aborted faulty.

Processed means:

The U1- updates were processed without errors

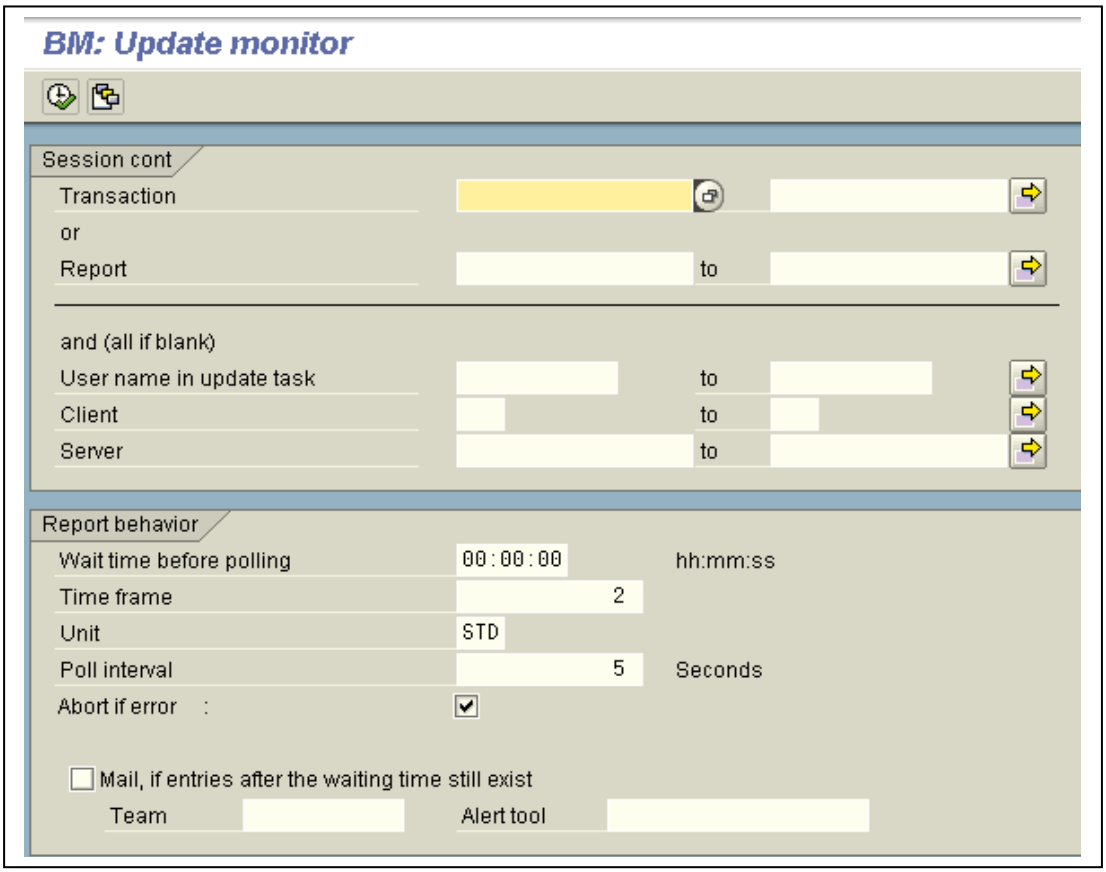
U1- updated contain time-critical data

If a report is indicated, only UT's created by this report are monitored during execution. Analog in transactions.

Via the **server** and/or **Client**, additional restrictions can be specified. The selection can also be restricted through **TIME FRAMES**. TIME FRAME is defined through **UNIT**. UNIT can accept/be the values: '**SEC**', '**MIN**', '**HOUR**' and '**DAY**'.

If TIME FRAME and UNIT is specified, then only UT's are considered that exist in a time frame. The time frame is defined through:

The **poll interval** can be set up on a scale from 1 (one) to 5 (five) seconds. This is the time interval in which the job checks the update. Before the program starts to monitor the entries in the updated table, a **waiting period** can be specified.



The update monitor has finished successfully when all selected UT's have reached the following status:

OK, update executed (U1- and U2- update) or

V1- update executed.

In case of unsuccessful execution, an alert can be triggered.

Return codes of the update monitor:

*&	0	= OK
*&	1	= U1-update executed
*&	2	= Error during insertion of table VBHDR
*&	3	= Error during reading of VBHDR
*&	4	= Dump during update
*&	5	= Commit failed
*&	6	= Error during reading of table VBMOD
*&	7	= Error during update of VBMOD
*&	8	= did not get memory/storage
*&	9	= Updating was aborted
*&	10	= No screen-working area
*&	11	= Incorrect UT mode
*&	12	= data record stopped in condition run
*&	13	= data record stopped in condition del
*&	14	= Incorrect UT parameter
*&	15	= DB deadlock during update
*&	101 - 122	= update was aborted
*&	200	= Error during U2 update
*&	251	= data record was deleted
*&	252	= V2 update is running
*&	253	= V1 update is running
*&	254	= data record is updated later
*&	255	= not executed yet

7.3.2. Monitor Lock Entries

You can monitor locked entries in the SAP program with this program (/BTCMAN/M_LCK). This program can also monitor entries in remote systems. The RFC destination must be indicated for this. NONE is the local system.

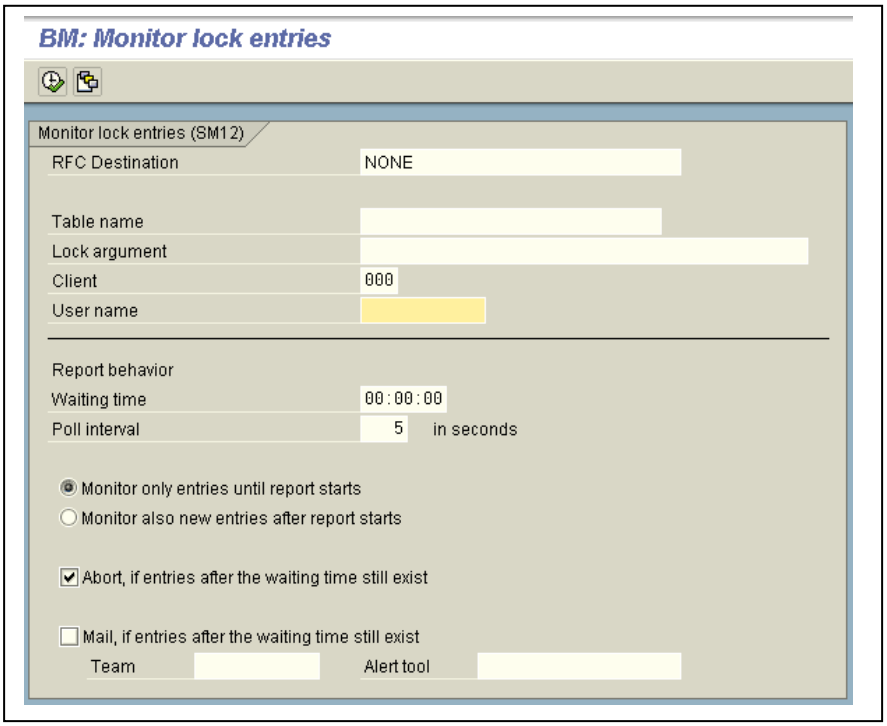
Via the **table name** and the **Lock argument** you can specify what should be monitored. This should also be specified in multiple client systems in order to limit the selection.

The **User** is the one who caused the lock entries.

The **Poll interval** can be set up in the range of 1 (one) to 5 (five) seconds. This is the time interval in which the job checks the update.

The **Wait time** is in this case the total time which the program waits to execute the monitoring. If entries still exist after the wait time, then the program is finished; if active flag ‚Abort' negative.

As standard, entries are only monitored until program start. However, the execution can also be set up that entries that are added during program run are also monitored.



In the case of unsuccessful execution, that means entries still exist after the wait time, an alert can be triggered.

Please note: The report can also read lock entries from other systems via the RFC destination. In this case, however, the entries read there are compared to the local system time. You have to make sure that the system times are the same

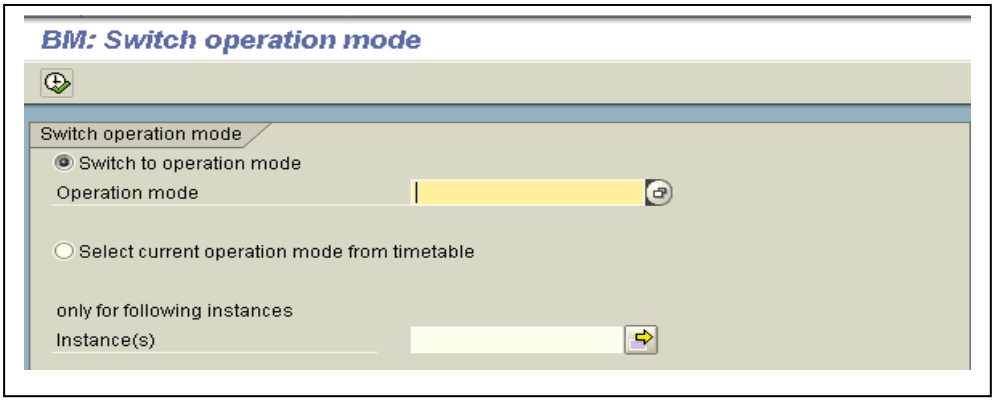
For safety reasons, the program then should be scheduled in the system in which the activity should be executed.

7.3.3. Switch Operation Mode

You can switch operating modes with this program (/BTCMAN/C_SOM). An operating mode defines the division of the different services in the SAP system.

1) Selected operating mode or
2) current operating mode from the time table

For option 1), the operating mode can only be switched for certain instances. This is not possible for Option 2).

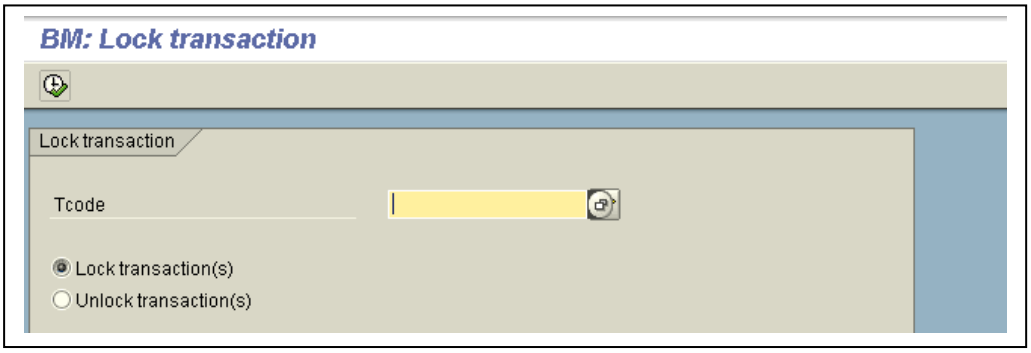


Please note: This program has to be scheduled on the system where the activity should be executed.

7.3.4. Lock Transactions

Program (/BTCMAN/C_LTR) makes is possible to lock transactions so that, during batch processing, handicaps don't occur from the dialog.

After batch processing, the transactions must be unlocked.

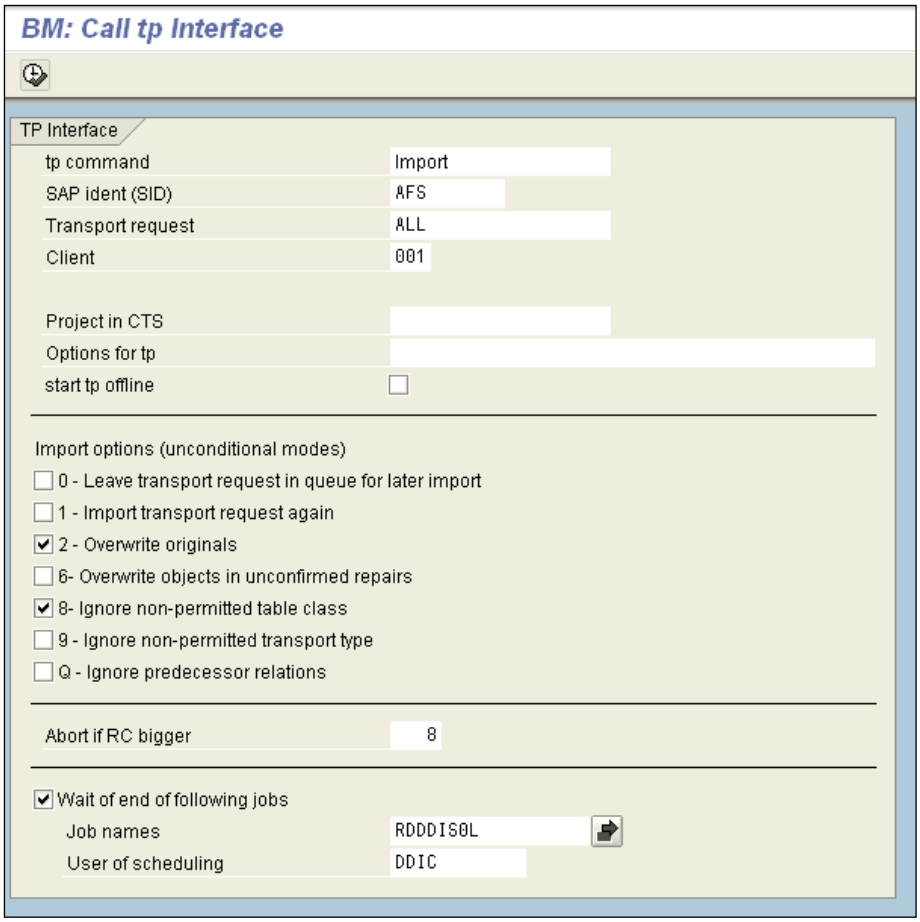


Please note: This program has to be scheduled on the system where the activity should be executed.

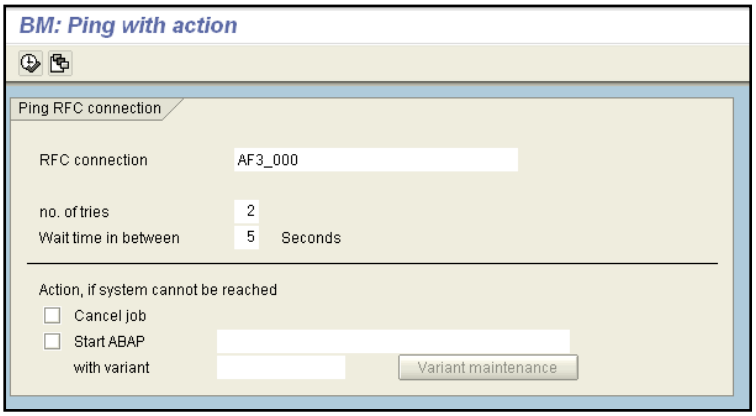
7.3.5. Little TP Interface

The program provides a tp interface to import transports into an SAP system. The individual parameters are explained in the documentation for the tp command at operating system level.

It is also possible to wait for the jobs of the order-independent steps.



7.3.6. Send Ping

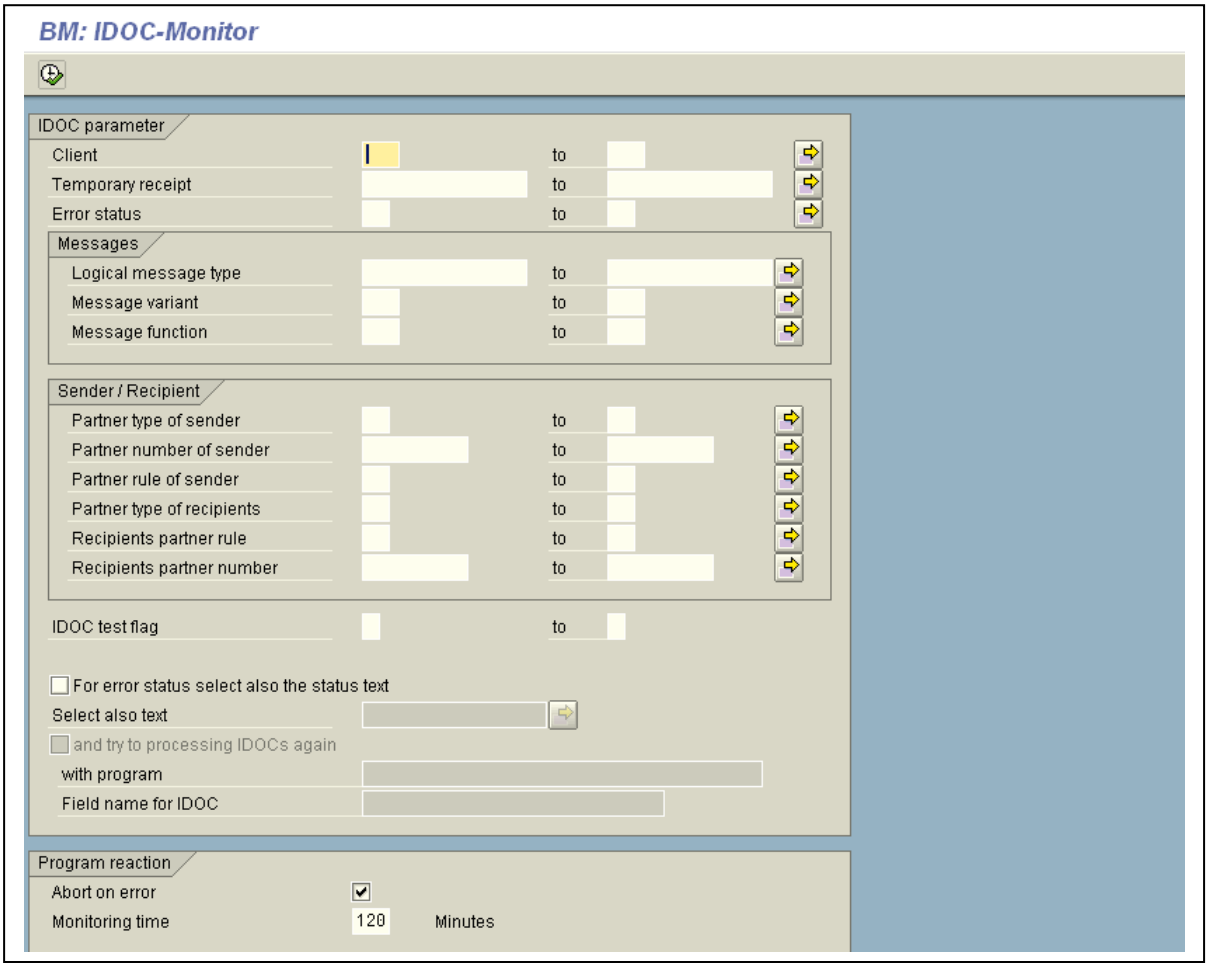


From the program, a ping command is triggered to check whether or not the target system can be reached. The ping does not indicate whether or not it is possible to log on to the system.

If the system cannot be reached, it can be terminated after a certain period to restart the same job as a recovery job; or an ABAP program (with variant) is started, e.g. to skip processes that are to run on the system.

7.3.7. Monitor IDOC entries

Program /BTCMAN/C_IDO checks whether or not all IDOCs have been processed that are used by other programs (such as RBDAPP01 - Inbound processing of IDOCs ready for transfer). Once all IDOCs are processed, the subsequent processing can be continued.

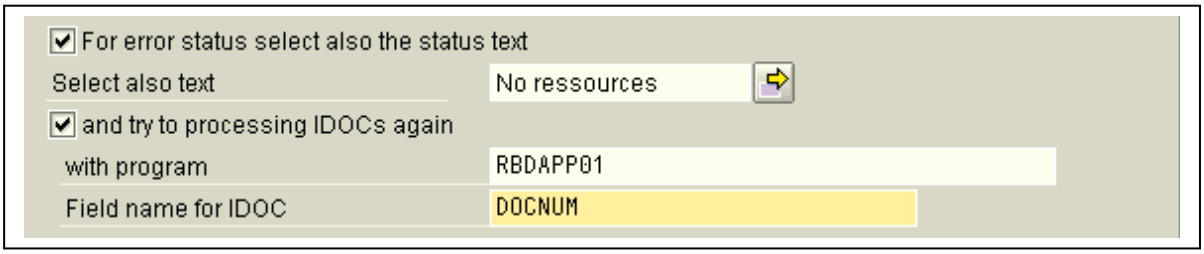


You should use the selections to restrict the amount of IDOCs selected. So that for instance the system only checks for status “64 - ready for transfer to application” whether or not all new IDOCs have been processed.

Via the monitoring time, you can set the maximum amount of time that the program will wait.

Option: “**For the error status, also select the status text**”.

A massive import of IDOCs may cause fault situations that are not actually errors. Example: “No resources, immediate processing not possible/not enough available dialog work processes”

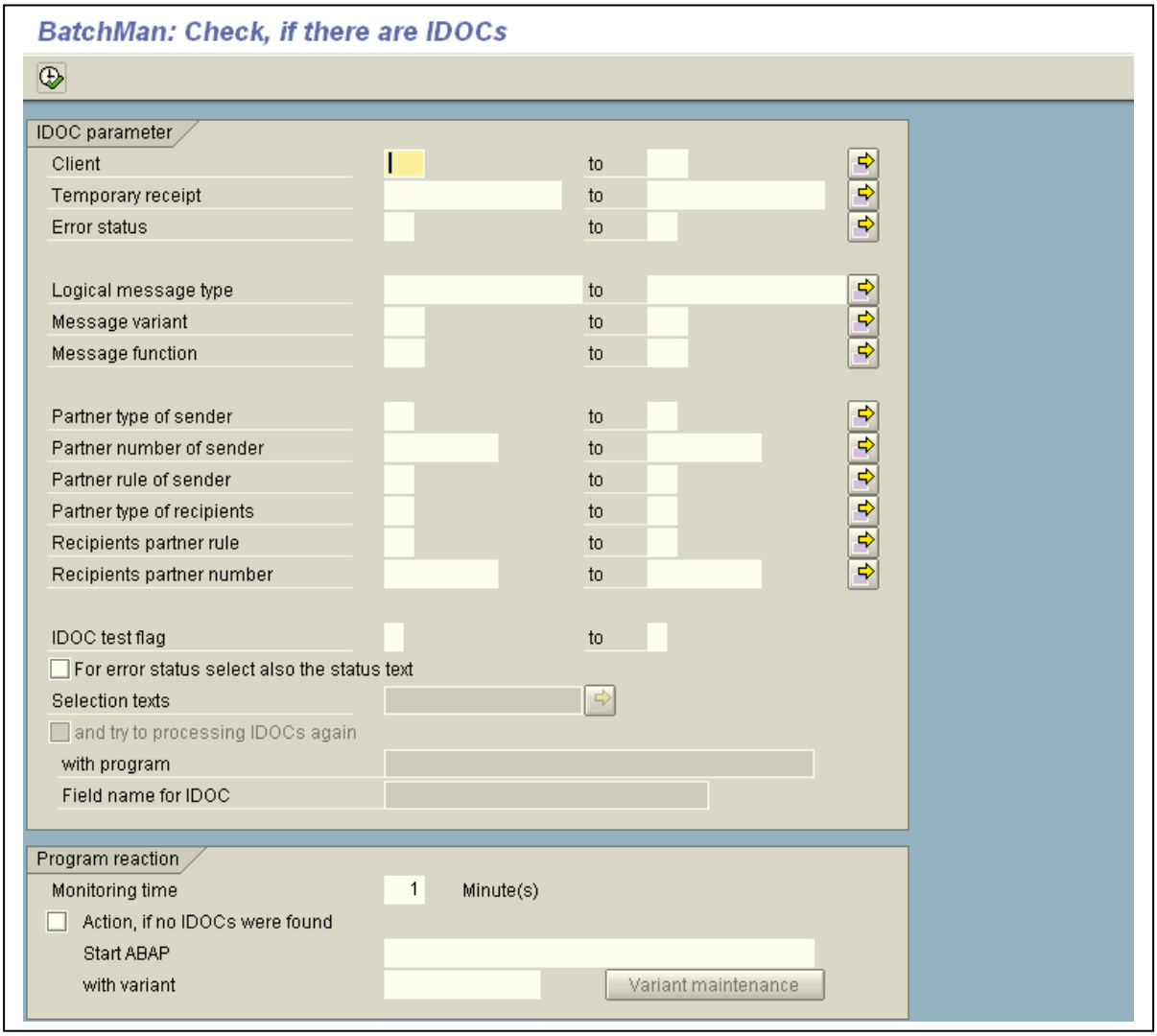


Here a search is therefore made for a part of the text (that you select from the first fault situation).

If the system finds IDOCs for that, then they are processed again by the program specified. The system individually processes the IDOCs found. Since several programs can be specified here, you are to specify the field for the IDOC number.

7.3.8. Prüfen, ob IDOC's vorhanden

Program /BTCMAN/C_ID2 checks whether or not IDOCs are available. If the system finds IDOCs, then subsequent processing can be continued. (Such as starting program RBDAPP01 - Inbound processing of IDOCs ready for transfer)



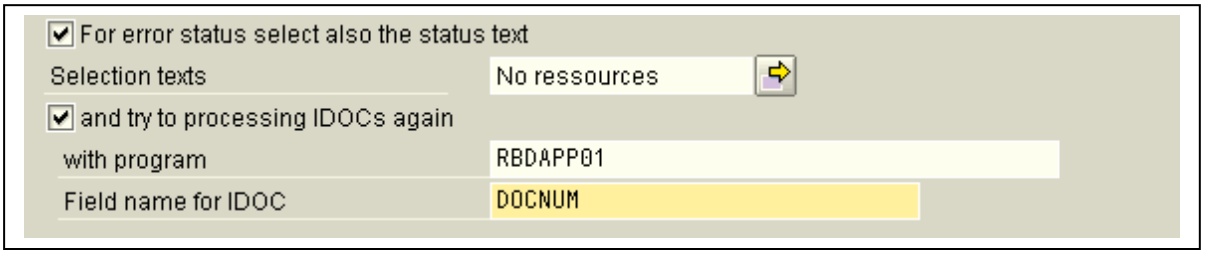
You should use the selections to restrict the amount of IDOCs selected. So that for instance the system checks whether or not all new IDOCs are to be processed for the message type and for status "64 - ready for transfer to application".

Via the monitoring time, you can set the maximum amount of time that the program will wait for the search.

If the system does not find any IDOCs, you can call a program. This can be for instance /BTCMAN/M_SKP to skip subsequent job(s) in the process.

Option: “For the error status, also select the status text”.

A massive import of IDOCs may cause fault situations that are not actually errors. Example: “No resources, immediate processing not possible/not enough available dialog work processes”



Here a search is therefore made for a part of the text (that you select from the first fault situation).

If the system finds IDOCs for that, then they are processed again by the program specified. The system individually processes the IDOCs found. Since several programs can be specified here, you are to specify the field for the IDOC number.

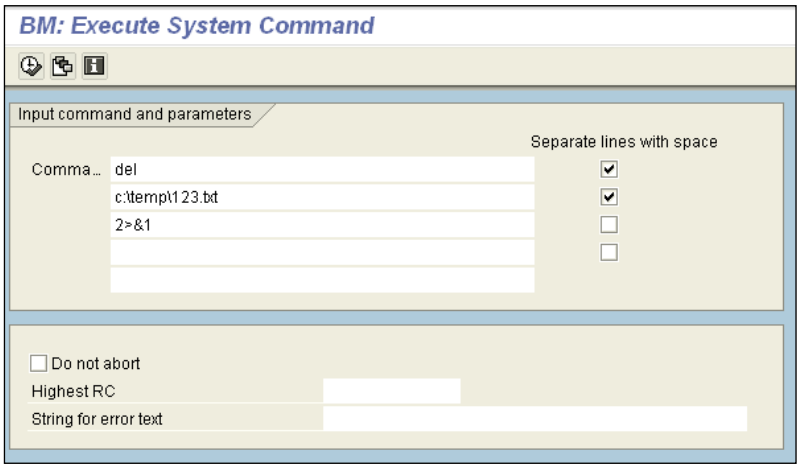
Hence, the program can be used to eliminate such errors.

7.4. Operating System Tools

7.4.1. Start System Command Local

With this program (/BTCMAN/C_CMD) you can execute an operating system command. All parameter lines are summed up to one line before executing the command. If the flag ‚Separate lines with space’ is activated, the entry lines are each linked with a space (blank character) in between.

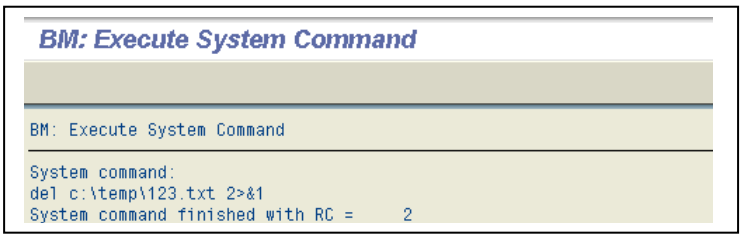
The individual entry lines can contain 45 characters at the most. This corresponds to the maximum length of a TVARV-variable. This makes the differentiation between Step types ‚external program’ and ‚external command’ clear. In this case you can work with TVARV variables, in order to, for example, make path entry according to system (quality (Q) - or productive (P) - System) dynamic.



The 2>&1 reroutes STDERR to STDOUT, in order to analyze errors better.

Note: Generic names (e.g. *.cmd) do not work.

Result: RC = 2 (file not found)



If the execution of a system command does not report back the correct return code to the SAP system, script files must be used. For example, the execution on UNIX systems often produces **1-** as return code.

Otherwise the return code is compared to the value in the field ‚**highest RC'.** If the value is higher or equal to the value in the field, then abort happens.

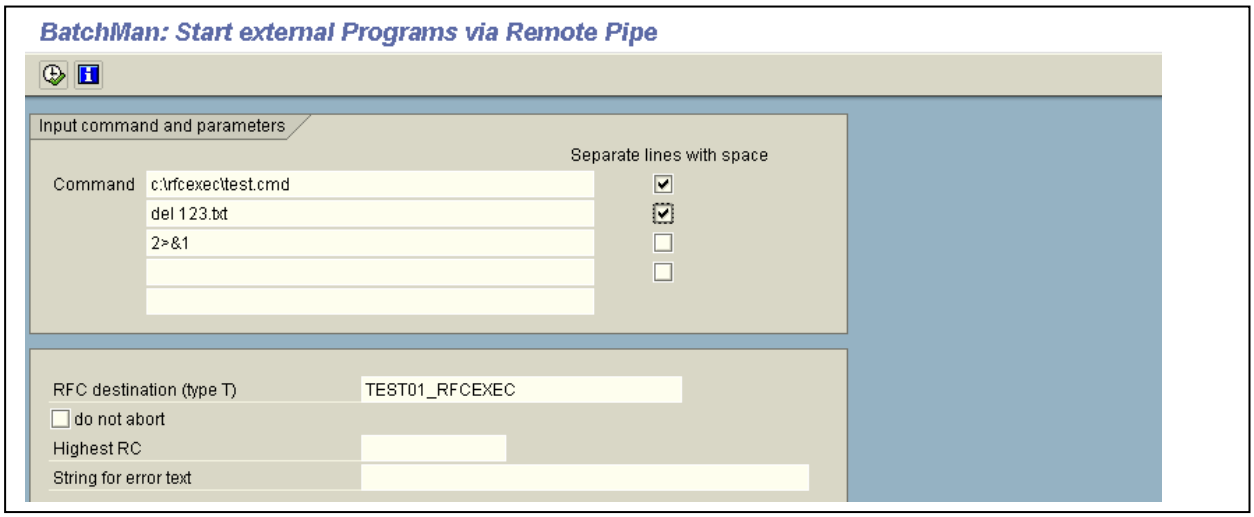
If messages are distributed via the OS program or the script, you can search for these messages via entering the **strings**. If the text is found, then abort happens.

Please note: This program has to be scheduled on the system where the activity should be executed.

7.4.2. Start System Command via RFC

The program /BTCMAN/M_EXT serves to call an external (command or) script via RFC (Remote Function Call) on server that do not have SAP installed.
For this, you have to create an **RFC destination** of the type T with the transaction SM59. The called program is RFCEXEC. The option Gateway always has to be maintained continuously in the corresponding destination.
Please note: In order to place an RFC on a UNIX- or NT-machine, a Gateway must be installed on the instance. Standalone-Gateways can not be found on the SAP-SAPGUI-CD.

The other parameter and their meaning correspond to the ones of the program /BTCMAN/C_CMD in the previous section.



In this case, the return code from the function module is always returned to SAP; so only script calls are useful.

7.4.3. FTP Client/ NT Copy

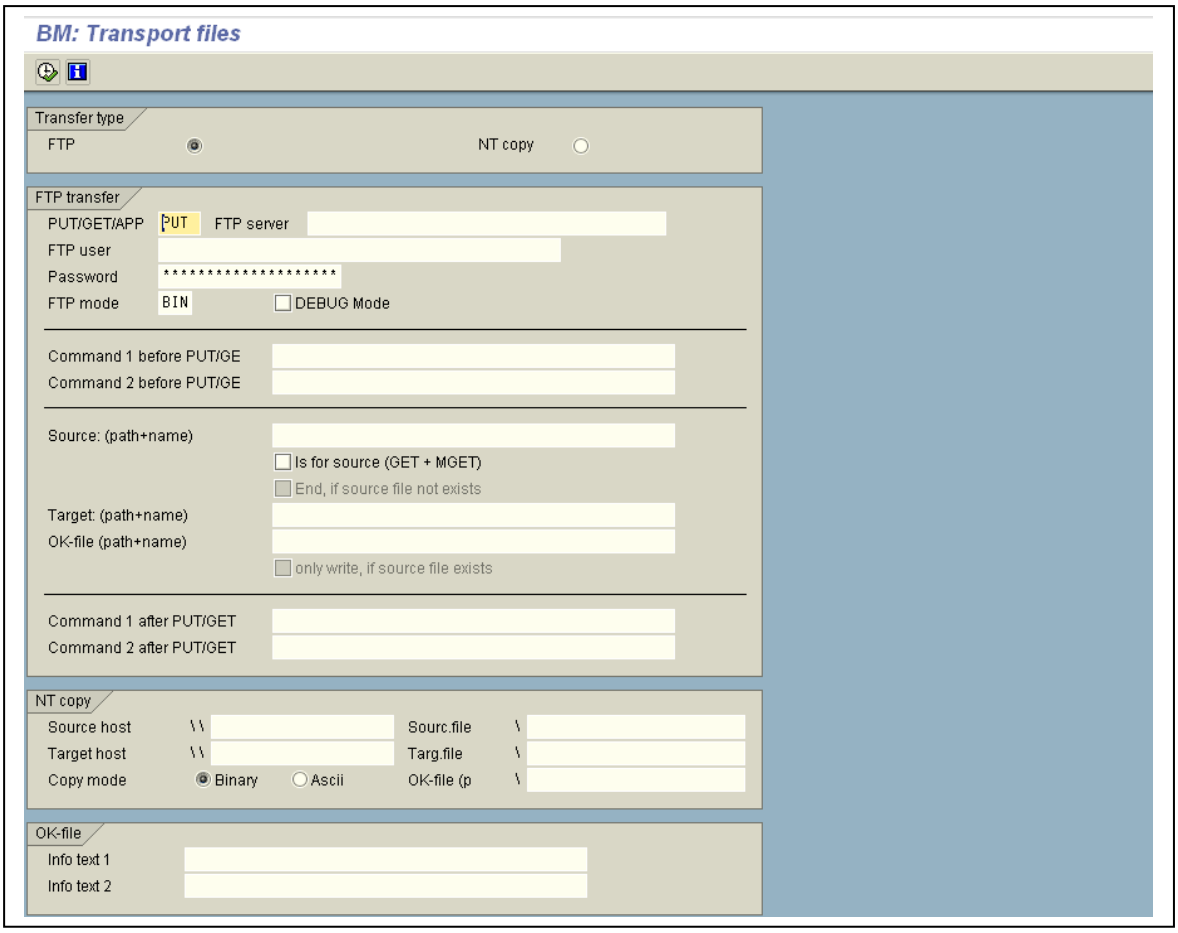
The program (/BTCMAN/C_FTP) is required for the file transfer. This program rules the transfer types FTP and NT- Copy.

In the fields for the file name, you can use logical file names and paths. The values with absolute specifications continue to function.

- <L= logical file> defines a logical file name (transaction FILE)
- <A=logical path> defines a logical path (transaction FILE)

Example: <A=SAVE_PATH>file.txt

For the FTP command, both <L=log.file> as well as <A=log.path> work for source, target and ok file.
For the NT copy: Both for the source host and target host <A=log.path> Also for source file, target file and OK file <A=log.path>
FTP is executed using SAP program SAPFTP. For this, RFC destination SAPFTPA must exist in SM59.
NT copy works only when using UNC names. For NT copy, the source and target systems must be entered with out the leading “\\”. The file ID of the files must be entered without the leading “\” in UNC notation. These characters are introduced automatically upon the execution.



In this report, a covered field exists in which the password is deposited encoded after it has been entered. This way it is not visible anymore in the display of the variants and when the selection cover sheet is created.

Please note: This program has to be scheduled on the system where the activity should be executed.

7.4.4. Move file (from file-periodic WDH)

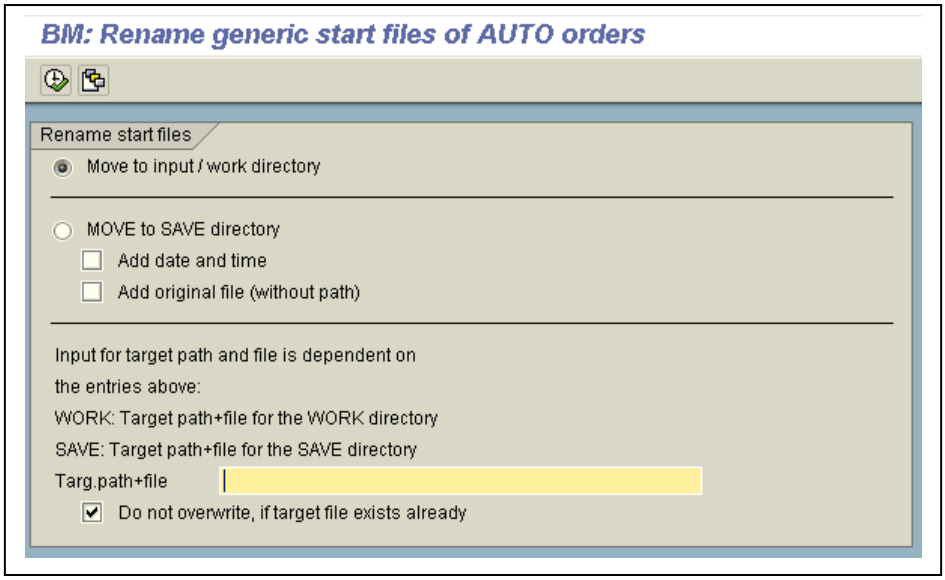
If file-periodic WDH orders are triggered on generic file names, these start files must be able to be MOVED on a fixed name. This program (/BTCMAN/M_REN) specifies the appropriate start file for the AUTO order and moves the file on a fixed name.

Consequently, the report must run before the real functioning file processing.

In the field ‚Target path and file', a logical file name and path can be used as well. The old values with absolute specifications still function.

- <L= logical file> defines a logical file name (Transaction FILE)
- <A=logical path> defines a logical path (Transaction FILE)
  Example: <A=SAVE_PATH>file.txt

After processing, the file can be moved to the archive with an additional MOVE or, alternatively, with the report **/BTCMAN/C_DLB**.

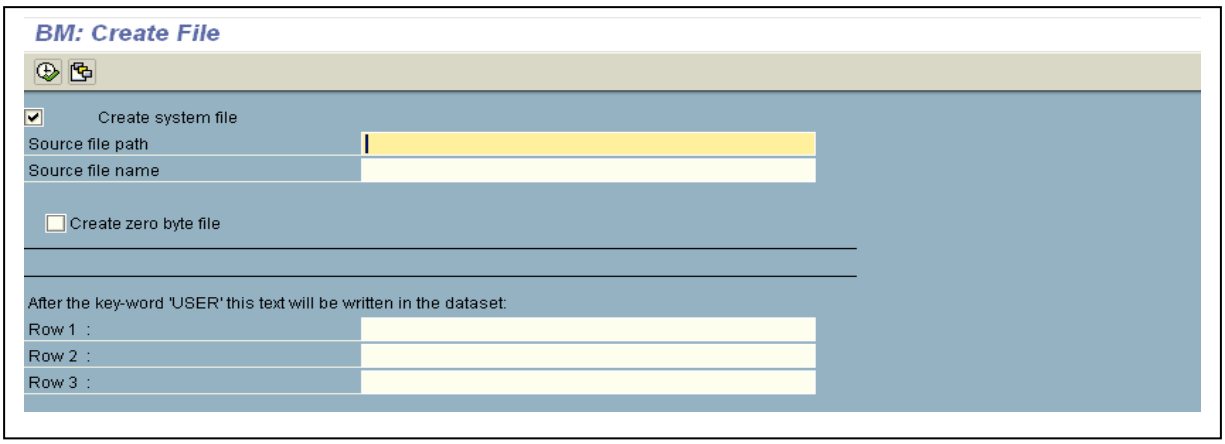


7.4.5. Create File at OS Level

This program (/BTCMAN/C_CRT) creates a file in the file system of the operating system.

In the field ‚Source file path', a logical path can be used as well. The old values with absolute specifications still function.

- <A=logical path > defines a logical path (transaction FILE)
  Example Archiving field: <A=SAVE_PATH>file.txt



Typical areas of application could be the following scenarios:

1. Triggering a file-dependent job that waits for an OK-file, by entering the actual qualified file name of the OK-file.
2. A job that was triggered by a file can generate again its trigger file (OK-file) via linking this report to its step list. This makes triggering of additional jobs - that are dependent on the existence of the same file - possible.
3. A file can be created that generates up to three lines of user-defined text.

Please note: This program has to be scheduled on the system where the activity should be executed.

7.4.6. Delete or archive datasets at OS level

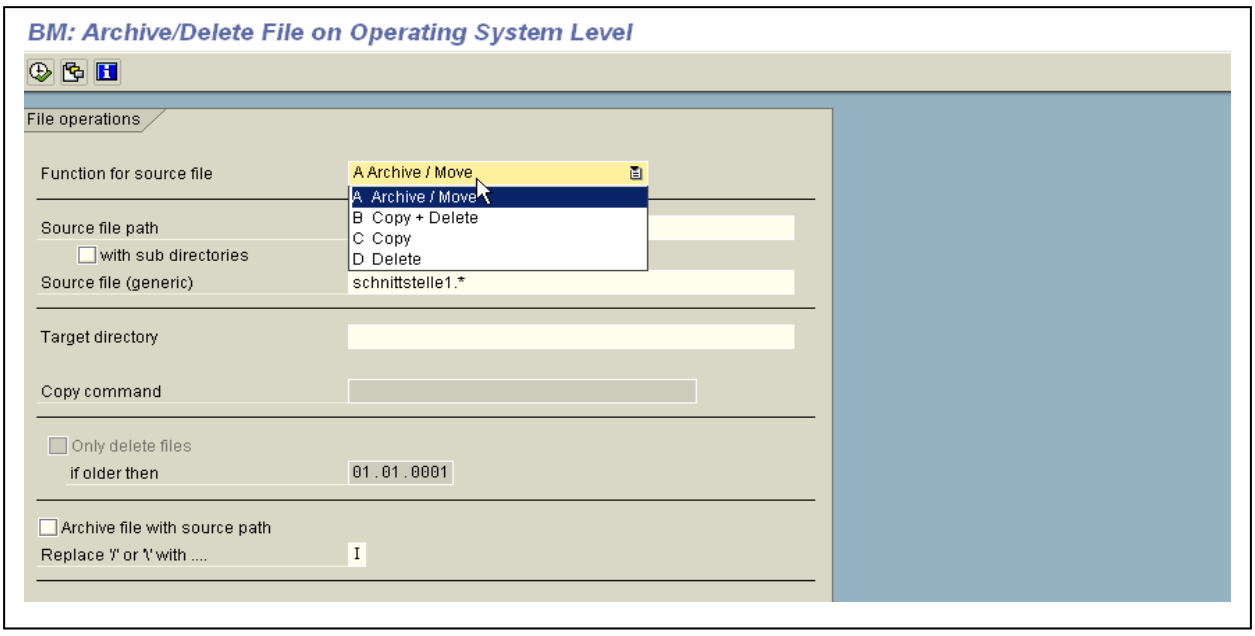
This program (/BTCMAN/C_DLB) archives and / or deletes a file on operating system level. The program can run after a step that processed the original file.

'Function A=Archiving / Move: Program archives the file and deletes the original file.

Function B=Copy and Delete': File will be copied and deleted

Function C=Copy' : File will only copied

'Function D=Deleting': Program deletes the original file. Archiving does not take place. The field ‚if older than' specifies the age of the files that should be deleted.



Replace '/' or '\' with ...': The name of the archive file is formed using the path and file name of the source file, and the time stamp at archiving time. Since the path belongs to the name of the archive file, it contains these directory hyphens (UNIX: '/', WinNT: '\'), which are not valid characters for a file name. Therefore, '/' or '\' must be replaced by a different character, e.g. by 'I' or 'X'.

'Name source file (generic)', 'Search string in source file name': The source file name can be specified generically, e.g. 'FILE*' or 'FILE*.PRD'.

'Archive file with source path': The name of the archive file is formed using the path and file name of the to-be-archived file (source file) and the time stamp (date+time) at the archiving time. In case of marked check box, the path of the source file is used when the archive file name is formed. If the check box is not marked, the archive file name is formed using the name of the source file and the time stamp.

In the fields ‚Source file path' and 'Archiving directory', a logical path can be used as well. The values with absolute specifications still function.

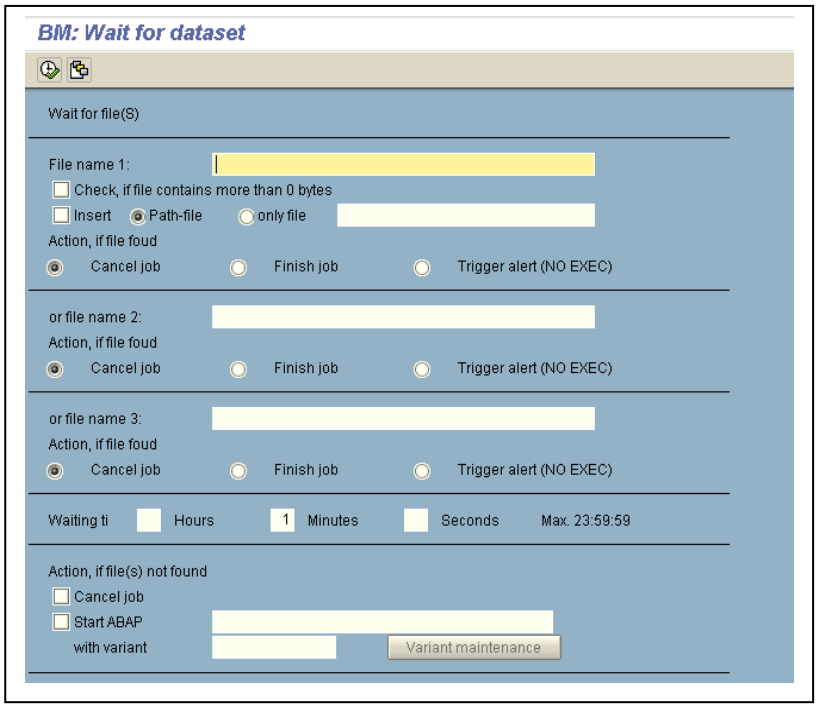
- <A=logical path > defines a logical path (transaction FILE)
  Example archiving field: <A=SAVE_PATH>file.txt

7.4.7. Wait for dataset(s)

With this program (/BTCMAN/C_WFD), it is possible to force up to three files to wait and you can trigger various actions.

In the field for the files, you can use logical file names and paths. The old values with absolute specifications continue to function.

- <L= logical file > defines a logical file name (transaction FILE)
- <A=logical path > defines a logical path (transaction FILE)
  Example: <A=SAVE_PATH>file.txt



This report makes is possible to trigger up to three files simultaneously. An area of application would be, e.g., if processing was started on a different server through the previous job and now a return is expected. This return is a file, which could have the following names depending on the processing success.

Name of the file	Occurred Action
return.ok	Processing was finished without errors.
return.err	An error occurred, but processing can be continued. If necessary, an alarm can be triggered.
return.abort	A serious error occurred. Processing must be terminated.

If the system finds one of the files, it ends the processing and triggers the event selected.

Finish job: The job is finished as usual/normally.

Job abort: Job is aborted with ERROR-Message.

Trigger alert (NO EXEC): Alarming is triggered that was entered under NO EXEC and the job is finished as usual.

Wait time: Maximum runtime of the report.

Action, if file(s) not found: If none of the files are found during the waiting period, you can abort. And /or a report is started with the variant (e.g. /BTCMAN/M_SKP for skipping the successor jobs).

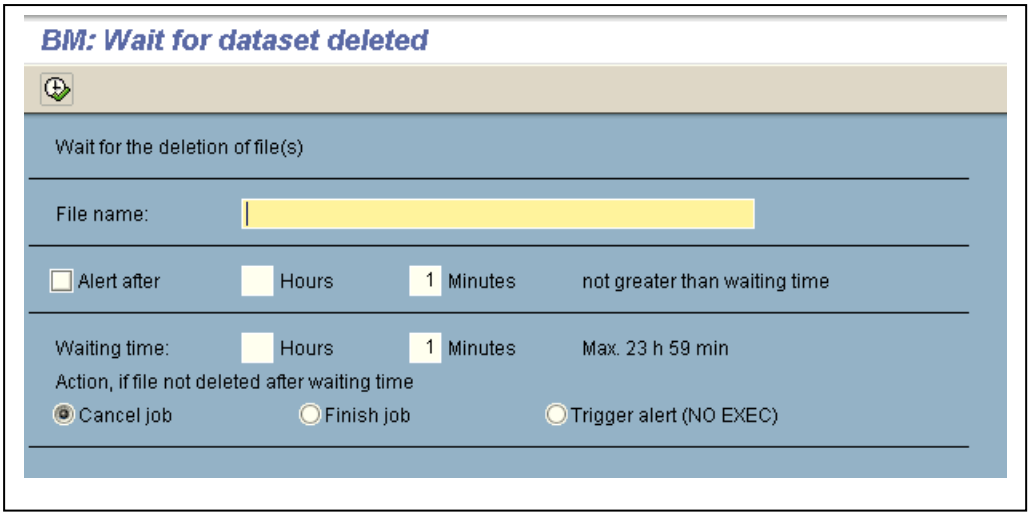
You can also wait for only **one** file. For this file, you can check if it contains data (**Check, if file contains more than zero bytes**). The appropriate flags and settings are valid here as well.

7.4.8. Wait for deleting dataset

This program (/BTCMAN/C_WF2) awaits a specified time that a created file has disappeared, that is deleted.

In the field for the file, you can work with logical file names and paths. The values with absolute specifications still function.

- <L= logical file > defines a logical file name (transaction FILE)
- <A=logical path > defines a logical path (transaction FILE)
  Example: <A=SAVE_PATH>file.txt



The **Waiting time** is the maximum run time of the program. If the file is still found after the waiting time, you can abort; or the entered alert entered in the job under NO EXEC is triggered the job finished as usual/normally. Moreover, an interim alert can be triggered.

7.5. Calendar Tools

A query of the change authorization for calendar maintenance SCAL is done for the following functions.

7.5.1. SAP Factory Calendar (Transaction SCAL)

Call the SAP standard transaction SCAL. Creating and changing of SAP public holidays, public holiday calendars and factory calendars.

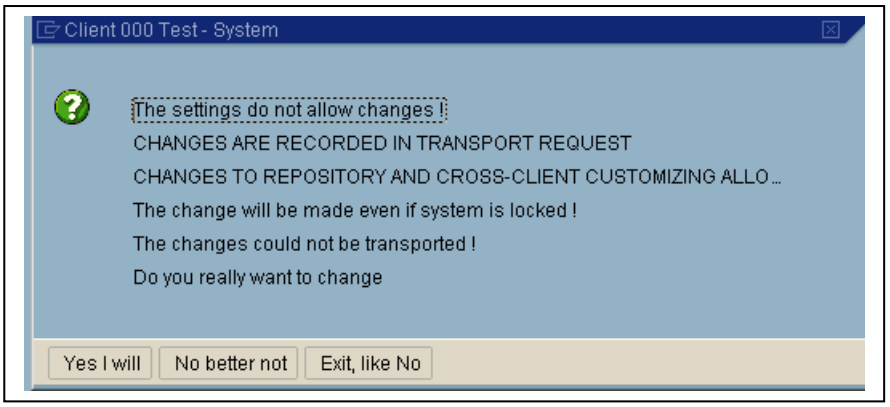
7.5.2. Maintain Factory Calendars (Graphic mode)

Simplified factory calendar maintenance. It is complicated to maintain special periods in the transaction SCAL. With this report, switching one day between 'workday' and 'Non-workday' is fairly simple.

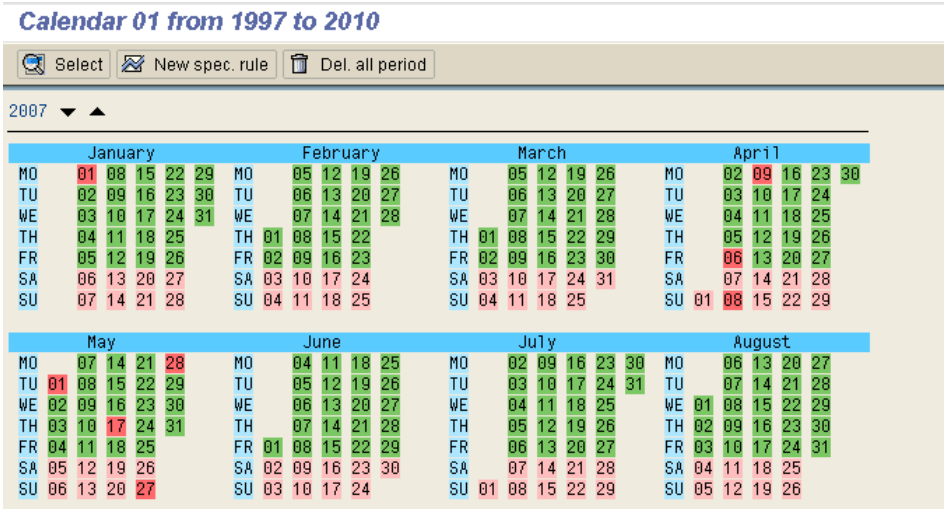
Factory calendars, however, must have been created in SAP beforehand. Public holiday calendars must be included in the factory calendars.

After entering the calendar-Ident, warning messages - depending on the system settings- appear which provide the following information:

- Changes become effective in spite of system lock (productive client), because information is entered directly into the SAP-tables
- The changes are, as customary in SAP calendars, not recorded in transports.

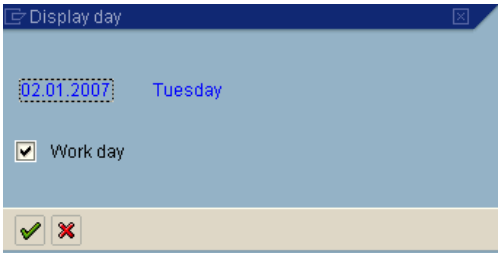


You can finish this function with NO and ABORT; if YES, the calendar with the current year is displayed.

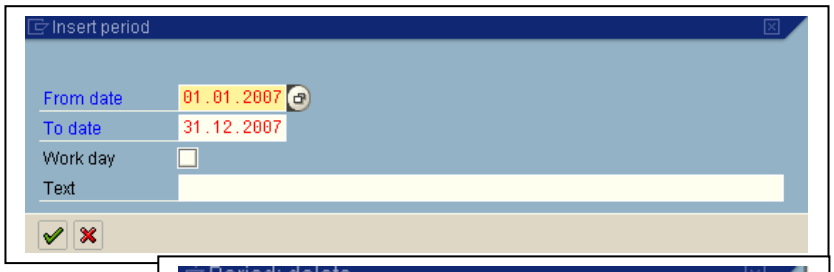


The arrows next to the displayed year allow you to switch to the previous or the following year.

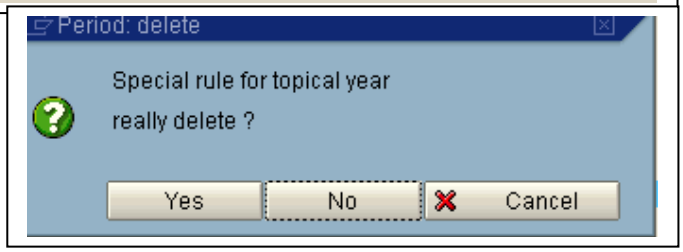
Select a day and selection via puck, or simply a double click on a day, will display the day. Via the flag '**workday'** you can switch easily and quickly between workday and non-workday. Special periods for this day will then be created or deleted automatically.



Via ‚New special rules' you can create special rules/ periods of more than one day in the displayed (current) year. Restriction: Periods that overlap the year must be maintained via SAP.



Likewise you can delete special rules/period via ‚Delete all periods' in the displayed year.



7.5.3. Calculate dependent calendar

The report serves to compute an SAP calendar from a fiscal year version which was defined in 'customizing' of the financial system. There you have to specify up to which calendar date a period is valid per posting period.

According to the end date of a period, the appropriate working day is set in the SAP calendar. These working days are saved in the SAP calendar as special periods.

The fiscal year version, for example, is created according to the 4-4-5 rule. In the calendar, the last day of the period is saved as working day.

During the evaluation process, all old special periods are deleted in the dependent calendars. After that, a new calculation is done for the special periods beginning at the ‚from year', and ending at the ‚up to year'.

With the report that follows in the function tree, additional calendars can be maintained from this calendar.

Prerequisite: The used fiscal year version must be maintained including the periods.

7.5.4. Calculate Factory Calendar from Fiscal Year Variant

This report serves to compute depending SAP-calendars.

The calendar that contains the defined Ultimo –working days delivers the corresponding working days for other calendars.

For example: The Ultimo calendar is created according to the 4-4-5 rule. Based on this calendar, the Ultimo plus or minus calendars can be calculated. At the most, 14 calendars are possible for plus, and 14 calendars are possible for minus.

During the computing process, all old special periods are deleted in the depending calendars. After that, a new calculation is done for the special periods beginning at the ‚from year' and ending at the ‚up to year'.

Prerequisite: The calendar for Ultimo must have been created and maintained. The to-be-computed calendars must also have been created.

All calendars must use the same holiday calendar.

8. TVARV Variables

In SAP variants it is possible to set certain field contents variables (Selected variables) as default. SAP knows three types:

- Table variables from TVARV
- Dynamic date calculation
- User-defined variables

Unfortunately, SAP only provides rudimentary date calculation/computation. It is not possible to automatically apply complicated rules on variables. These variables are maintained in table TVARV (up to SAP 4.6D) and are allocated to certain fields in the variants.

Please note: The table TVARVC (starting with SAP 6.20) is client-dependent. Within this section only the TVARV table is meant. All executions are valid also for the table TVARVC, deviations are pointed out.

In **BatchMan** it is possible to execute more complex date calculation/computation through TVARV. The procedure is done as follows:

- Define TVARV-variables
- Define rules for variables in **BatchMan**
- Define and set up variable updates
- Create variants and assign variables

TVARV variables are also variables defined in the table TVARV, which have assigned values in this table. By assigning these variables to variant field, the value of the variables is transferred to the appropriate field of the variant.

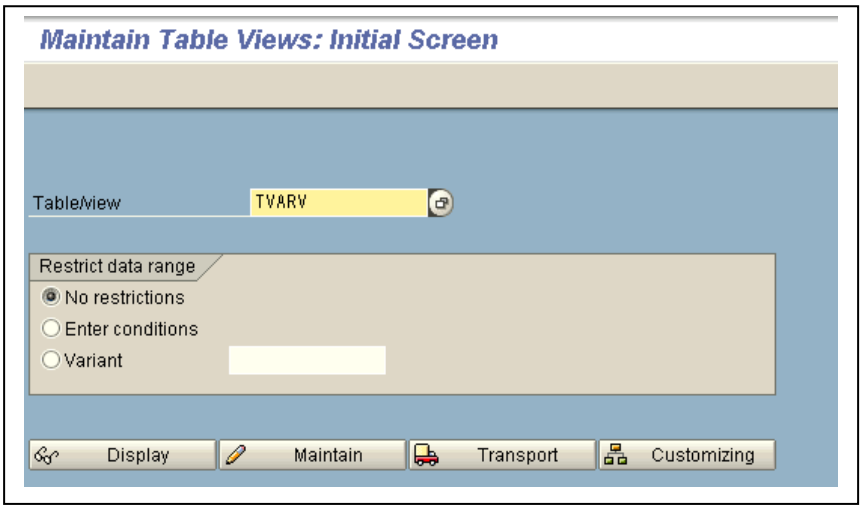
Analog to the selection options, table TVARV knows two types of variables:

Parameter	one value per field
Select-Options	several values per field with options

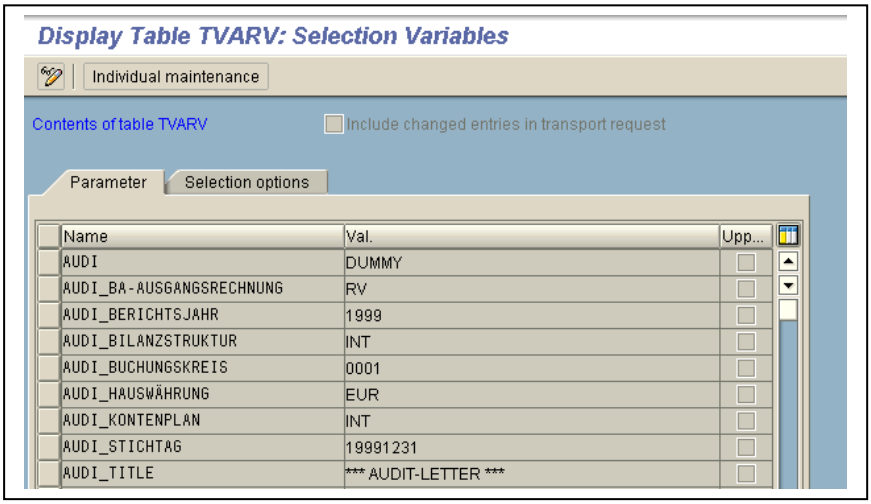
8.1. Define TVARV

The TVARV variables must be maintained in the system in which they are used. For a master system, the call is done via *Function tree BatchMan→TVARV© maintenance → Table maintenance → Maintain Table TVARV(C)* or maintain in the client systems with TA Code SM31 maintain table TVARV.

Please note: SAP Mistakes occurred during the transformation of table maintenance for the TVARVC table starting with version 6.20. There possible still TVARV has to be entered as table. The maintenance, however, goes to the client-dependent TVARVC. Help can be found in the OSS, which has to be considered/ observed on a current basis. Depending on the HotPackage status, this is also handled differently.



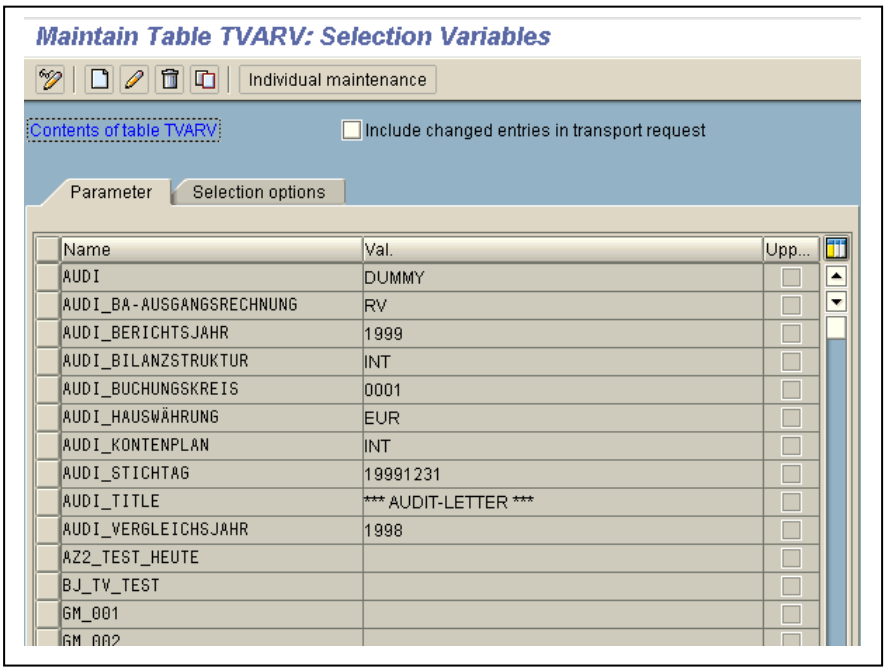
After pushing the button ‚Maintenance' and confirming the prompt, the table maintenance appears.



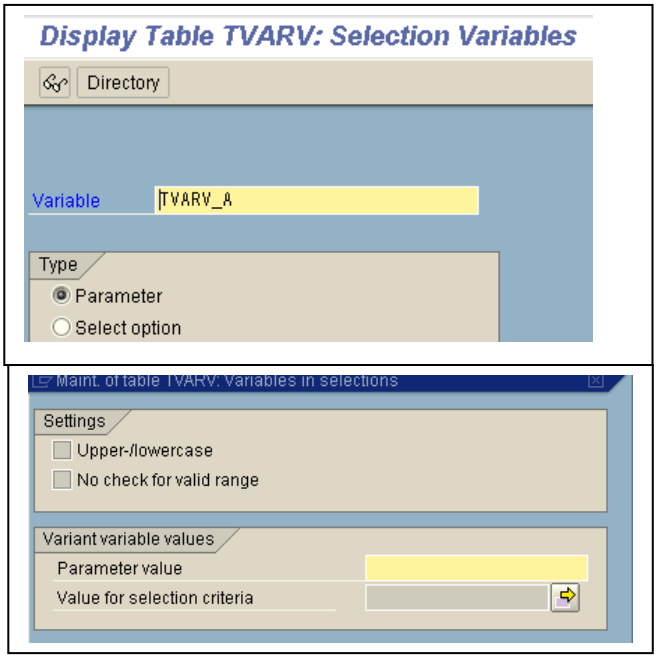
With each installation, SAP delivers a few variables, which are also filled with values.

Maintenance can be done in two different ways: Individual maintenance or maintenance in the table. As standard, the table maintenance view appears. Via the display<-> change button, maintenance is switched on.

Via the appropriate buttons, variables can be created, changed, deleted and copied. Via the file cards you can switch between variables for parameter and/or selection options.



With the button ‚Individual Maintenance', the variables can be treated individually.

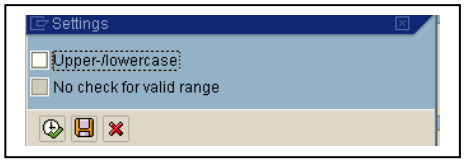


If a variable is created here, then an additional PopUp-window requests information about certain parameter and the contents/ value for the variable.

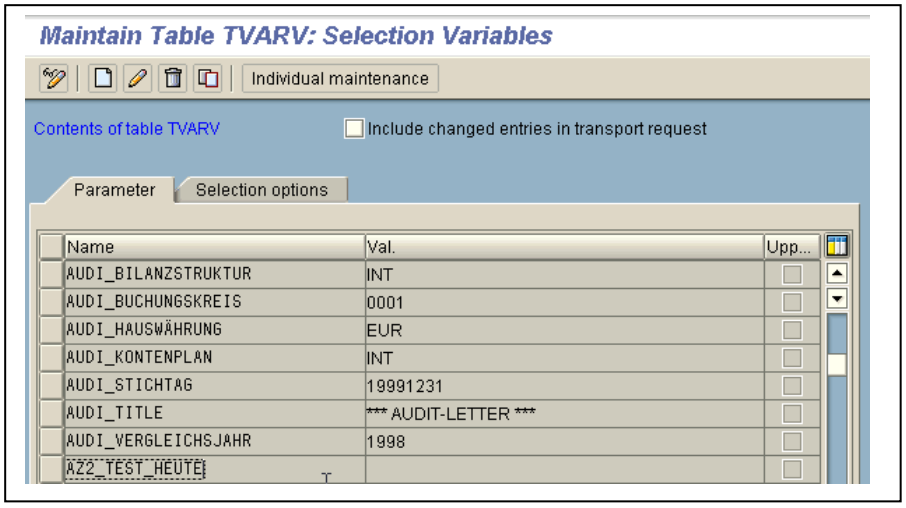
If the contents remain empty, then an additional request appears when the saving action is done. Then the button, ‚Safe empty' is important.

The settings are called via the button ‚settings'.

The interval check can only be activated in variables of the type ‚Select Options'.



Via the button ‚catalogue' you get back to the table maintenance view. The ‚settings' for the variable types can be maintained here as well.



8.2. Maintaining Rules for TVARV

In the TVARV rules, each created TVARV variable gets an assigned rule (regularity). If the TVARV-variable is the type parameter, only the low-value is filled. In TVARV-variables of the type select option, the low-value and the high-value must be filled. Via the appropriate button, you can switch between low and high.

The rules are maintained in the master system.

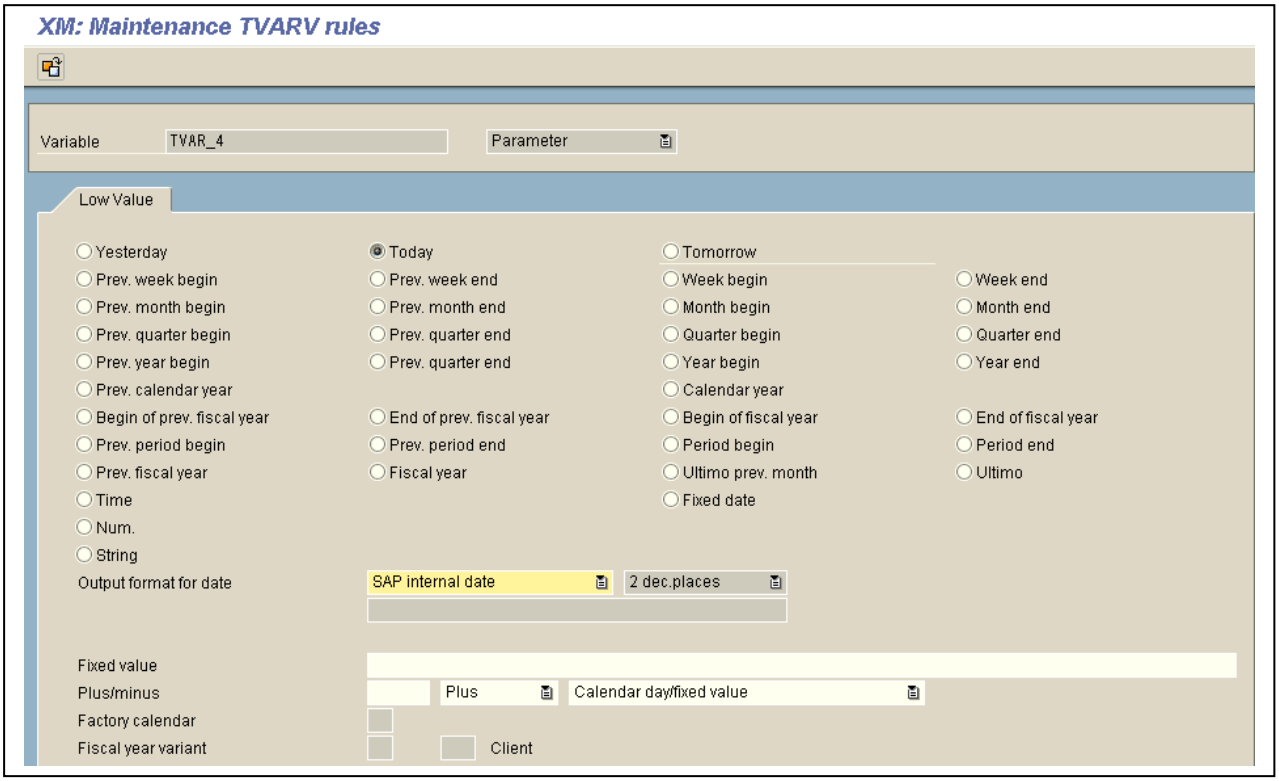
Function tree BatchMan→TVARV© Maintenance → Table maintenance → Maintain TVARV(C) rules

A number of date functions are available. These are client-independent. Next to it, you can also execute calculations based on business/fiscal year variants. These functions are **client-dependent** and can lead to terminations/aborts when updating on the wrong client!

Several rules are possible for TVARV variables that are a select option. A ‚sequential number' was used for this purpose.

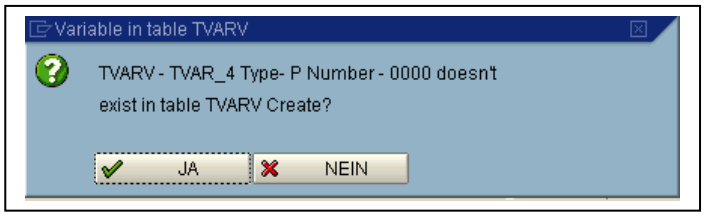
Next to date variables you can also specify variables with system parameters. The TVARV variables are assigned a fixed value like e.g.: C:\temp\dat_<DATE><OPSYS><SAPRL. The substituted values can, for example, change to C:\temp\dat_19980721HP- UX40B.

Screen for the rule maintenance:



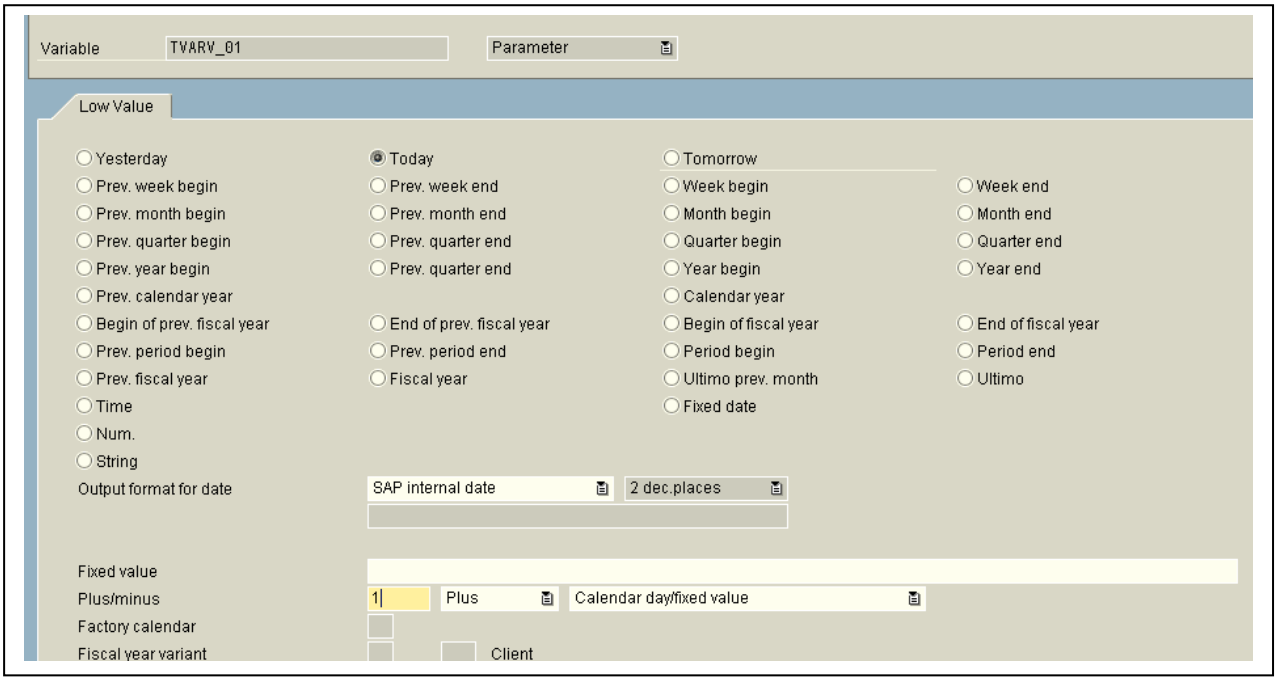
During creation of a rule, the notification appears in the status line ‚TVARV - TVARV_01 Type - P in the table /BTCMAN/M_TVREG not existing. Please maintain and save'. This only indicates that the variable did hot have a rule yet which now has to be created.

If the variable does not exist in the TVARV table, then a request appears at the initial screen/entry if you want to create the variable in the TVARV table.



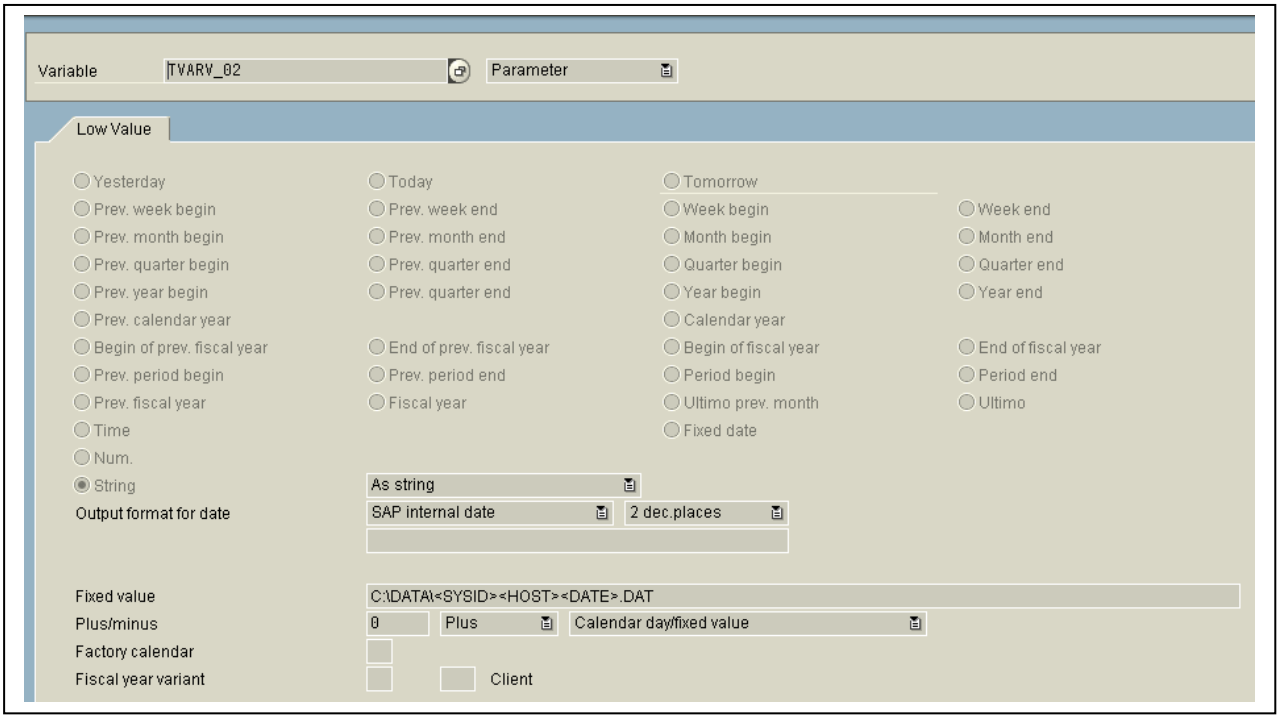
The manual maintenance of the TVARV table is then not necessary anymore. (chapter 8.1)

Some examples:



The variable TVARV_01 is a PARAMETER with the following rule:

- Refers to the current date (TODAY)
- A workday is added (PLUS 1 workday active)
- Output as three-digit month



The variable TVARV_02 is a PARAMETER with the following rule:

- A fixed part ,C:\DATA\; afterwards the variable file name with
- SAP System Ident <SYSID>
- Host name <HOST>
- Date <DATE>

The variable TVARV_03 is a select option (LOW and HIGH) with the following rule:

- LOW Pre/Before-quarter begin with output date in free format ‚MM.YYYY'
- HIGH Pre/Before-quarter end with output date in free format ‚MM.YYYY'

In this case, the date should be produced in a specified form. Standard format (SAP internal date format) for the output of the date fields is always YYYYMMDD. In free format, the following letters can be used:

Y for the year **M** for month and period **D** for the day

As shown in the examples, a series of reserved words exists for the usage of fixed values:

Reserved Word	Replacement Value
<OPSYS>	Operating system
<INSTANCE>	Instance of SAP application
<SYSID	Name of the R/3 application according to system field SY-SYSID
<DBSYS>	Data bank system according to system field SY-DBSYS
<SAPRL>	R/3 Release according to system field SY-SAPRL
<HOST>	Host name according to system field SY-HOST
<CLIENT>	Client according to system field SY-MANDT
<LANGUAGE>	Logon language according to system field SY-LANGU
<DATE>	Date according to system field SY-DATUM
<YEAR>	Year according to system field SY- DATUM, four-digit
<SYEAR>	Year according to system field SY- DATUM, two-digit
<MONTH>	Month according to system field SY- DATUM
<DAY>	Day according to system field SY- DATUM
<WEEKDAY>	Weekday according to system field SY-FDAYW
<TIME>	Time according to system field SY-UZEIT
<STIME>	Hour and minute according to system field SY- UZEIT
<HOUR>	Hour according to system field SY- UZEIT
<MINUTE>	Minute according to system field SY- UZEIT
<SECOND>	Second according to system field SY- UZEIT
<P=name>	Value of a profile parameter of the running system
<V=name>	Value of a variable according to variable table (maintenance via TCODE FILE)
<F=name>	Return value of a function module
<T=name>	TVARV variable
<L= log.file>	define a logical file name (transaction FILE)
<A=log. path>	define a logical path (transaction FILE)

With the integration of such parameter into physical file or path names, a differentiated and at the same time standardized allocation of file names is supported. For example, the parameter **<TIME>** can be useful if one file of a job is saved several times within a short time span. Besides the system field values, the name allocation can be handled flexibly, especially via usage of the last specified parameter:

With **<P=name>,** the values of the profile parameters of the running system can be included. You receive a list of the profile parameter and their value if you execute the report RSPARAM.

With **<V=name>,** the values of variables can be entered, which were defined with the transaction FILE in accordance with the maintenance of the control tables.

With **<F=name>,** return values of function modules can be included. The name of here used function module must begin with the prefix 'FILENAME_EXIT_'. Please note that the function module in the reserved word is only addressed with the part of its name which follows this prefix. For example, if you use the function module FILENAME_EXIT_EXAMPLE, the placeholder would be <F=EXAMPLE>. The function module must have the export parameter OUTPUT, for which a reference type may not be indicated. Import parameter must have default values. Table parameter is not supported.

With **<T=name>,** values of other TVARV variables can be inserted.

8.3. Updating the Variables

There are two options to update these variables.
1.) A number of variables is updated manually
2.) A number of variables are updated via a **BatchMan** report. Different reports are available within **BatchMan**.

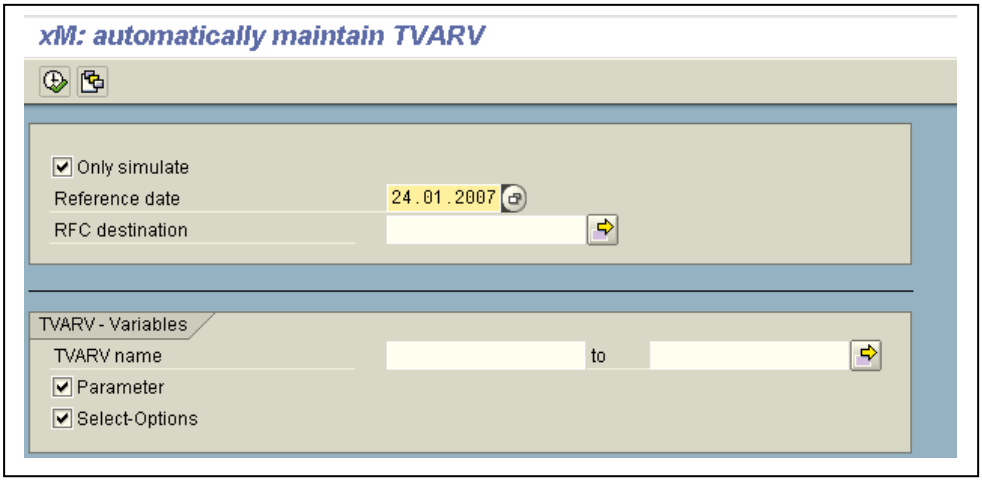
8.3.1. Updating TVARV(C) Variables

Report /BTCMAN/M_RV1 will be update the values of the TVARV variables. You need to determine which variables must be updated.

It could be the case that you have to avoid to update all variable at the start-of-day. Then you only have to enter the variables, which should be updated, in the variant.

Since the rules are created in the master system, the variables have to be updated in the master system and the client systems.

*Function tree **BatchMan**→TVARV© Maintenance → Reports→ Updating TVARV(C) Variables*



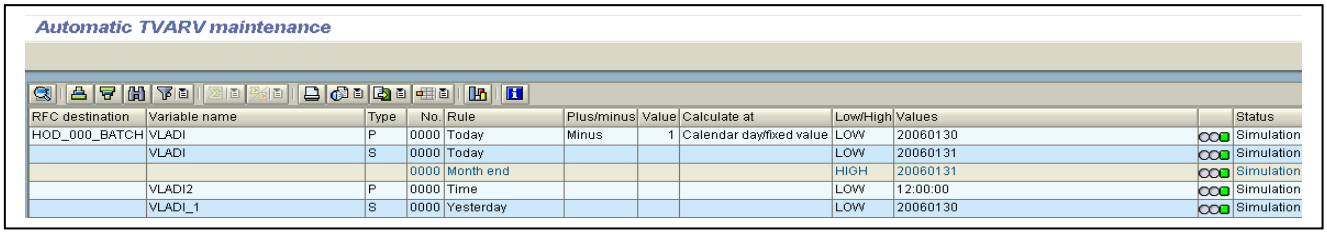
Corresponding to the **reference date,** the selected variables are set to the new value, if the field **only simulate** is not active.

The **RFC Destination** (NONE = local) determines on which system the TVARV variables are updated.

Note:

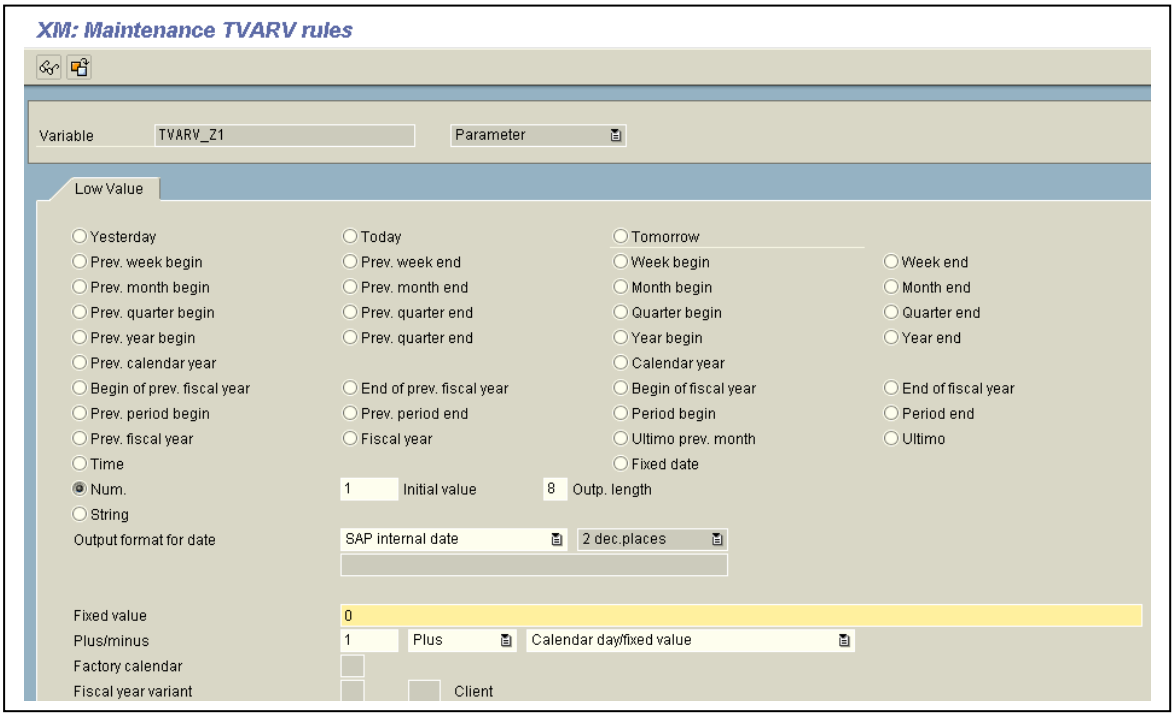
- Missing TVARV variables are created
- You can only maintain variables that have a defined rule.
- Only single values should be selected, so that these - after you create a new TVARV are not supplied with false values via intervals by mistake.
- Please choose a date variable for the reference date ("current date") to ensure the correct sequence in the internal calculation process.

This way vvariables in the example are like this:



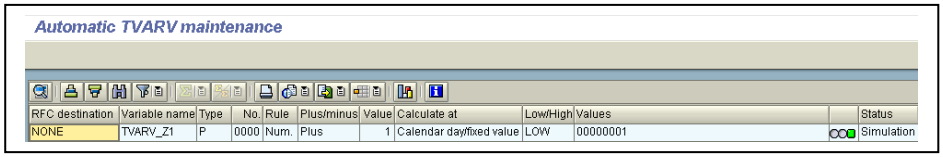
8.3.2. Initializing TVARV(C) Variables – Type Number

This report (/BTCMAN/M_RV3) sets variables that are defined as number back (reset) to their initial value.

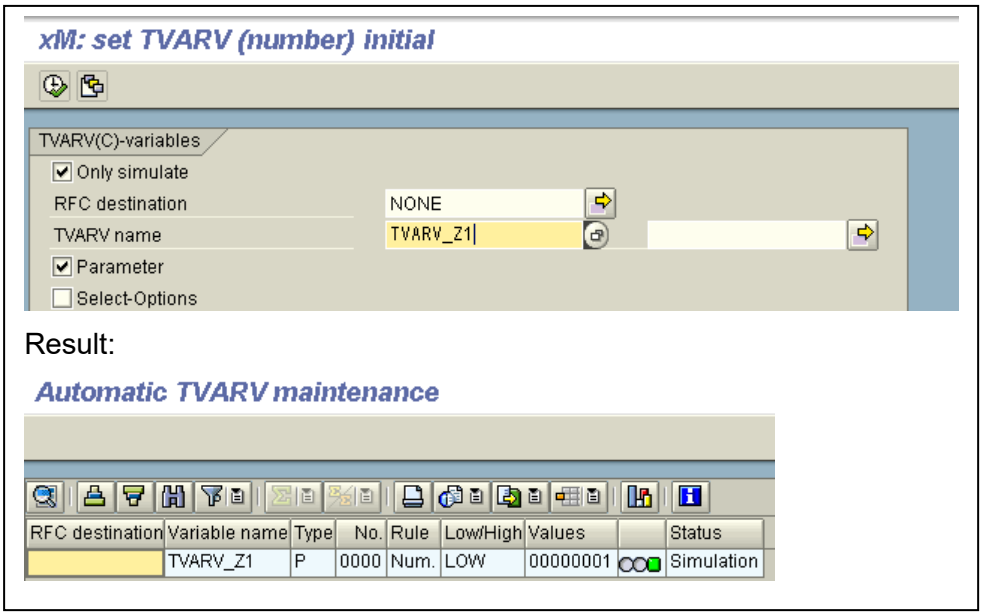


For this you indicate the variable as number and specify the **initial value** and the **output length**. The **value in Plus/Minus** determines the value with which the variable is added (if **PLUS**) or subtracted (if **MINUS**) as **fixed value**.

The variables are updated with the previously described report /BTCMAN/M_RV1.



Reset the variable to initial value ‚00000001'.



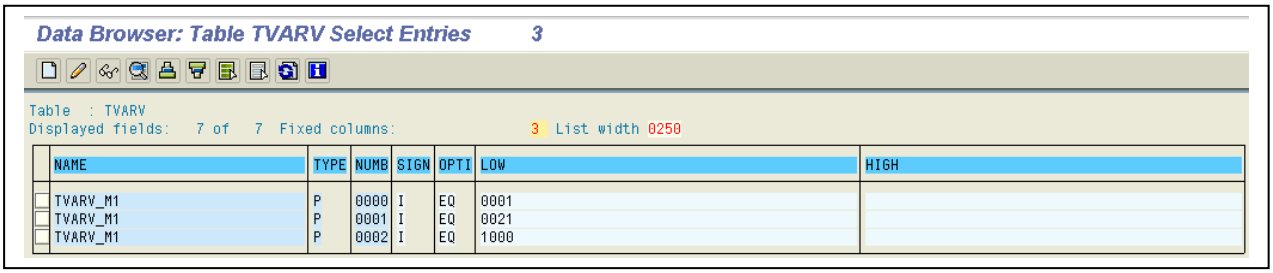
8.3.3. TVARV(C) Single Value from TVARV(C) Multi Value

This report (/BTCMAN/M_RV4) updates the single/individual value of another variable which contains several values (Multi-value variable).

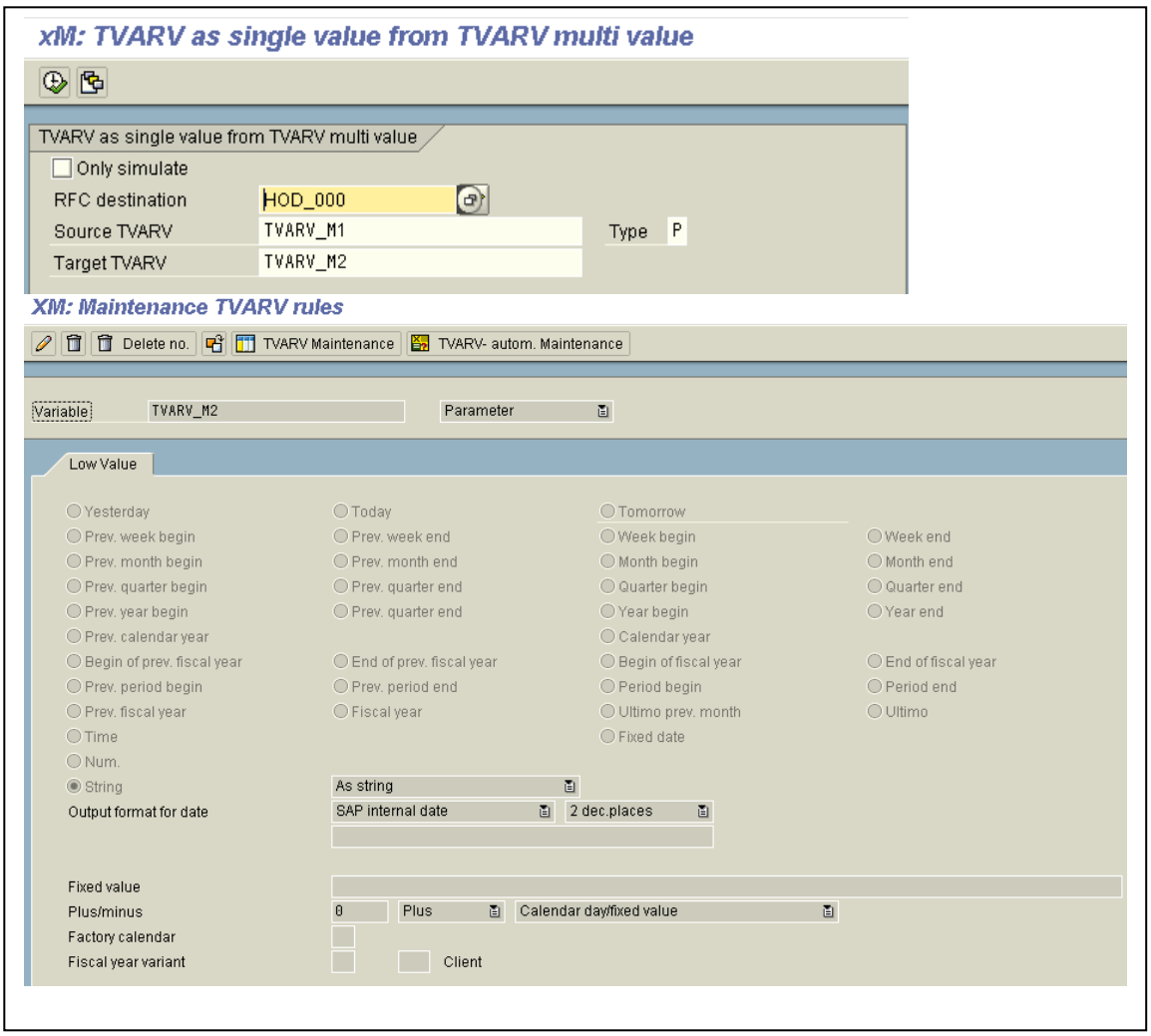
For this purpose, you have to maintain the multi-value variable accordingly in the TVARV table. In the example, a variable TVARV_M2 should be updated from the variable TVARV_M1. The variable TVARV_M1 contains the postings.

This variable TVARV_M1 is created with transaction SM31. However, since only the sequential number ‚0000' (field NUMB) is maintainable there, additional values are maintained via the transaction SE16 in the table TVARV; and consequently, NUMB has to be increased one number.

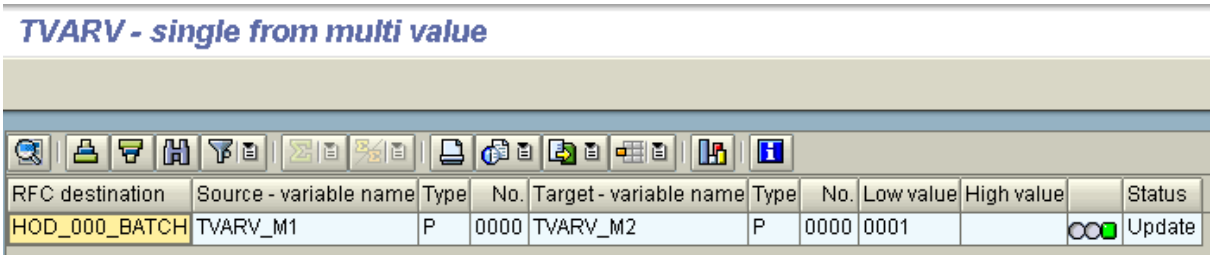
Create variable TVARV_M2 as rule:



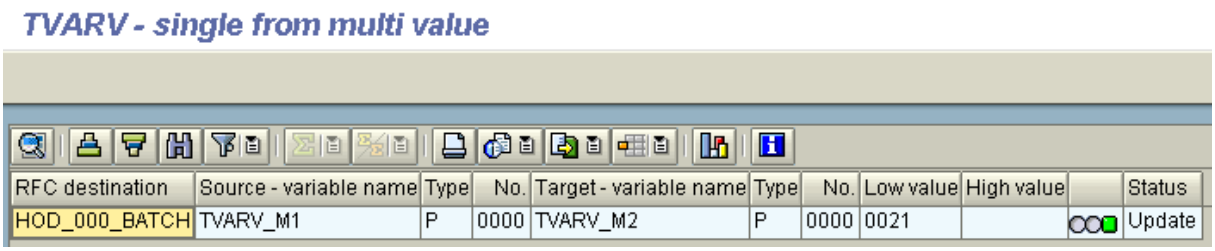
Executing the report will update the variable TVARV_M2. If the variable TVARV_M2 is still empty, then the first value is taken. Afterwards always the next value from the variable TVARV_M1.


xM: TVARV as single value from TVARV multi value
TVARV as single value from TVARV multi value
Only simulate
RFC destination
HOD_000
Source TVARV
TVARV_M1
Type
P
Target TVARV
TVARV_M2
XM: Maintenance TVARV rules
Delete no.
TVARV Maintenance
TVARV- autom. Maintenance
Variable
TVARV_M2
Parameter
Low Value
Yesterday
Today
Tomorrow
Prev. week begin
Prev. week end
Week begin
Week end
Prev. month begin
Prev. month end
Month begin
Month end
Prev. quarter begin
Prev. quarter end
Quarter begin
Quarter end
Prev. year begin
Prev. quarter end
Year begin
Year end
Prev. calendar year
Calendar year
Begin of prev. fiscal year
End of prev. fiscal year
Begin of fiscal year
End of fiscal year
Prev. period begin
Prev. period end
Period begin
Period end
Prev. fiscal year
Fiscal year
Ultimo prev. month
Ultimo
Time
Fixed date
Num.
String
As string
Output format for date
SAP internal date
2 dec.places
Fixed value
Plus/minus
0
Plus
Calendar day/fixed value
Factory calendar
Fiscal year variant
Client

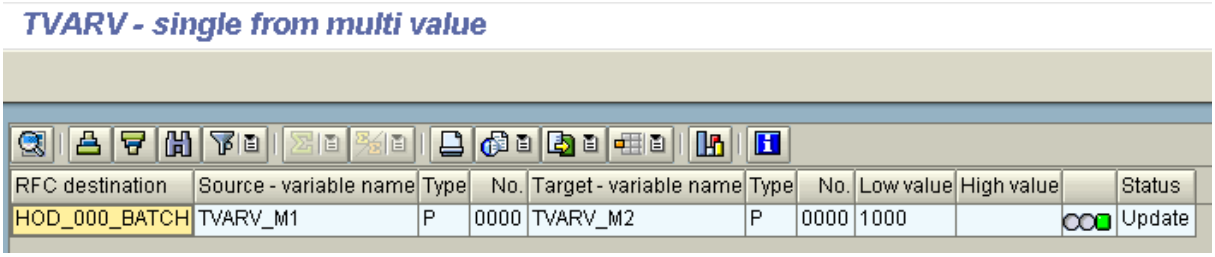
First run:



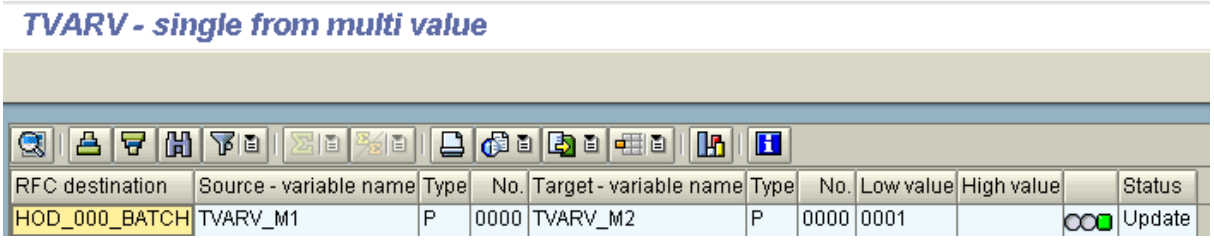
Second run:



Third run:



Fourth run:



8.3.4. TVARV(C) Variable from System Fields

This report (/BTCMAN/M_RV6) offers the option –depending on the fields of the system table – to enter values (LOW) in a TVARV variable. This is useful for values that are dependent on the environment.

Example

For a 3 system environment with development-, quality- and productive system, different serves must be used depending on the SYSID:

System field:	SYSID
TVARV :	Name of the TVARV variable
Condition 1:	E01
Value for TVARV:	SERVER_E
Condition 2:	Q01
Value for TVARV:	SERVER_Q
Condition 3:	P01
Value for TVARV:	SERVER_P
Condition 4:	T01
Value for TVARV:	SERVER_Q
Condition 5:	
Value for TVARV:	

If none of the conditions are met, then the value of the TVARV will not be updated. That means that the old value continues to exist/remains.