

File & DataManager



File & DataManager Installation- and Usermanual

Produktinformation and Support

Phone: + 49 (0) 40/32 80 86 - 0
E-Mail: batchman@honico.com
Web: www.honico.com
Status: March 2024

SAP® Certified
Powered by SAP NetWeaver®



Table of Contents

1.	Introduction	3
2.	Installation and setup	5
2.1.	General information	5
2.2.	Agent installation in Microsoft Windows	5
2.3.	Agent installation in Unix/Linux	8
2.4.	BM setup	10
2.5.	Setting up systems with BatchMan Agent	11
2.6.	Installation https	13
2.6.1.	Use of self-issued certificates	14
2.6.2.	Use of CA certificates	17
2.6.3.	Switch agent to https	17
2.6.4.	Switch FDM to https	19
2.7.	Setting up load distribution and failsafe	22
2.8.	Set up: System landscape	24
2.8.1.	FTP connections	24
2.8.2.	SFTP connections	25
2.8.3.	SAP connections	27
2.8.4.	SMB connections (Windows file release and SAMBA server)	28
2.8.5.	BatchMan agent connection via BatchMan agent	29
2.8.5.	SMTP Connections	30
2.8.5.	HTTP Connections	31
3.	File interface	31
3.1.	General data	32
3.2.	File processing	33
3.2.1.	Storage of variables in a separate overview tab	35
3.2.1.	Detail view/maintain file processing	36
3.2.2.	Dynamic file names	38
3.2.3.	Copy	43
3.2.4.	Rename	44
3.2.5.	Move	44
3.2.6.	Delete	44
3.2.7.	Create directory	45
3.2.8.	Create File	45
3.2.9.	Zip	45
3.2.10.	Unzip	46
3.2.10.	Split	47
3.2.12.	Merge	47
3.2.13.	Convert	48
3.2.14.	Execute Command	49
3.2.15.	Encrypt and decrypt files with PGP	50
3.2.16.	Synchronization	54
3.2.17.	Print	55
3.2.18.	Mail	55
3.2.19.	REST	56
3.2.20.	Additional file operations	62
3.3.	Scheduling dates	62
3.4.	Start criteria	66
3.5.	Alerts	71
4.	Monitoring	73

1. Introduction

Having to intervene manually in daily file management is a constant problem in most companies. Uniform processes, content checks and/or monitoring are missing; often several people are handling errors with great effort. Our objectives are to get the greatest level of automation for file handling, to eliminate scripting, to implement comprehensive error handling and monitoring. Thus, staff is relieved, system resources are saved, and compliance conformity is ensured.

The solution is FDM by BatchMan

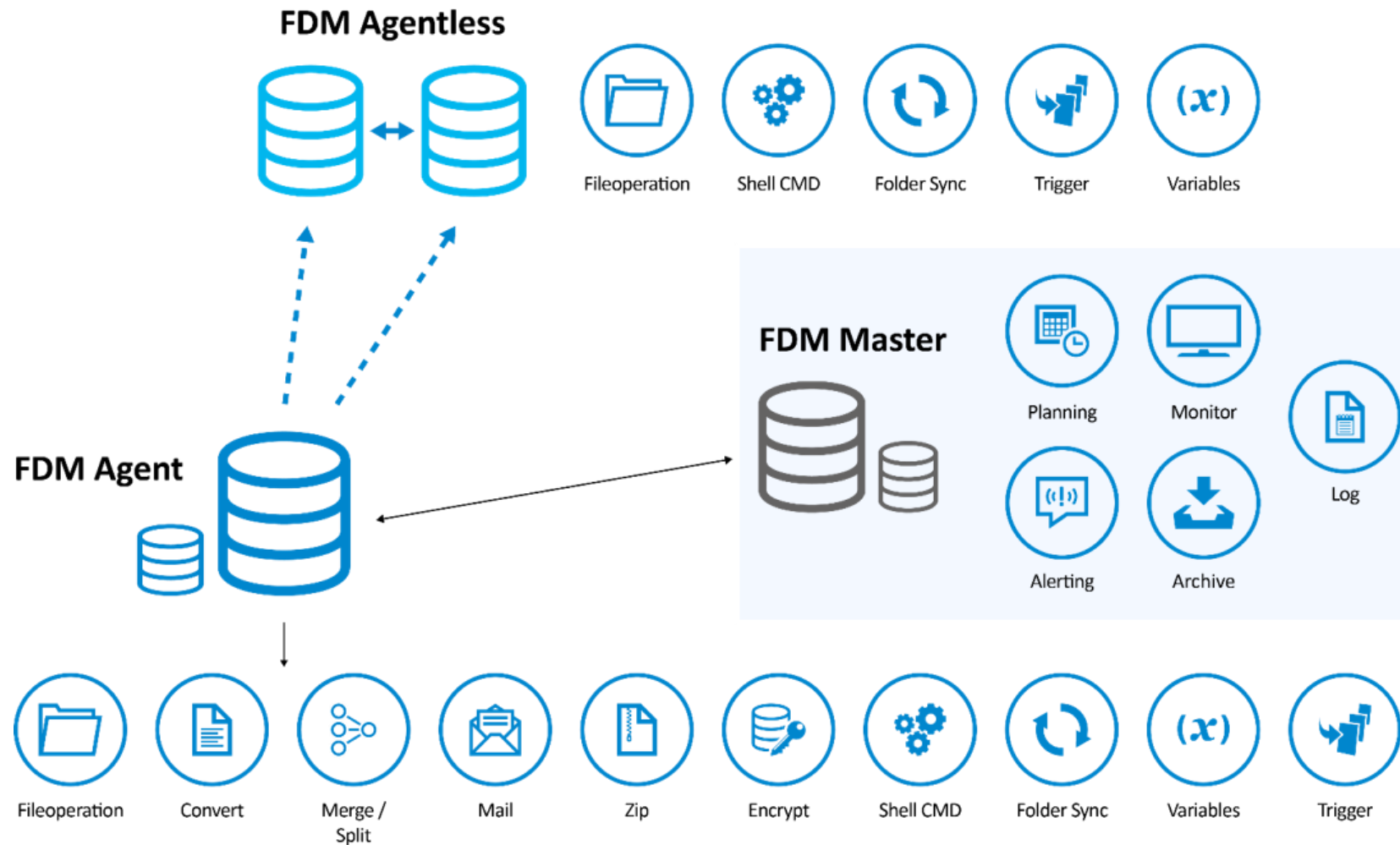
- » High degree of automation for file handling
- » Convenient and uniform process setup, process control, and process design in SAP
 - » Continuous data check and comprehensive error handling, alerts
 - » Connect to the ServiceDesk and additional applications, like using WebServices
- » Versatile functions for reporting and monitoring
- » Eliminate scripting, save valuable system resources
- » Compliance conformity (track all information for settings, users, files, and processes)
- » Event-controlled and time-controlled settings
- » Integrate with existing calendars, like SAP
- » Any file, format, and size
- » Any file protocol FTP, SFTP, SMB, Windows file release, etc.

BatchMan File & DataManager consists of two components. One part is the central controlling FDM Cockpit (BatchMan master system), which is installed in any SAP System. The other part is at least one BM-FDM agent. This BM agent takes the executing role; from there all file operations are started later.

Simply put, you can see the system landscape from the perspective of the relevant BM agent in the FDM cockpit of the BM master system.

As already mentioned, it is possible to operate several BM agents simultaneously. Additional BM agents can act as standalone agents (for instance to connect other network areas) or they are summarized in a group to provide load distribution or a failsafe. As a special case, it is possible to install BM agents that are not to take over any executing role, but are only intended to connect an remote system that cannot be connected via standard file transfer protocols (FTP, SFTP and so on).

The picture shows a typical FDM system landscape with a central BatchMan master system (right), which acts as a single point of control and provides the complete GUI with master data maintenance, monitoring, reporting and so on. A BM agent is installed on the central file server (middle) and thus forms the central interface to all other systems that are to be connected to the file system landscape. These systems can be connected via various protocols such as FTP, SFTP, SAP (RFC), SMB, etc. If none of these protocols is available on a system, it can be connected via the installation of a separate BM agent.



2. Installation and setup

2.1. General information

BatchMan agent is implemented in Java and is thus portable for each operating system that supports Java > 1.8 .

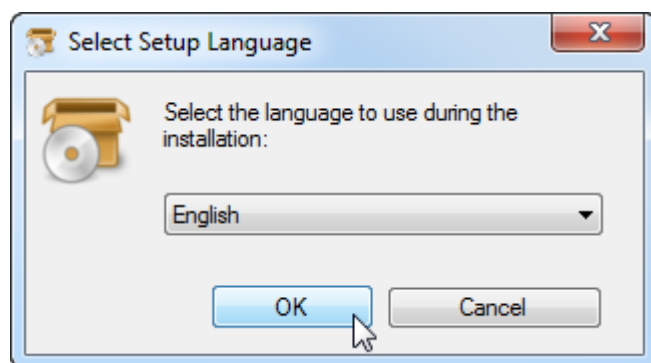
We recommend a free openjdk version e.g. from amazon (<https://aws.amazon.com/de/corretto/>)

Communication between agent and SAP system is made via http(s). Note that firewall settings between communication partners may impact this process.

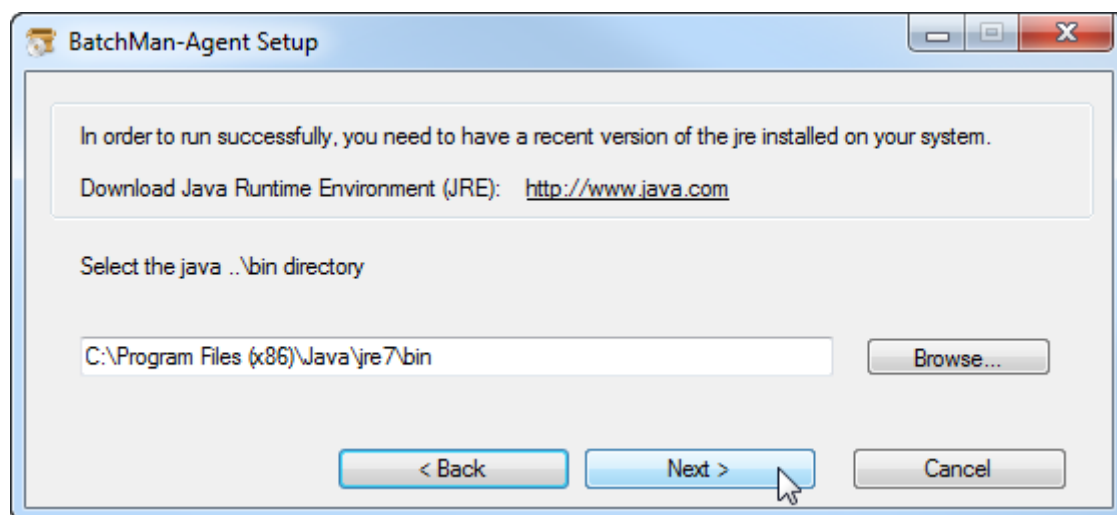
2.2. Agent installation in Microsoft Windows

Unzip the attached zip file and start the AgentSetup.exe file included.

First, select the language you want to use during the installation.



On the following dialog box, select the installation directory (...bin directory) of the Java installation (JRE) that you want to use for executing the BM agent later.



The BM agent runs as system service in Microsoft Windows.

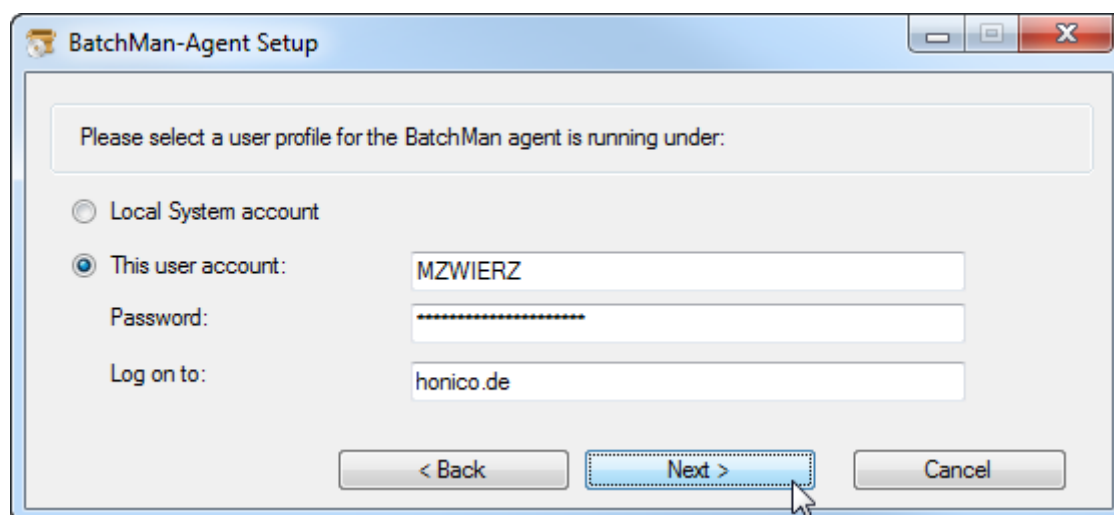
This system service can run either under a local user, under a user account from a connected network domain, or under the local system account.

The various user types differ mainly in the authorizations needed later for the file system and Windows network shares, and so on.

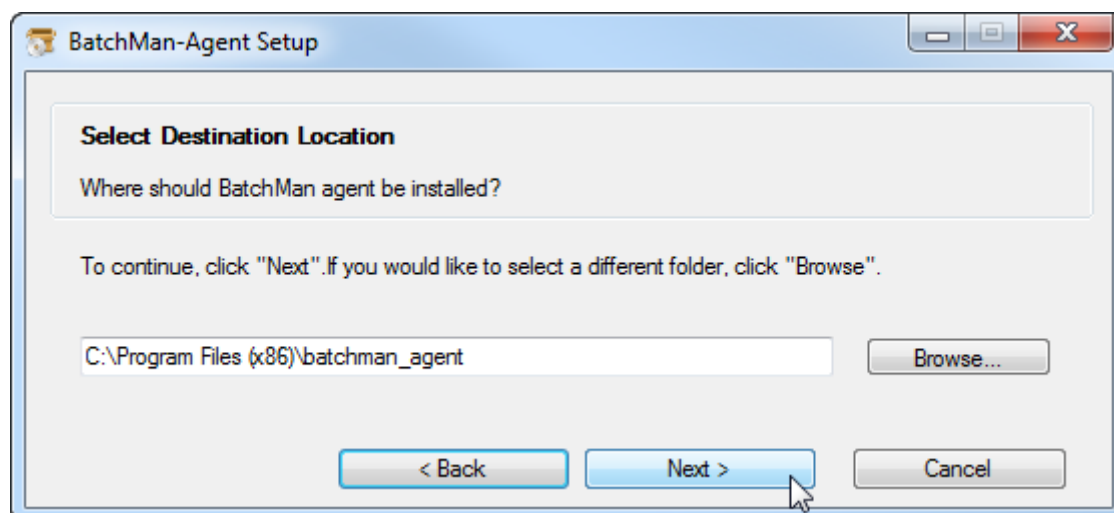
For a local user, enter the host name of the related PC into the "Logon to:" line, for a domain user, enter the relevant name of the domain into this line.

If the service is to run under the local system account, then you do not need any further logon data.

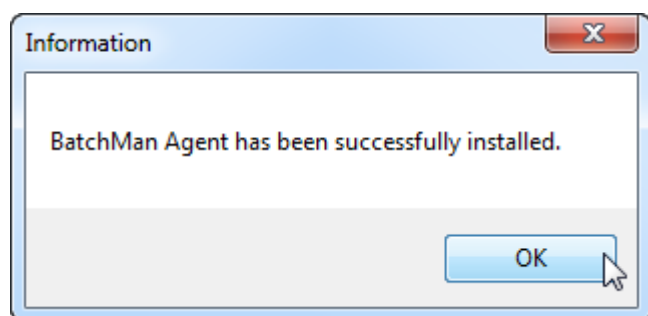
You can change the user account defined here (if needed) later in the Windows control panel by editing the related service.



After you enter the user data, define the target directory for the BM agent installation in the following dialog box.



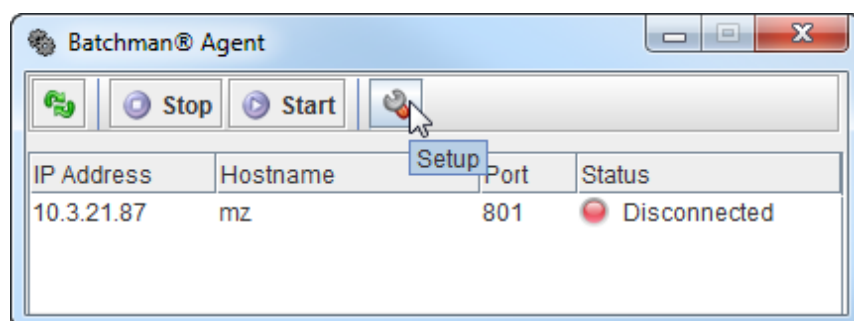
Then the installation is concluded.



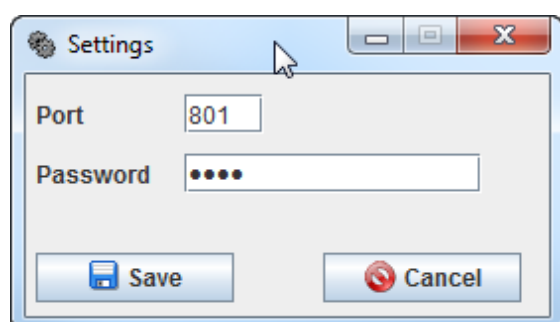
After successful installation, a service called BatchMan agent is set up.

On the desktop folder of the user who made the installation, the system created a link to start and stop the BM agent. Copy this link if needed to the Desktop folders of other users.

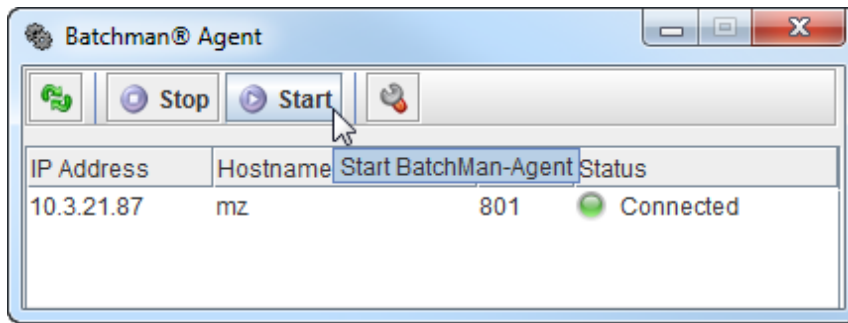
Start the link created to start the Agent Monitor.



To set up the BM agent, choose the related configuration icon. In the following dialog box, you have the option to define a port for communicating with BatchMan in the SAP System, you can also assign a password to secure this communication. You will have to repeat both entries later when setting up the connection in the SAP System.



You can use the start/stop buttons to start and stop the BatchMan agent; this is the same as starting and stopping the service set up. You can thus start/stop the BM agent also using the control panel from Windows.



The agent is now active (LED is green).
 If you like, you can close the monitor now (using the X on the upper right).
 You can restart it at any time via the link on your desktop.

2.3. Agent installation in Unix/Linux

Copy the BatchMan agent directory into a respective work directory. To start the BatchMan agent monitor, enter the following command in a shell:

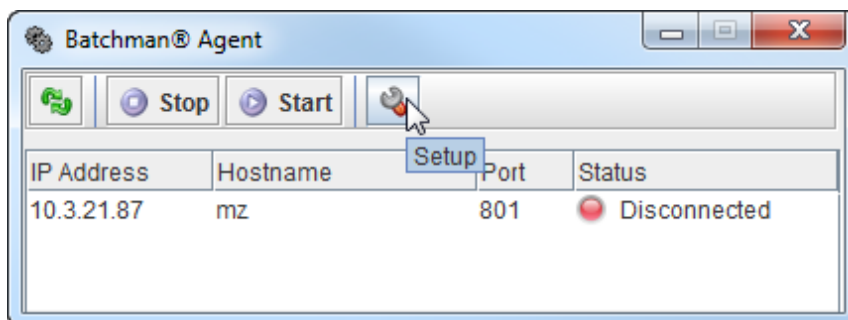
```
[JAVA_HOME]/java -jar [WORKDIR]/batchman_agent/bin/batchman_agent.jar &
```

In this case, replace [JAVA_HOME] with your respective Java installation directory [WORKDIR] with the work directory selected above.

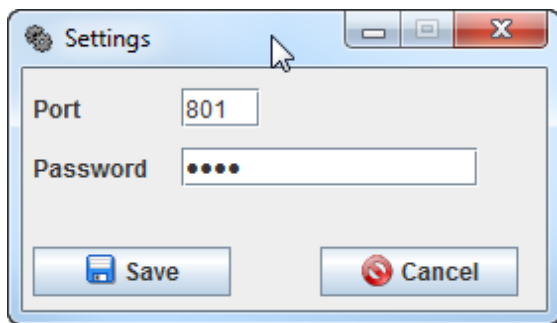
For instance:

```
/usr/java/jre1.6.0_10/bin/java -jar /root/work/batchman_agent/bin/batchman_agent.jar &
```

For your BatchMan agent monitor, you can also create a link (with the execution path mentioned above) that starts the Monitor by just double-clicking.



To set up the BM agent, choose the related configuration icon. In the following dialog box, you have the option to define a port for communicating with BatchMan in the SAP System, you can also assign a password to secure this communication. You will have to repeat both entries later when setting up the connection in the SAP System.

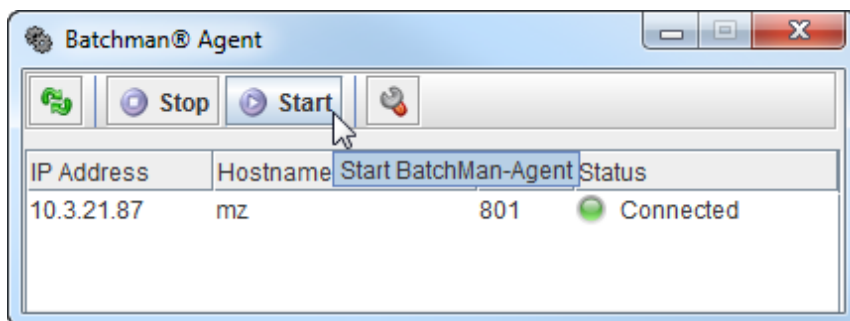


By using the start/stop buttons, you can start the BatchMan agent.

The start button carries out this command:

```
nohup [JAVA_HOME]/java -jar -Xrs [INST-DIR]/batchman_agent/bin/batchman_agent.jar -start
```

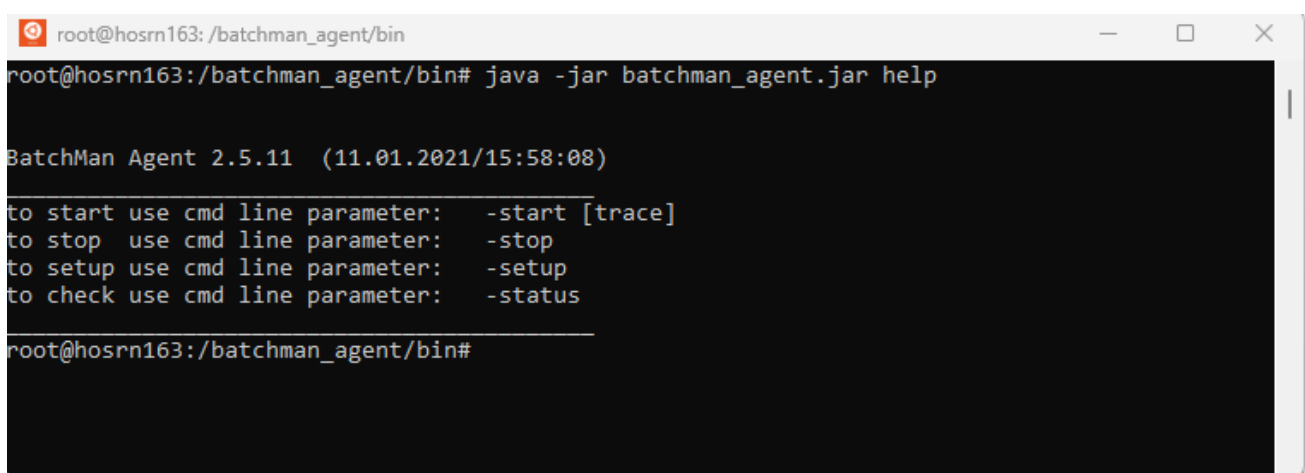
Using this command, you can start the BM agent also without the GUI.



After you start the BM agent, it is active, the LED is green, and it has the connected status.

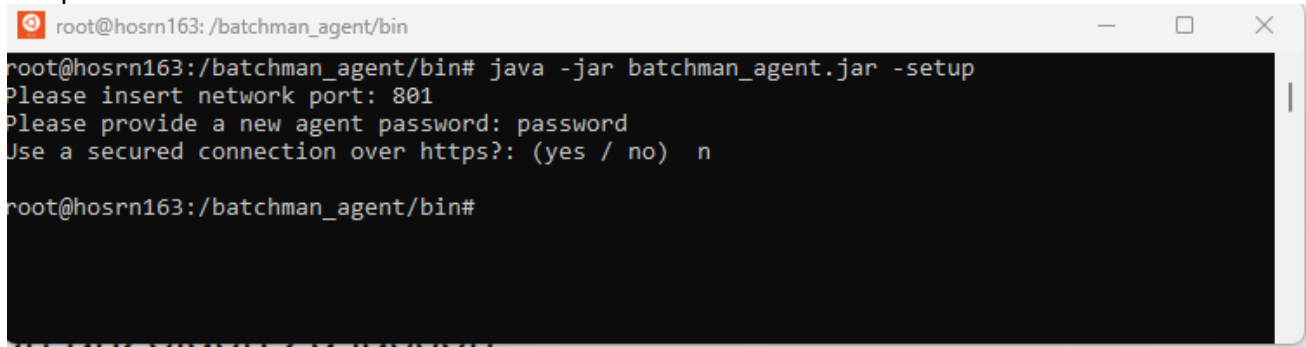
After you start, you can close the monitor at any time (using the X on the upper right) and reopen it. The BatchMan agent runs under no hub thus remaining active even without a shell or after a logoff.

It is also possible to set up and operate the agent without a GUI, type command "java -jar batchman_agent.jar help" in the Agents /bin Folder to display the available command line operations.



Start "java -jar batchman_agent.jar -setup" to setup the agent. Provide a chosen port and password to the setup and take a note for later. You will need this to connect to the Agent. Also chose the

protocoll that shall be used. If you chose https you will need to set it up in the Agent according to Chapter 2.6

A terminal window titled 'root@hosrn163: /batchman_agent/bin' with standard window controls. The terminal shows the execution of 'java -jar batchman_agent.jar -setup'. It prompts for a network port (801), a new agent password (password), and whether to use a secured connection over https (no).

```
root@hosrn163: /batchman_agent/bin# java -jar batchman_agent.jar -setup
Please insert network port: 801
Please provide a new agent password: password
Use a secured connection over https?: (yes / no) n
root@hosrn163: /batchman_agent/bin#
```

2.4. BM setup

To access the setup of the BatchMan, choose:

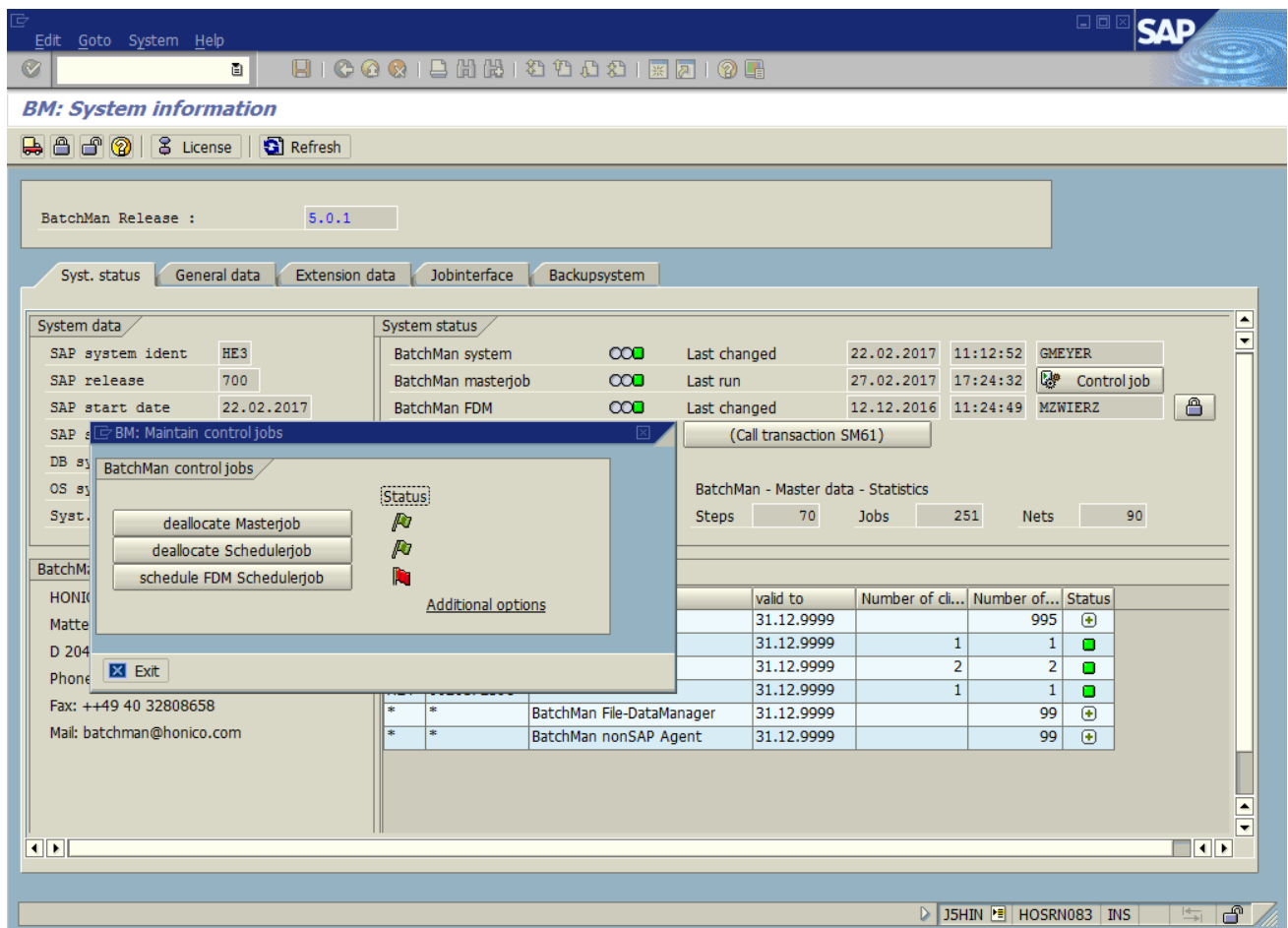
BatchMan® - Function tree → Customizing → Setup programs → System information

To activate the FDM, choose the button [Start FDM] in the status bar “BatchMan FDM”, then the LED of the FDM system status changes from red to green.

To deactivate the FDM, choose the button [Start FDM] in the status bar “BatchMan FDM”, then the LED of the FDM system status changes from green to red.

Be aware that the BM-FDM only runs within an active BM System. Switching off the BM thus prevents also any FDM automatic mode.

To automate the FDM, you need an FDM scheduler job (program; /BTCMAN/M_MFI). This program runs periodically and checks scheduling dates, start criteria, starts processing, and checks their status. You can schedule or unschedule this job comfortably using the Control jobs button. You can also schedule the program manually using the SAP job administration (SM36).



By default, the scheduler program (BTCMAN/M_MFI) uses two dialog processes for scheduling and status checks. If there is a great scheduling load or job runtimes > job periodicity, you can increase the number of work processes used. For this, create a new variant for the scheduler program, enter the desired number of work processes there, and then plan the FDM scheduler job with this new variant.

2.5. Setting up systems with BatchMan Agent

To get to the system set up, choose:

BatchMan® Function tree → Customizing → Technical customizing → non-SAP server definitions

Here you can maintain the links to your BatchMan agent.
Enter any link name and click left on the Create button (F5).

As the type, select BatchMan agent. Enter IP/host name, port and the password defined in the agent (cf. 1). If the agent can only be reached by proxy, maintain the corresponding fields also.
Click on Save and then Check. If the status is green, the link is successful.

2.6. Installation https

To use https, you need encryption certificates.

There are two different types, so-called CA certificates issued by a certification authority or self-issued certificates.

Self-signed certificates carry a certain security risk, which you can neglect in this special case, if you observe the following basic points:

- Issue the certificates yourself, do not take any existing certificates unless they come from a trusted CA.
- The file **serverkeys** contains the decisive private key that secures the https communication. Therefore, only put this file in the agent directory and then delete all copies. Never send this file by mail.
- Save the agent directory with the appropriate authorizations
- You later enter the public-key (**server.cer**) of this certificate in the SAP system (transaction STRUST). This means that you trust the associated private key (**serverkeys**). Do not play certificates of unknown origin in the STRUST (especially in this case, if you certify the agent URL).

2.6.1. Use of self-issued certificates

To generate certificates yourself please use the keytool from the java JRE directory with the following parameters:

```
keytool.exe -genkey -alias server-alias -keyalg RSA -keypass PASSWORD -storepass PASSWORD -keystore serverkeys -validity 6000
```

```
keytool.exe -export -alias server-alias -storepass PASSWORD -file server.cer -keystore serverkeys
```

```
keytool.exe -import -v -trustcacerts -alias server-alias -file server.cer -keystore server.jks -keypass PASSWORD -storepass PASSWORD
```

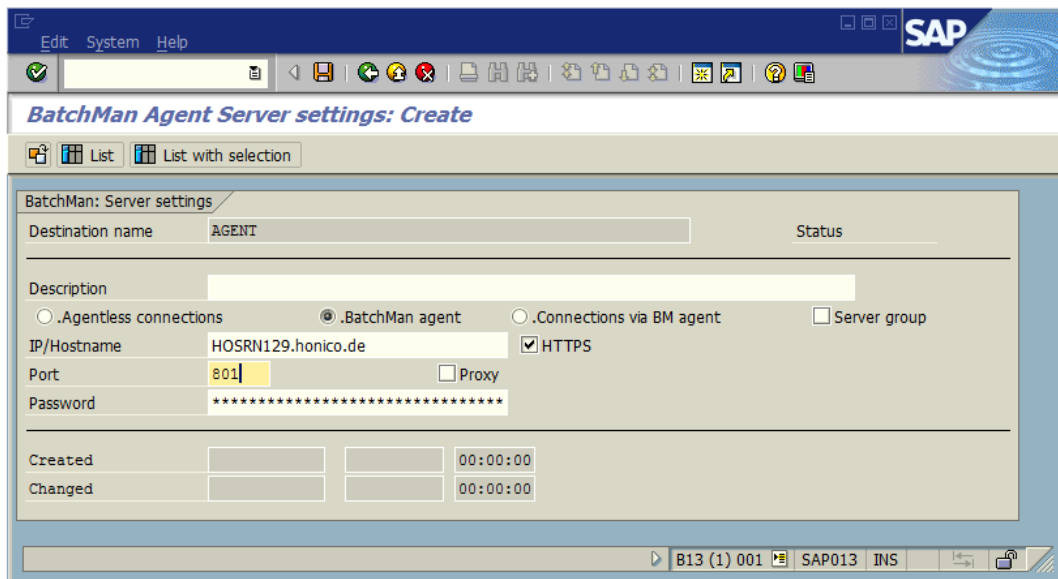
You can change the PASSWORD in the parameters storepass and keypass, but it must be the same for both parameters.

Answer inquiries (name, organization, etc.) with the IP / hostname or domain / URL, which you later use to contact the agent in the BM.

Example:

Agent IP/Hostname = HOSRN129.honico.de

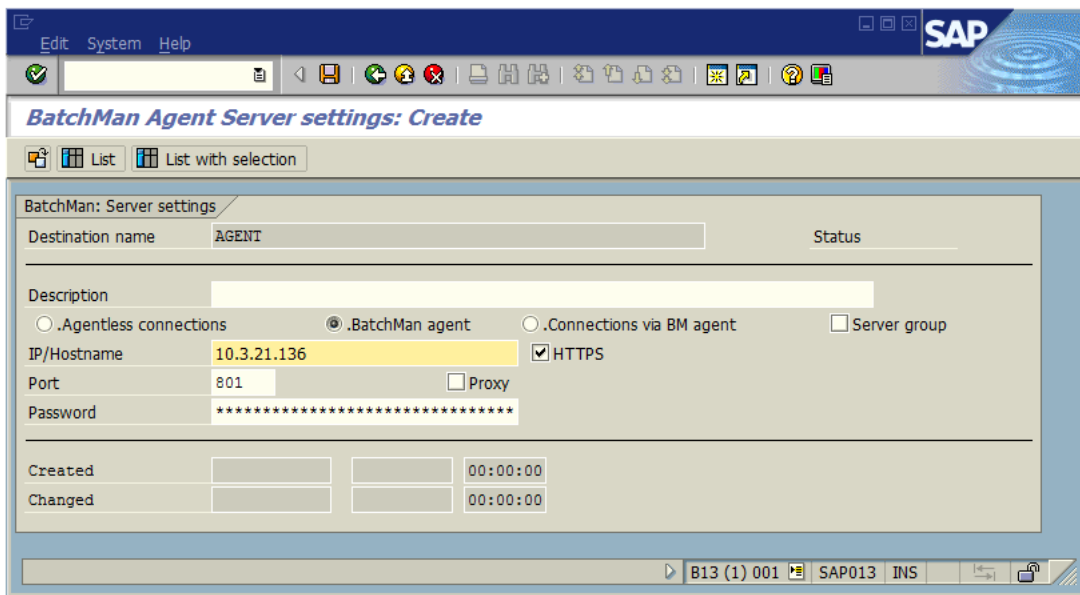
→ Answer all certificate questions with HOSRN129.honico.de.



Example:

Agent IP/Hostname = 10.3.21.136

→ Answer all certificate questions with 10.3.21.136



That's what it would look like:

```

root@vm: /home
root@vm:/home# keytool -genkey -alias server-alias -keyalg RSA -keypass iqRmVt5
#-R3vxqslC33 -storepass iqRmVt5#-R3vxqslC33 -keystore serverkeys -validity 6000
What is your first and last name?
[Unknown]: HOSRN129.honico.de
What is the name of your organizational unit?
[Unknown]: HOSRN129.honico.de
What is the name of your organization?
[Unknown]: HOSRN129.honico.de
What is the name of your City or Locality?
[Unknown]: HOSRN129.honico.de
What is the name of your State or Province?
[Unknown]: de
What is the two-letter country code for this unit?
[Unknown]: de
Is CN=HOSRN129.honico.de, OU=HOSRN129.honico.de, O=HOSRN129.honico.de, L=HOSRN1
29.honico.de, ST=de, C=de correct?
[no]: yes

root@vm:/home# keytool -export -alias server-alias -storepass iqRmVt5#-R3vxqslC
33 -file server.cer -keystore serverkeys
Certificate stored in file <server.cer>
root@vm:/home# keytool -import -v -trustcacerts -alias server-alias -file serve
r.cer -keystore server.jks -keypass iqRmVt5#-R3vxqslC33 -storepass iqRmVt5#-R3v
xqslC33
Owner: CN=HOSRN129.honico.de, OU=HOSRN129.honico.de, O=HOSRN129.honico.de, L=HO
SRN129.honico.de, ST=de, C=de
Issuer: CN=HOSRN129.honico.de, OU=HOSRN129.honico.de, O=HOSRN129.honico.de, L=H
OSRN129.honico.de, ST=de, C=de
Serial number: 3d329571
Valid from: Wed Jan 24 22:38:58 CET 2018 until: Thu Jun 29 23:38:58 CEST 2034
Certificate fingerprints:
    MD5: 38:7E:B2:EC:DE:7C:0E:01:9D:F3:00:CE:DA:97:63:5C
    SHA1: 19:E6:C6:7C:CC:83:12:6D:82:73:CE:B7:A0:15:6B:B8:61:90:BA:FF
    SHA256: 00:4C:2D:02:3E:A1:02:CC:A2:E1:4D:03:C0:77:B6:90:BE:30:21:33:81
:70:5F:F9:D2:D2:5B:AA:4F:1A:0A:A8
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: CC 19 F0 4B 6B E9 B4 4B 10 AF 91 49 DD D9 2F 5F ...Kk..K...I../_
0010: 30 A2 0E FE 0...
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore
[Storing server.jks]
root@vm:/home#

```

If you want to run multiple agents, you have multiple Agent IP / Hostnames - so you will need either a separate certificate for each Agent IP / Hostname at its specific address, or you will use wildcards in the certificate path.

Example:

You create a certificate with *.honico.de, this certificate is then valid for all agents whose IP / hostname matches the pattern *.honico.de.

RFC2818 applies [3.1. Server Identity] (<https://www.ietf.org/rfc/rfc2818.txt>)

You now have the encryption certificates in the form of 3 files (server.jks, serverkeys, server.cer)

2.6.2. Use of CA certificates

Please pay attention to the validity of the certificate. By the time the certificate expires, the communication with the agent ends !!!

Use validity periods that are not subject to the risk of expiration. In point 2.6.1 we use, for example, certificates with a validity period of approx. 16 years.

Make sure that your CA certificate in the CN root contains the IP / hostname of the agent to be called (possibly with wildcards) (see also 2.6.1 Use of self-issued certificates).

If you have a CA certificate, you need to convert this to the java keystore format, therefore you must have your CA certificate in PKCS12 format (or be converted).

For the conversion into the java keystore format use the keytool from the java JRE directory with the following parameters:

```
keytool.exe -v -importkeystore -srckeystore CA.p12 -srcstoretype PKCS12 -destkeystore serverkeys -keypass  
PASSWORD -storepass PASSWORD
```

```
keytool.exe -export -storepass PASSWORD -alias 1 -file server.cer -keystore serverkeys
```

```
keytool.exe -import -v -trustcacerts -alias 1 -file server.cer -keystore server.jks -keypass PASSWORD -storepass  
PASSWORD
```

CA.p12 in this example is your CA certificate in PKCS12 format.

You can change the PASSWORD in the parameters storepass and keypass, but it must be the same for both parameters.

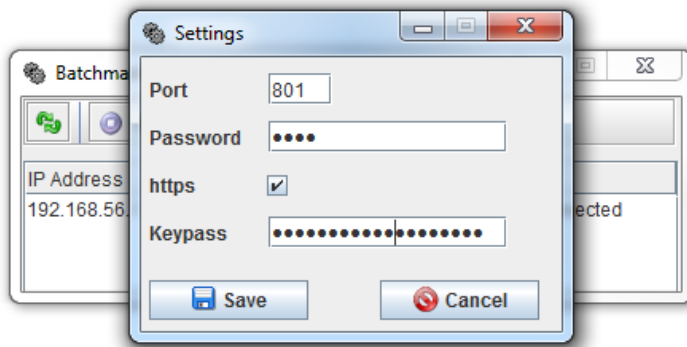
You now have the encryption certificates in the form of 3 files (server.jks, serverkeys, server.cer)

2.6.3. Switch agent to https

Copy the files *server.jks*, *serverkeys* into the directory of the BatchMan Agent
..\batchman_agent\bin\.

It is possible that these files already exist there, then please overwrite them.

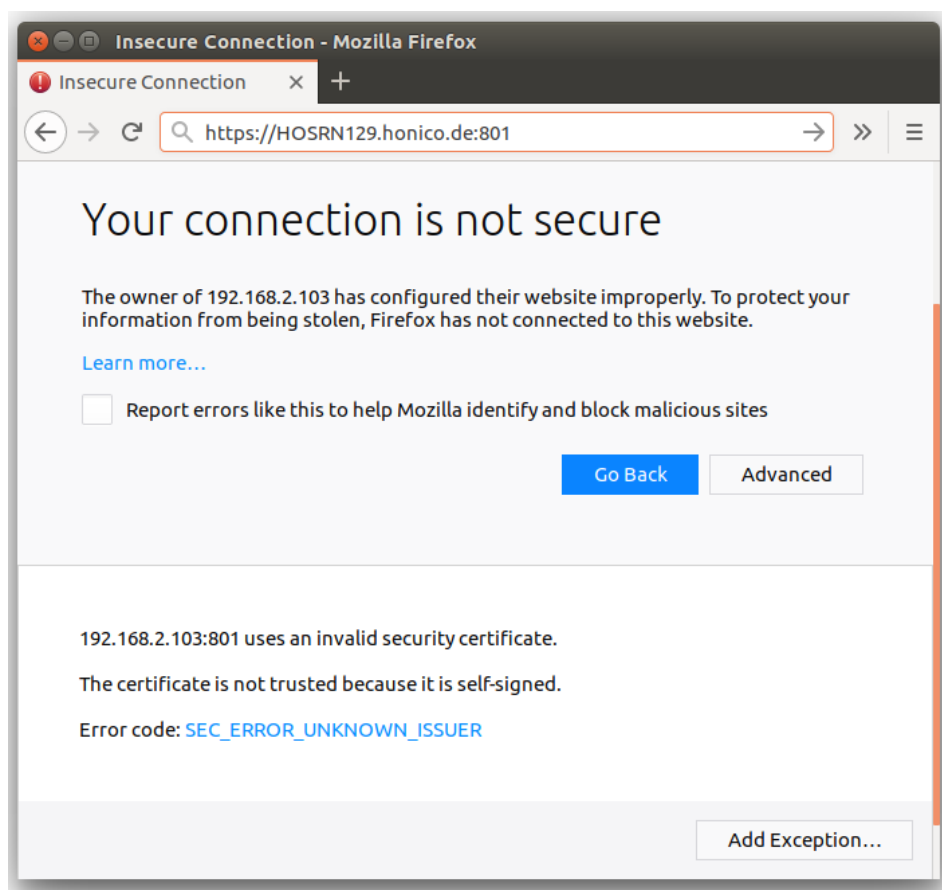
Now please enable https in the Agent Settings and set the PASSWORD at Keypass, which you used above as -keypass or -storepass.



You can then restart the BatchMan Agent.

The agent is now running on https. You can test this in the browser by calling [https://\[IP/Hostname\]:\[Port\]](https://[IP/Hostname]:[Port]).

If you have signed the certificate yourself according to 2.6.1 a security warning appears. This security warning also appears if you use CA certificates that do not have a trust relationship based on your browser settings.



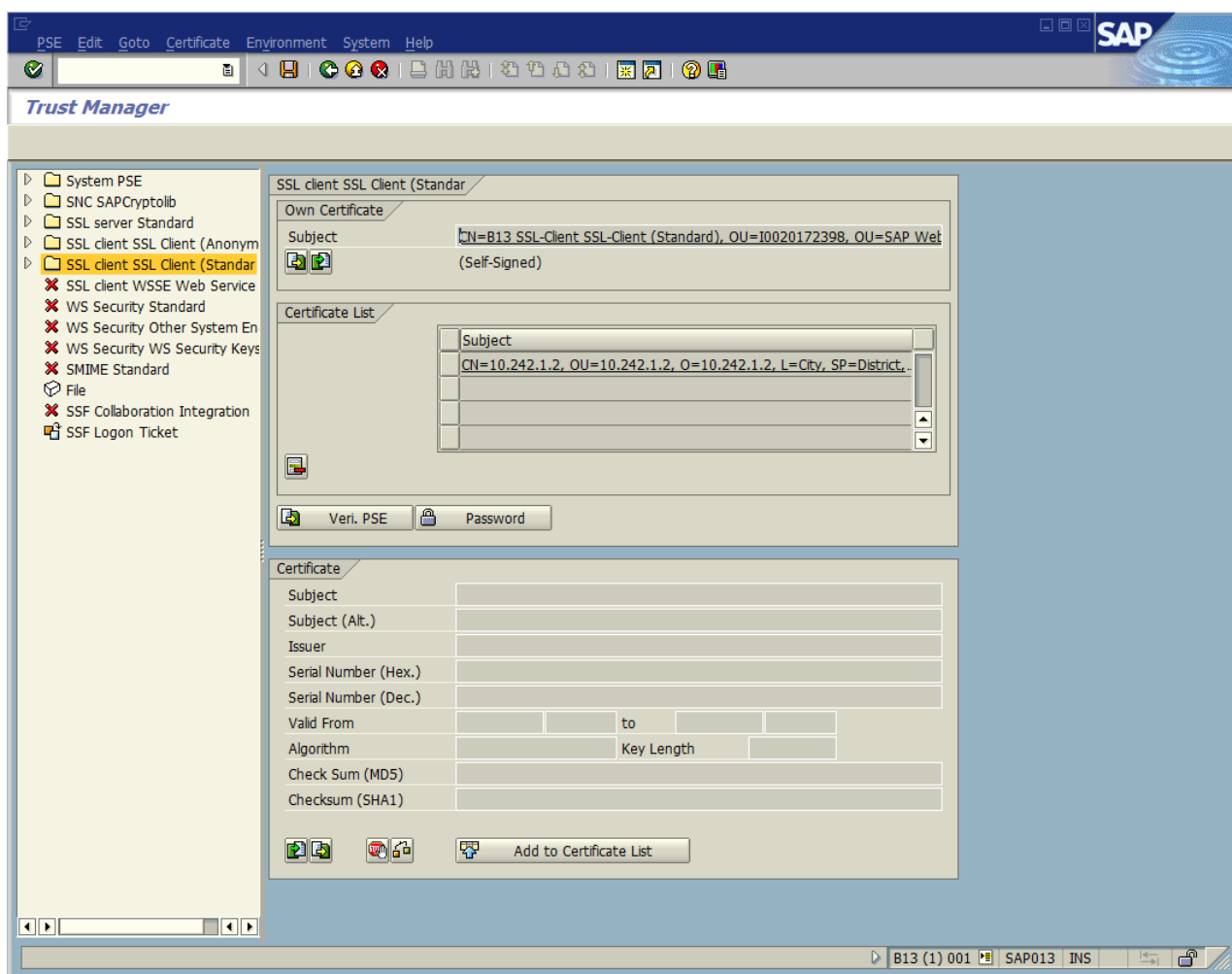
Warnings that the certificate is untrusted can be accepted with the security measures described above.

Please note, however, that no warnings appear stating that the security certificate was issued for a different address / URL. This would mean that your certificate was not issued for the IP / Hostname or Domain / URL of that agent (see above).

2.6.4. Switch FDM to https

The agent is now running on https, now we have to change the SAP page to https. To do this, open transaction STRUST and in the menu tree on the left

Double-click *SSL Client SSL Client (default)*.



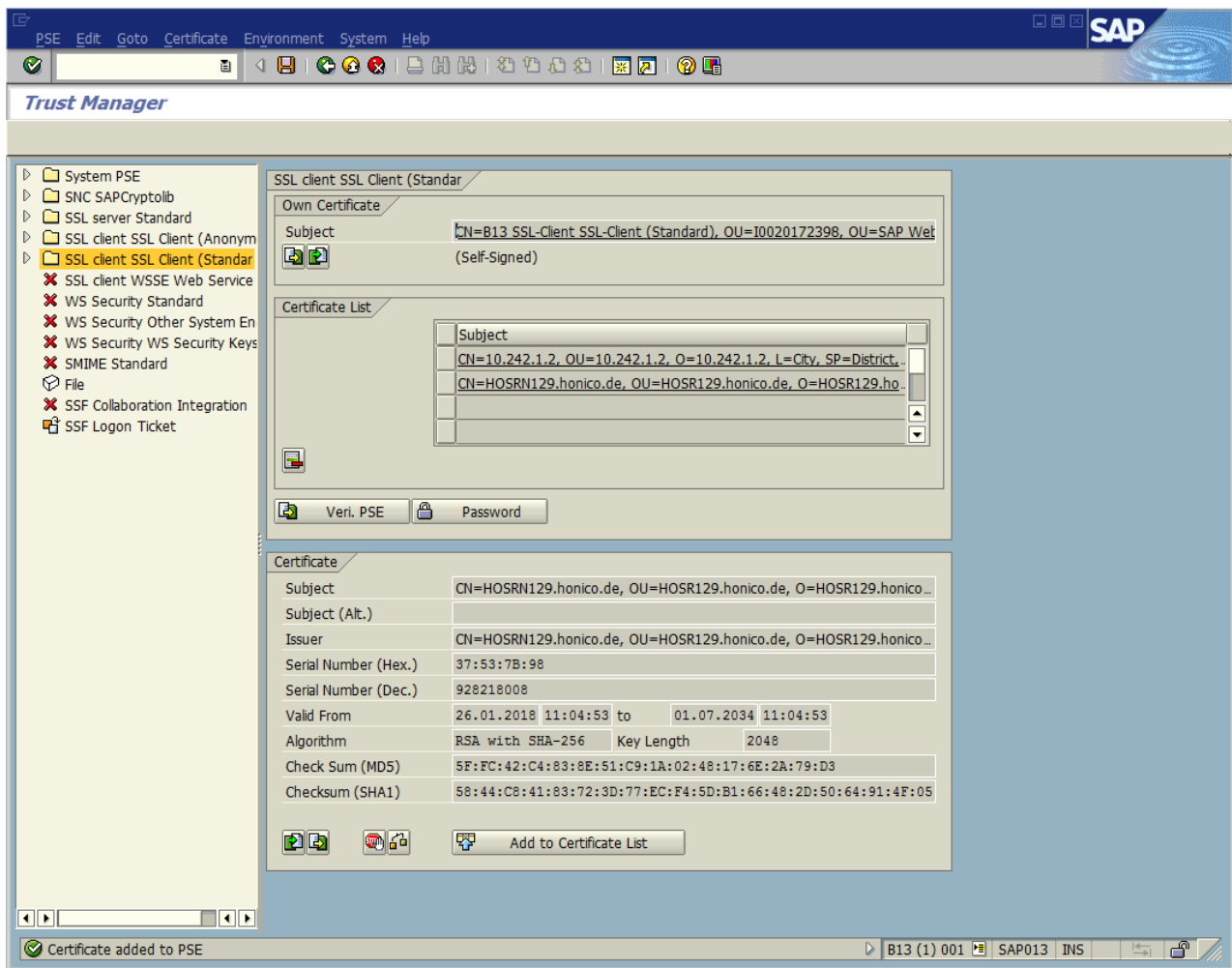
(If this entry is still a red cross, then please first [right mouse button] -> Create)

Now go right, bottom left on the button:

Import certificate

Select the file *server.cer* here (see 2.6.1.resp.2.6.2).

Then click on the button *Add to Cert. List*.



Now you just have to change the type to https in the server connections of BatchMan:
JSHEN->Customizing->Technical Customizing->Serverdefinitionen non-SAP
 Select your agent connection and set the flag for https.

BatchMan Agent Server settings: Create

BatchMan: Server settings

Destination name: AGENT_HOSRN129 Status: _____

Description: _____

☐ .Agentless connections
 ☒ .BatchMan agent
 ☐ .Connections via BM agent
 ☐ Server group

IP/Hostname: HOSRN129.honico.de ☒ HTTPS

Port: _____ ☐ Proxy

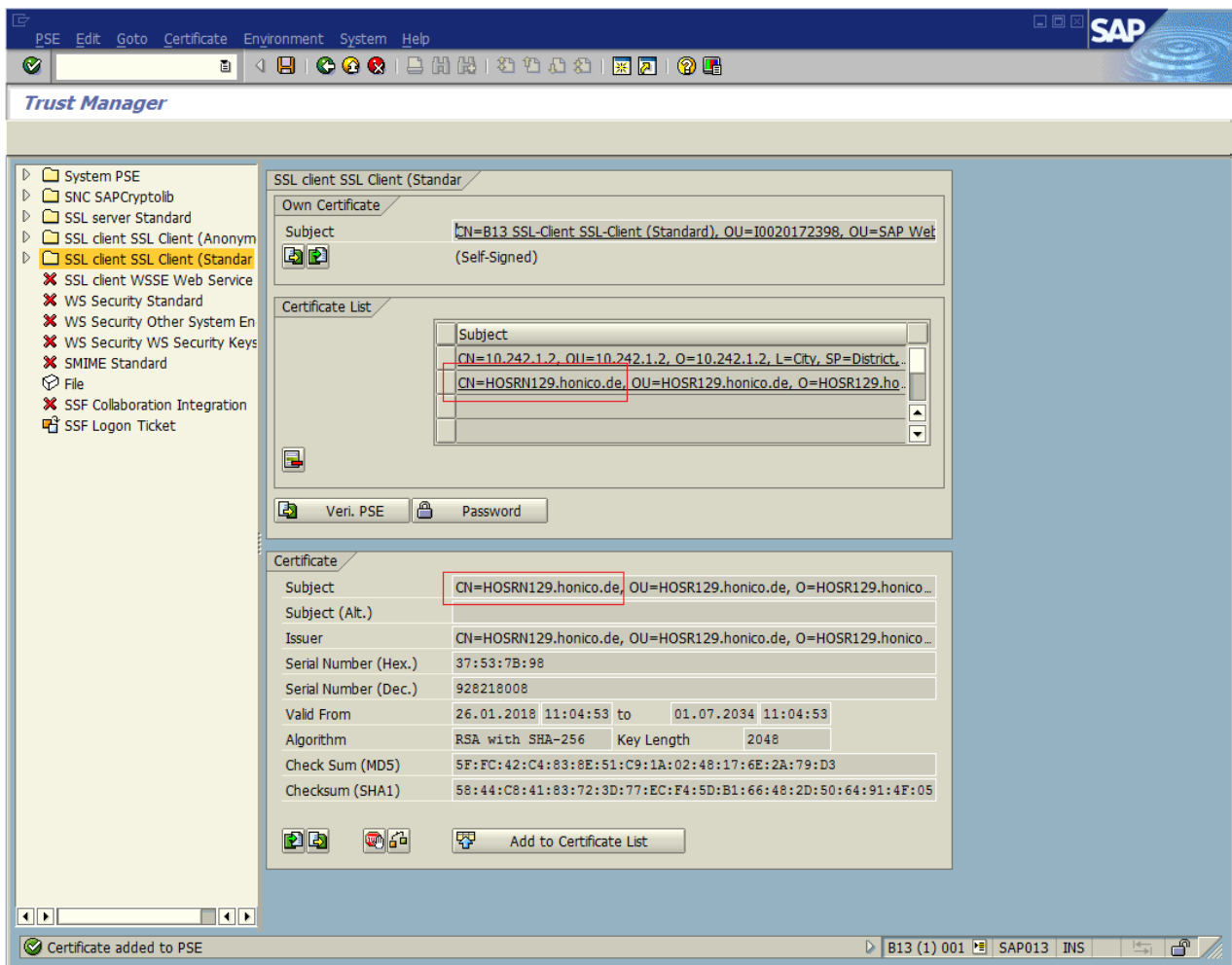
Password: *****

Created: _____ 00:00:00

Changed: _____ 00:00:00

B13 (1) 001 SAP013 INS

Pay attention again to the fact that under *IP / Hostname* a value is available, which corresponds to the pattern (CN = ...) of the above recorded certificate.



2.7. Setting up load distribution and failsafe

If you have installed the BatchMan agent on a second system and you want load distribution or failsafe with this system, enter the second system like the system described above.

BatchMan Agent Server settings: Maintain

Destination name: FILESERVER_P1 Status: _____

Description: Fileserver (BM-Agent)

☐ .Agentless connections
 ☒ .BatchMan agent
 ☐ .Connections via BM agent
 ☐ Server group

IP/Hostname: 10.3.21.75 ☐ HTTPS

Port: 5667 ☐ Proxy

Password: *****

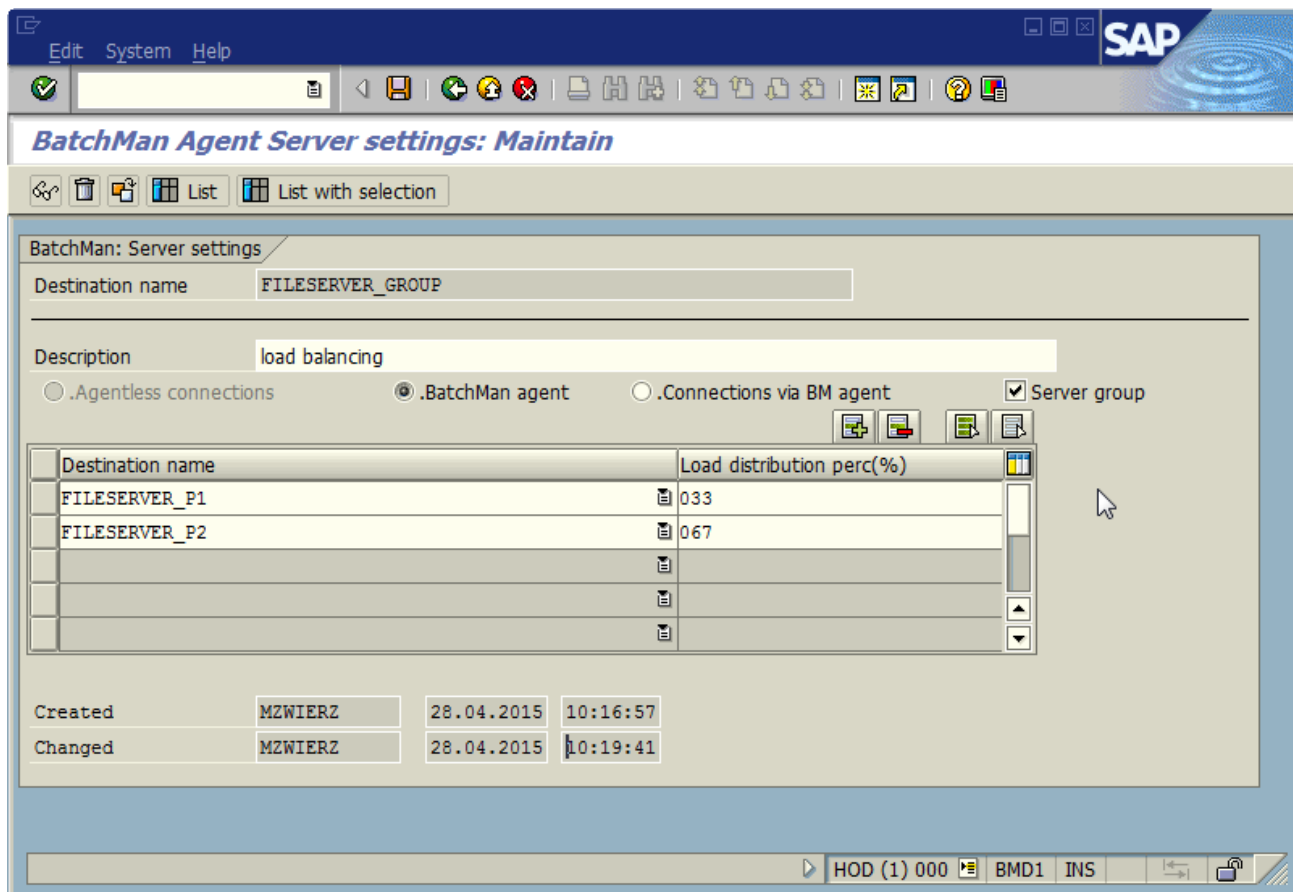
Created	MZWIERZ	17.07.2014	10:36:09
Changed	MZWIERZ	28.04.2015	10:13:33

Navigation: HOD (1) 000 BMD1 INS [Icons]

Then you have the option to combine these systems.

For this, select a new link name and click on the Server group field.

You then have the option to combine several defined links in one group. For this, select the corresponding links with the desired load distribution share. If a system is only to be used as a failsafe, it must be given a load distribution share of 0%. This system will then only be used if all other systems have failed or can no longer be reached.



2.8. Set up: System landscape

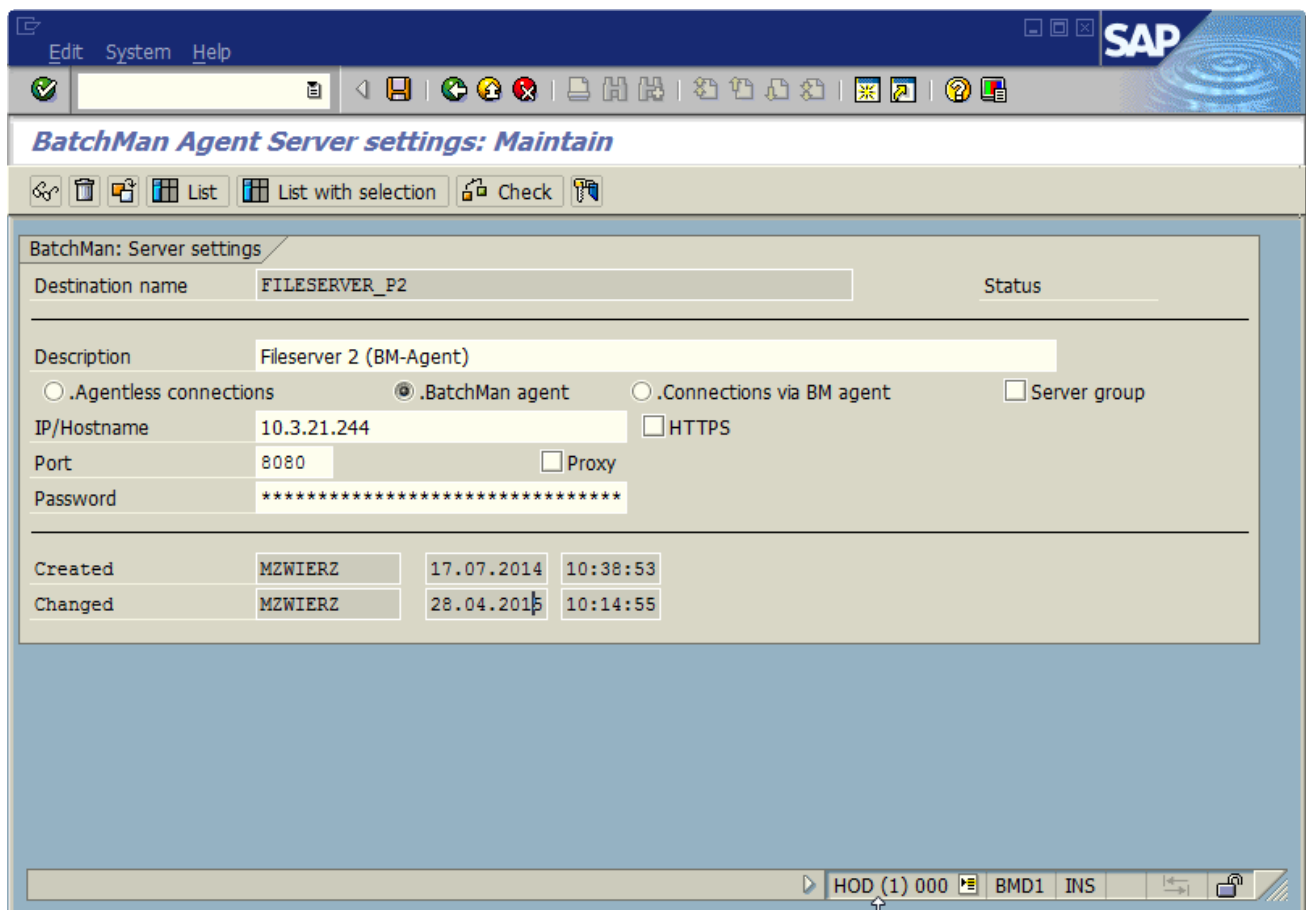
For BatchMan agent to communicate with other systems, you have to maintain the related links to these systems. All connected systems communicate directly and exclusively with the BatchMan agent (via BM agent).

2.8.1. FTP connections

To set up an FTP connection, enter a connection name and select Connection as the connection type via BM agent and the file protocol FTP. Then select a BatchMan agent (agent over which the connection runs). Then into the related fields enter the IP of the FTP server and the values for user and password to logon to the FTP server.

By default, the system uses the active FTP mode and the binary (1:1) transfer, if you want to use the passive FTP mode for the related FTP connection or you need an ASCII conversion during the transfer, you can change the value accordingly.

After you save the connection data, you can test the connection with the Check button. The system tests the availability of the system as well as the successful logon with the user data entered. If the connection test is successful, the system sets the status to green, otherwise the system issues a corresponding error message in the status bar below.



2..8.2. SFTP connections

To set up an SFTP connection, enter a connection name and select Connection as the connection type via BM agent and the file protocol SFTP. Then select a BatchMan agent (agent over which the connection runs). Then enter into the related fields the IP of the SFTP server and the values for user and password to logon to the SFTP server.

If you need a key file for the SFTP logon, then into the Key file field enter the storage location where the related private key file is located on the BatchMan agent. If the key file has additional encryption, enter the related password into the Password field, otherwise leave this field empty.

After you save the connection data, you can test the connection with the Check button. The system tests the availability of the system as well as the successful logon with the user data entered. If the connection test is successful, the system sets the status to green, otherwise the system issues a corresponding error message in the status bar below.

BatchMan Agent Server settings: Maintain

BatchMan: Server settings

Destination name: HONICO_SFTP_(244) Status

Description: HONICO_SFTP

☐ .Agentless connections
 ☐ .BatchMan agent
 ☒ .Connections via BM agent

BatchMan Agent: FILESERVER_P1

Protocoltype: SFTP SFTP prot...

IP/Hostname: 10.3.21.244

User: Administrator

Password: *****

Keyfile:

Created	MZWIERZ	25.03.2015	10:48:38
Changed	MZWIERZ	28.04.2015	10:44:19

HOD (1) 000 BMD1 INS

2.8.3. SAP connections

To set up an SAP connection, enter a connection name and select Connection as the connection type via BM agent and the protocol SAP. Then select a BatchMan agent (agent over which the connection runs). Into the IP/Hostname and system number fields, enter the related values and set the SAP user for which the future connection is to run (client, user, password).

After you save the connection data, you can test the connection with the Check button. The system tests the availability of the system as well as the successful logon with the user data entered. If the connection test is successful, the system sets the status to green, otherwise the system issues a corresponding error message in the status bar below.

BatchMan Agent Server settings: Maintain

BatchMan: Server settings

Destination name: Status:

Description:

☐ .Agentless connections ☐ .BatchMan agent ☒ .Connections via BM agent

BatchMan Agent:

Protocoltype:

IP/Hostname:

Systemnumber:

Client:

User:

Password:

Created:

Changed:

HOD (1) 000 BMD1 INS

SAP Default connection to an active response

For the local SAP system, it is possible to set an active response for the jobs. The status of the active jobs will be updated in each run of the scheduler. Thus, a "real" termination of a job and the end registration in the FDM can result in a delay of max. 1 scheduler period. In order to avoid these delays, the active feedback of the jobs is recommended. In this case, the job logs its node independently to the SAP system and the status query in the scheduler is no longer required. To use the active response, set the default flag for the local SAP system.

BatchMan Agent Server settings: Display

List
 List with selection
 Entry
 Entry
 Check

BatchMan: Server settings

Destination name	HE3_SAP_VERBINDUNG	Status
------------------	--------------------	--------

Description

☐ .Agentless connections
 ☐ .BatchMan agent
 ☒ .Connections via BM agent

BatchMan Agent: AGENT

Protocoltype: SAP SAP system

IP/Hostname: 10.3.21.83

Systemnumber: 00

Client: 000

User: mzwierz

Password:

☒ Default sap system for active status feedback

Created	MZWIERZ	12.12.2016	11:03:15
Changed	MZWIERZ	30.01.2020	11:46:44

The Callback User needs Authorizations

S_RFC for /BTCMAN/M_FILEOPERATION

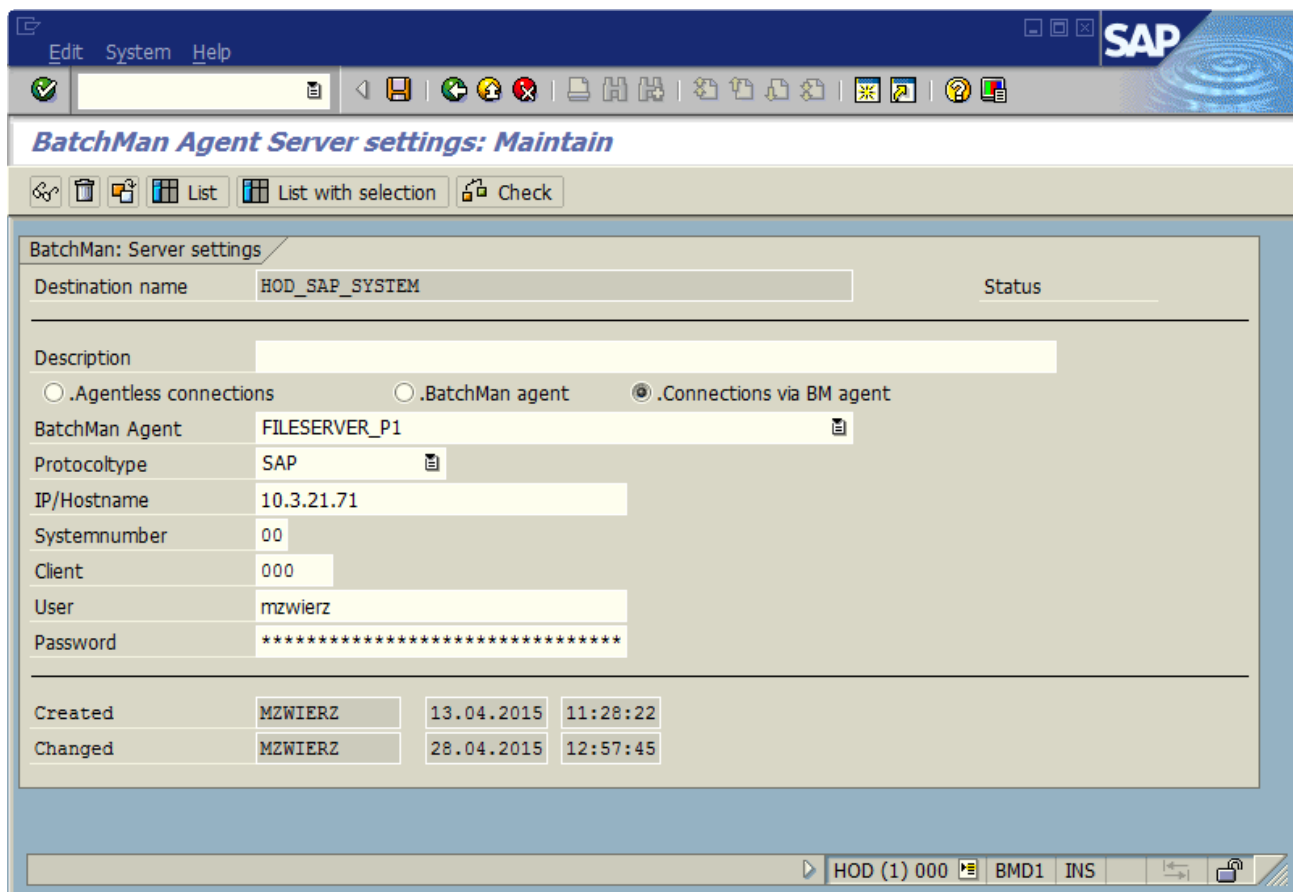
Possibly other Authorizations if the Connection is also used for FDM File Operations via SAP.

2.8.4. SMB connections (Windows file release and SAMBA server)

To set up an SMB connection, enter a connection name and select the Via BM agent connection type and the file protocol SMB. Then select a BatchMan agent (agent over which the connection runs). Then into the related fields enter the IP of the server and the values for user and password to logon to the server.

If this user is a domain user, write the name of the relevant domain before the user name separated by a semicolon ([Domain];[User]).

After you save the connection data, you can test the connection with the Check button. The system tests the availability of the system as well as the successful logon with the user data entered. If the connection test is successful, the system sets the status to green, otherwise the system issues a corresponding error message in the status bar below.



2.8.5. BatchMan agent connection via BatchMan agent

For a BM agent connection via BM agent, the agent is not connected directly to the SAP System, but indirectly via another BM agent, which acts as a proxy. You can also save this kind of connection directly as a BatchMan agent connection, a via connection mainly makes sense if the BM agent cannot be reached directly from the SAP System.

To set up another BM agent, enter a connection name and select the Via BM agent connection type and the file agent file protocol. Then select a BatchMan agent (agent over which the connection runs) and IP, user, and password.

After you save the connection data, you can test the connection with the Check button. The system tests the availability of the system as well as the successful logon with the user data entered. If the connection test is successful, the system sets the status to green, otherwise the system issues a corresponding error message in the status bar below.

BatchMan Agent Server settings: Maintain

BatchMan: Server settings

Destination name: AGENT_BY_AGENT_244 Status: _____

Description: _____

☐ .Agentless connections
 ☐ .BatchMan agent
 ☒ .Connections via BM agent

BatchMan Agent: FILESERVER_P1

Protocoltype: BM BM File Agent

IP/Hostname: 10.3.21.244 ☐ HTTPS

Port: 8080 ☐ Proxy

Password: *****

Created: MZWIERZ 25.03.2015 15:04:18
 Changed: MZWIERZ 28.04.2015 13:14:23

2.8.5. SMTP Connections

To set up an SMTP connection, enter a connection name and select the connection type via BM Agent and the mail protocol SMTP. Then select a BatchMan Agent (agent through which the connection runs). Enter the IP, port and connection type of the server as well as the values for user and password for logging into the server in the corresponding fields. The sender should be given in RFC 2047 format: *Name <mail@mail.com>*

BatchMan Agent Server settings: Maintain

BatchMan: Server settings

Destination name: SMTP Status: _____

Description: mail

☐ .Agentless connections
 ☐ .BatchMan agent
 ☒ .Connections via BM agent

BatchMan Agent: AGENT

Protocoltype: SMTP Protocol

IP/Hostname: mail.gmx.net

Port: 25

Connection type: TLS

Sender: BatchMan Mail <info@honico.com>

User: info@honico.com

Password: *****

Created: MZWIERZ 12.12.2016 11:23:54
 Changed: MZWIERZ 08.10.2020 17:14:20

2.8.5. HTTP Connections

To set up a http connection, enter a connection name and select the connection type via BM Agent and the protocol http. Then select a BatchMan Agent (agent through which the connection runs). Enter the IP and optionally the port in the corresponding fields. If you provide the User and Password fields, the http connection is called with http basic authentication. The IP / Hostname field can contain variables in the form <VARIABLE>, which are later replaced when called up. E.g: <https://jsonplaceholder.typicode.com/posts/<POSTCOUNT>>

BatchMan Agent Server settings: Maintain

Destination name: HTTP Status:

Description:

☐ .Agentless connections ☐ .BatchMan agent ☒ .Connections via BM agent

BatchMan Agent: AGENT

Protocoltype: HTTP protocol

IP/Hostname: https://jsonplaceholder.typicode.com

Port: Proxy: ☐

User:

Password:

System monitoring and alarm

Alertteam: Alerttool:

Created	MZWIERZ	09.10.2020	09:47:23
Changed	MZWIERZ	09.10.2020	09:52:49

3. File interface

Related and dependent file operations are grouped as a file interface. A file interface mainly defines scheduling dates, start criteria, and a list of the file operations to be processed sequentially. Scheduling dates define the day on which the start criteria are checked. Start criteria define the criteria that must be met so that a file interface is started.

For instance:

File interface A defines a weekly scheduling in the scheduling dates, each time on Monday. In the start criteria, file interface A defines a file dependency with file B. Thus, each Monday (scheduling date), the system checks for the existence of file B (start criteria), if the file is there the system starts file interface A.

To get to the master data maintenance of the file interfaces, choose:

BatchMan®-Function tree → File interface → Master data → Maintain file interface

3.1. General data

After you enter a file interface name, you can select the function you want:



Create a new file interface



Edit an existing file interface

The General data tab is where the executing file server (BM agent) is defined. After scheduling is completed and the start criteria have been met, this file server receives the list of the defined file processing and then executes this. All file operations defined in the file interface are controlled from this file server.

Besides the file server, you can maintain a description of the file interface on the General data tab. For the documentation of the file interface, you can specify a link to external documentation or use internal documentation (see BM).

BM: Fileinterface maintain SAP

BM: Fileinterface maintain

Name: Status: inactive

General data | Fileprocessing | Scheduling dates | Start criteria | Alerting

Fileserver:

Description:

Created: MZWIERZ 27.10.2014 15:52:26
Changed: MZWIERZ 27.10.2014 16:23:34

HOD (1) 000 BMD1 INS

After you save a file interface, the system issues further functions on the application toolbar in the display mode:

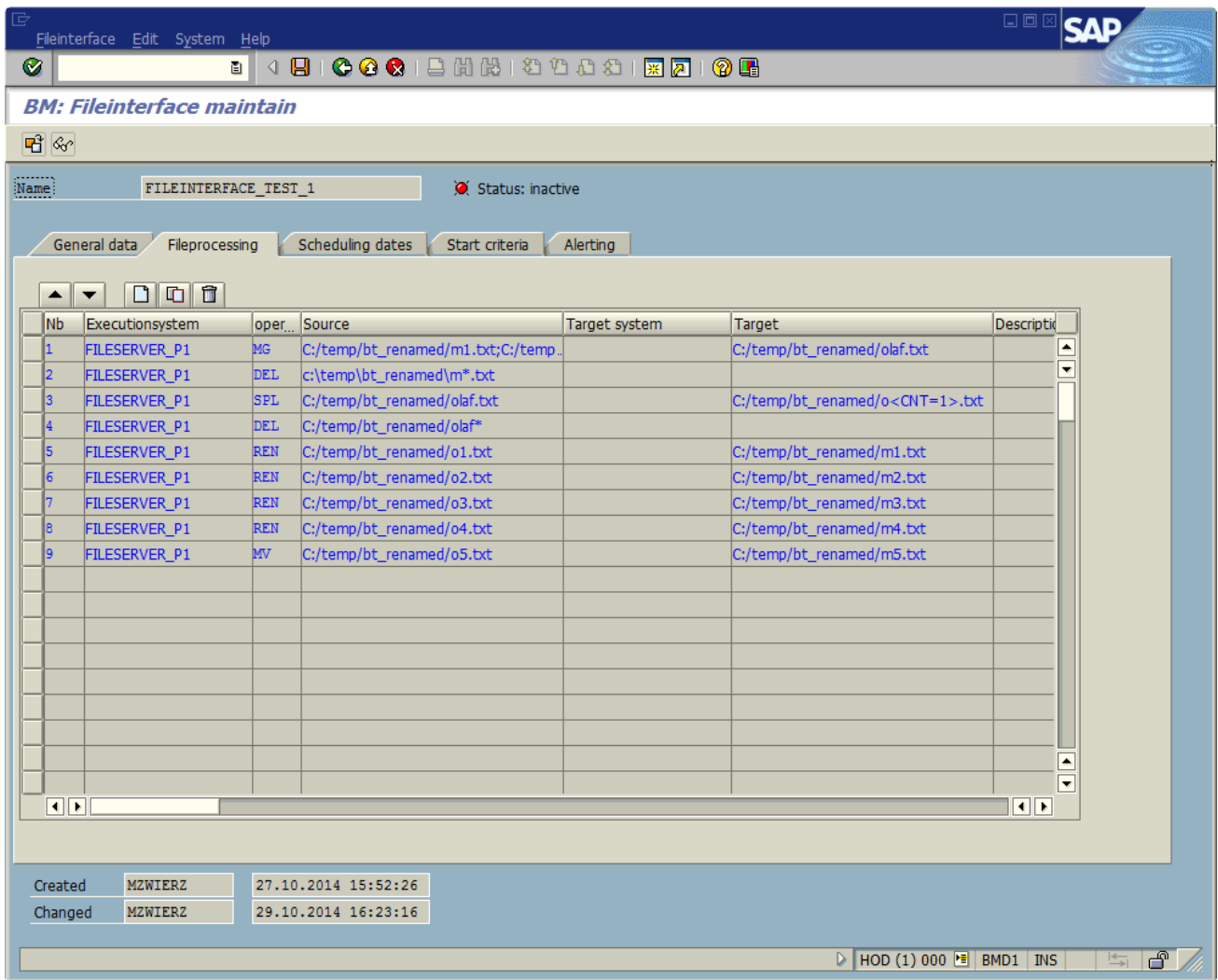
	Change to initial master data maintenance of another file interface
	Copy file interface
	Rename file interface
	Delete file interface
 Release	Release file interface for scheduling
 Lock	Block file interface (cancels the release for scheduling)
 Start manually	Immediately start file interface (ignores scheduling dates and start criteria)
	Provide file interface master data as SAP transport request
	Replicate file interface master data on another SAP System

You can also call these functions using the menu. In addition to the functions mentioned above, you can start the version management of the current file interface in the Edit menu item (see Reporting).

3.2. File processing

The File processing tab displays a table of file operations to be processed by the BatchMan agent in their respective sequence.

File processing mainly defines the execution system, file operation, and the target system (a target system is not needed for each file operation), and source (source file/directory) and target (target file/directory). Depending on the file operation selected, there are other fields, which however are only visible in the detailed view (by double-clicking on the line).



Following buttons are available:



File processing in the order of execution up one position



File processing in the order of execution down one position



Create new file processing



Copy file processing



Delete file processing

To create file processing, choose the Create button. To edit or get a detail view of an existing file processing, choose the related line.

3.2.1. Storage of variables in a separate overview tab

For better visibility of important variables, they can be displayed and edited in an extra job tab “FDM Variables”.

FDM variables have the form <var=name/value>

The screenshot shows a software interface with a table at the top and a text area below it. The table has three columns: 'Name', 'Wert', and 'Typ'. The first row contains 'app', 'job', and 'PARAM'. Below the table is a section titled 'Rest Content' with a 'Media Type' dropdown set to 'application/json' and a 'Pretty Printer' button. The text area contains a JSON object:

```
{ "method": "startJob", "object": { "jobname": "<var=jobname/test_job>", "active": true } }
```

. At the bottom, there is a status bar showing 'Ze 1, Sp 1' and 'Ze 1 - Ze 2 von 2 Zeilen'.

Name	Wert	Typ
app	job	PARAM

Rest Content

Media Type: application/json Pretty Printer

```
{ "method": "startJob", "object": { "jobname": "<var=jobname/test_job>", "active": true } }
```

Ze 1, Sp 1 Ze 1 - Ze 2 von 2 Zeilen

If such a variable is recognized, it is moved to the “FDM Variables” tab and can be easily displayed and edited there without having to laboriously search for and edit it in the content text.

The screenshot shows a software interface with a 'Job name' field containing 'TESTEWS' and a 'Status' field with a red circle icon. Below these are tabs: 'General data', 'FDM Variables', 'Step list', 'Alerting', and 'Categories'. The 'FDM Variables' tab is active, showing a table with two columns: 'Variable name' and 'Value'. The first row contains 'jobname' and 'TEST_JOB_BM'.

Job name: TESTEWS Status: [Red Circle Icon]

General data FDM Variables Step list Alerting Categories

Variable name	Value
jobname	TEST_JOB_BM

3.2.1. Detail view/maintain file processing

BM: FileInterface - Filehandling (maintenance)

Name: FILEINTERFACE_TEST_1

Description:

Executionsystem: [dropdown]

Operation: [dropdown]

Target system: [dropdown]

status independent: [checkbox]

further options

Convert case: [dropdown]

☐ overwrite existing file

☐ Update modification date

Selection Sourcefile: Modification date (Difference to current time)

Diff.to curr.time(min): [field] to: [field]

Changed on: 27.01.2015 11:36:29 to: 27.01.2015 11:36:29

Source: [field]

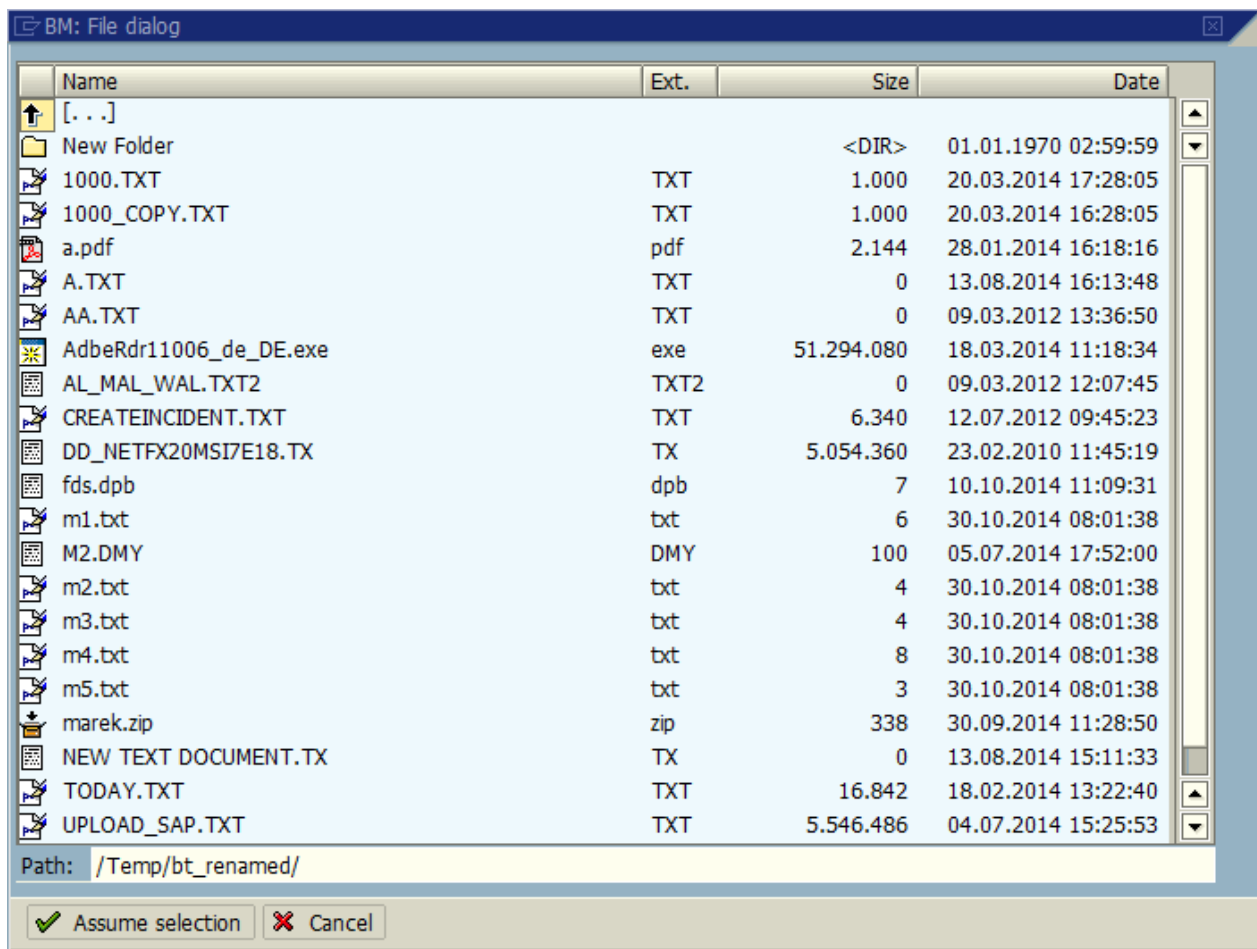
Target: [field]

Continue Cancel [help icon]

The individual fields have the following significance:

Name:	Name of the file interface that summarizes individual file processing steps, which are defined here.
Description:	Description for the individual file processing step
Execution system:	System on which the file operation is started and the system where the source is (see below)
Operation:	File operation (Copy, Rename, Move, Zip, and so on.)
Target system:	Target system when copying and moving
Tolerate faulty status:	Select between step level, job level, and step + job level
Step level:	If a step aborts, the system confirms this abort automatically (aborted/conf.). All subsequent steps are processed. At the end, the job remains in the Aborted status (subsequent job does not start).

Job level:	If a step aborts, this remains in the Aborted status, all subsequent steps are thus not processed. At the end, the job receives the Aborted/conf. status.
Step and job level:	If a step aborts, the system confirms this abort automatically (aborted/conf.). All subsequent steps are processed. At the end, the job gets the Aborted/conf. status (subsequent job starts).
Difference to current time (min):	Additional selection option that further filters the files, which are defined under the source, according to date of the last file change. The values for the date selection from-to are specified in minutes and then calculated depending on (difference) the current system time.
Abort if no source files are found:	This option is active by default and results in a job abort if the system finds no files with the directory/file names defined in the source or if no files meet the selected change date.
Update change date:	This option sets the change date of the target file to the current system time. If this option is not set, the system sets the change date of the target file to the change date of the source file. (Caution: Not all target systems support setting the change date of the target file to the change date of the source file.)
Uppercase/lowercase:	Change uppercase/lowercase for the file name or file name extension of the relevant target file.
Overwrite existing file:	Overwrites an existing target file, otherwise the system issues the error message "Target file already exists".
Source:	Source file or directory for the related file operation, the source must be on the execution system.
Target:	Target file or directory for the related file operation, not every file operation needs a target file. After execution of the file interface, the target file is on the target system if this is defined (for instance, copy) otherwise it is on the execution system (for instance, unzip).
If the execution system or target system is entered, the source and target have a file dialog on the related system for input help (F4).	



3.2.2. Dynamic file names

Using variables, source and target can be calculated dynamically to the execution time. Enter variables in angle brackets <...> as part of the source file or target file.

For instance: /tmp/out_folder/<T=VARI1>08<DATE>.txt

Following variables are possible:

<OPSYS>	Operating system
<INSTANCE>	Instance of R/3 application
<SYSID>	Name of the R/3 application as per system field SY- SYSID.
<DBSYS>	Database system as per system field SY- DBSYS
<SAPRL>	R/3 Release as per system field SY- SAPRL
<HOST>	Computer name as per system field SY- HOST
<CLIENT>	Client as per system field SY- MANDT
<LANGUAGE>	Logon language as per system field SY- LANGU
<DATE>	Date as per system field SY-DATUM
<YEAR>	Year as per system field SY-DATUM, with four places
<SYEAR>	Year as per system field SY-DATUM, with two places
<MONTH>	Month as per system field SY-DATUM
<DAY>	Day as per system field SY-DATUM
<WEEKDAY>	Weekday as per system field SY- FDAYW
<TIME>	Time of day as per system field SY- UZEIT

<STIME>	Hour and minute as per system field SY- UZEIT
<HOUR>	Hour as per system field SY- UZEIT
<MINUTE>	Minute as per system field SY- UZEIT
<SECOND>	Second as per system field SY- UZEIT
<P=name>	Value of a profile parameter of the running system
<V=name>	Value of a variable as per variable table (maintain with TCODE FILE)
<F=name>	Return value of a function module
<L= logical file>	Defines a logical file name (transaction FILE)
<A=logical path>	Defines a logical path (transaction FILE)
<T=name>	TVARV variable

All time-sensitive variables (<TIME> and so on) are calculated at the start of the file interface concerned and thus remain the same within the various file processing steps.

If you use the TVARV variables from BatchMan TVARV rules (cf. define TVARV BM4_UserHB), you have the option to have the system automatically update the corresponding TVARV rules in the file processing overview before the start of the file interface concerned – for this, select the “Auto-update TVARV rules” checkbox.

The screenshot shows the SAP Fileinterface maintain (BM) screen. The title bar indicates 'BM: Fileinterface maintain'. The main window has tabs for 'General data', 'Fileprocessing', 'Scheduling dates', 'Start criteria', and 'Alerting'. The 'Fileprocessing' tab is active. A table lists file processing rules with columns: Nb, Execution system, Operation, Source, Target system, Target, and Inactive. The fourth rule (Nb 4) is highlighted with a red box, showing the source 'C:/temp/bt_renamed/olaf<T=TESTV>*' and target 'C:/temp/bt_renamed/olaf.txt'. A red box also highlights the 'Auto-Update TVARV' checkbox, which is currently unchecked. The status bar at the bottom shows 'HOD (1) 000 BMD1 INS'.

Nb	Execution system	Operation	Source	Target system	Target	Inactive
1	FILESERVER_P1	MG	C:/temp/bt_renamed/m1.txt;C:/temp/bt_r...		C:/temp/bt_renamed/olaf.txt	
2	FILESERVER_P1	DEL	c:/temp/bt_renamed/m*.txt			
3	FILESERVER_P1	SPL	C:/temp/bt_renamed/olaf.txt		C:/temp/bt_renamed/o<CNT=1>.txt	
4	FILESERVER_P1	DEL	C:/temp/bt_renamed/olaf<T=TESTV>*			
5	FILESERVER_P1	REN	C:/temp/bt_renamed/o1.txt		C:/temp/bt_renamed/m1.txt	
6	FILESERVER_P1	REN	C:/temp/bt_renamed/o2.txt		C:/temp/bt_renamed/m2.txt	
7	FILESERVER_P1	REN	C:/temp/bt_renamed/o3.txt		C:/temp/bt_renamed/m3.txt	
8	FILESERVER_P1	REN	C:/temp/bt_renamed/o4.txt		C:/temp/bt_renamed/m4.txt	
9	FILESERVER_P1	MV	C:/temp/bt_renamed/o5.txt		C:/temp/bt_renamed/m5.txt	

Created: MZWIERZ 27.10.2014 15:52:26
 Changed: MZWIERZ 30.10.2014 10:49:40

HOD (1) 000 BMD1 INS

In addition to the variables mentioned above, there are two more file-interface specific variants:

<TF,Offset,Length> If the file interface is started using a file trigger, this variable refers to the file name of the trigger file found (with resolved wildcards). With the Offset and Length parameters, you can resolve individual parts of the trigger file/file name. Be aware that when manually starting the file interface, no trigger file resolution can occur because the start is done without a file trigger. Numerical values are permitted under offset and length, also the variable L can be used, which stands for the actual length of the trigger file name. It is also allowed to subtract from the variable L (for instance <TF,L-3,3><TF,0,L><TF,1,L-5>). If the defined length exceeds the actual length of the trigger file name, then the defined length is set internally to the actual length.

For instance:

Trigger file	= /temp/today_3010.txt
Source	= /tar/<TF,6,4>_<TF,0,5>.<TF,L-3,3>
Source; variable resolved	= /tar/3010_today.txt

<SF,Offset,Length> In contrast to <TF.. , <SF.. does not refer to the trigger file, but to the respective source file. This mainly makes sense if the individual source file is only known at the execution time (because selection is done in the source with wildcards, for instance). With the Offset and Length parameters, you can resolve individual parts of the source file name. Numerical values are permitted under offset and length, also the variable L can be used, which stands for the actual length of the source file name. It is also allowed to subtract from the variable L (for instance <SF,L-3,3><SF,0,L><SF,1,L-5>). If the defined length exceeds the actual length of the source file name, then the defined length is set internally to the actual length.

For instance:

Source	= /source/test*.txt
Source; resolved	= /source/test_0815.txt
Target	= /target/<SF,5,4>_<SF,0,4>.txt
Source; variable resolved	= /target/0815_test.txt

<TF=trgf,Offset,Length> The <TF,Offset,Length> variable described above relates to the trigger file found and registered by the scheduler. Before each execution <TF,Offset,Length> is replaced with exactly the 1 trigger file name found. The advantage here is that the trigger file is registered by the scheduler and does not trigger again afterwards. The disadvantage is that with this type of file recognition and registration, each trigger file only triggers a single processing and thus possibly is too slow when it comes to processing multiple trigger files promptly. <TF=trgf,Offset,Length> variables offer faster processing here. These variables are resolved by the respective file operation. There is no registration here and all trigger files corresponding to the respective pattern are resolved (multi-trigger). This variable can be used like a "normal" wildcard in all source file names.

For instance:

Source	= /source/<TF=*.end,0,L-4>*.txt
Directory /source/	1.end 1.txt 2.end 2.txt 2_copy.txt 3.txt 4.end
Source; <TF= resolved	= /source/1*.txt /source/2*.txt
Source; completely resolved	= /source/1.txt /source/2.txt /source/2_copy.txt

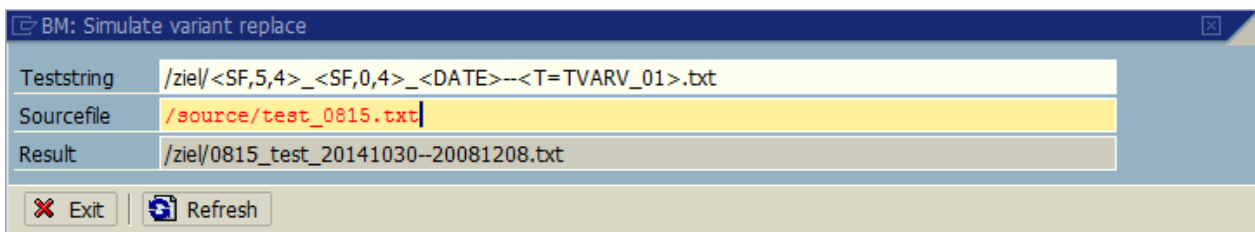
REGEX(regex,input) With the term *REGEX(regex, replcmt)* you can shorten the content of *input* to regular expression matches of the regex *regex* at runtime, so to speak: *input.find(regex)*
Variables are also possible for input. A typical application example here is when the L-x notation is not sufficient because, for example, the file extension is 3 or 4 characters long. Without being able to predict x exactly, you can use REGEX here, which is based, for example, on the "." Point of the file extension.

For instance:

Directory /source/	= /source/test.xls /source/test.xlsx
Source:	= /source/ REGEX(^.[^.]*,<SF,0,L>)---REGEX([^.]+\$,<SF,0,L>)
Source; completely resolved	= /source/test---xls /source/test---xlsx

Where *^[^.]** is the regex for everything before the "." Point and *[^.]+\$* the regex for everything that matches after the "." period.

You have the option to simulate the variable replacement. To do this, choose the field containing the variables (source or target) and then choose the  button.



In the Test string field, enter the file with the desired variables. If you choose the Update button (or Enter), the system simulates the variable replacement in the Results field.

If you use the variables <SF... or <TF... the system also displays the Source file and Trigger file fields. In these fields, enter the file names expected to simulate the replacement.

3.2.3. Copy

The Copy file operation permits the copying of files and directories.

The system always copies the source from execution system to the target in the target system.

The Copy command supports the following combinations of source and target:

<u>Source</u>	->	<u>Target</u>
Directory	->	Directory
Directory/file name	->	Directory/file name
Directory/file name	->	Directory
Directory/...Wildcard...	->	Directory
Directory	->	Directory/...<SF,Offset,Length>...
Directory/file name	->	Directory/...<SF,Offset,Length>...
Directory/...Wildcard...	->	Directory/...<SF,Offset,Length>...

3.3.2.1. Copy http(s)

For local execution systems (execution system = file server), you can also enter files under source that can be reached via http- or https protocol.

The directory path of the source file must be fully specified, wildcards are not supported by the http protocol.

BM: FileInterface - Filehandling (maintenance)

Name: COPY_HTTP
Description: Download

Executionsystem: FILESERVER_P1
Operation: COPY Copy
Target system: FTP_10.3.21.244
status independent

further options

Convert case
☐ overwrite existing file
☐ Update modification date

Selection Sourcefile: Modification date (Difference to current time)
Diff.to curr.time(min) to: Changed on 28.04.2015 15:55:38 to: 28.04.2015 15:55:38
☒ Abort, if no source files found

Source: http://help.sap.com/download/documentation/additional/getstart/470/Getting_Started_470_DE.pdf
Target: /temp/doku/

Continue Cancel

3.2.4. Rename

For the Rename file operations, the system moves the source from the execution system to the target in the delivery system.

The following combinations of source and target are supported:

<u>Source</u>	->	Target
<i>Directory</i>	->	<i>Directory</i>
<i>Directory/file name</i>	->	<i>Directory/file name</i>
<i>Directory/file name</i>	->	<i>Directory</i>
<i>Directory/...Wildcard...</i>	->	<i>Directory</i>
<i>Directory</i>	->	<i>Directory/...<SF,Offset,Length>...</i>
<i>Directory/file name</i>	->	<i>Directory/...<SF,Offset,Length>...</i>
<i>Directory/...Wildcard...</i>	->	<i>Directory/...<SF,Offset,Length>...</i>

3.2.5. Move

For the Move file operations, the system moves the source from the execution system to the target in the target system. If the execution system and the target system are identical, then the system makes a “real” move. The respective files are not moved byte for byte, instead the storage location of the file is changed in the global file table. The speed of such an operation depends on the size of the file.

If the execution system and target system are different, then the move is done internally as copy+delete sequence. The move is thus not basic, so it is theoretically possible that, for instance, copying works but deleting does not (such as a missing write authorization).

The following combinations of source and target are supported:

<u>Source</u>	->	Target
<i>Directory</i>	->	<i>Directory</i>
<i>Directory/file name</i>	->	<i>Directory/file name</i>
<i>Directory/file name</i>	->	<i>Directory</i>
<i>Directory/...Wildcard...</i>	->	<i>Directory</i>
<i>Directory</i>	->	<i>Directory/...<SF,Offset,Length>...</i>
<i>Directory/file name</i>	->	<i>Directory/...<SF,Offset,Length>...</i>
<i>Directory/...Wildcard...</i>	->	<i>Directory/...<SF,Offset,Length>...</i>

3.2.6. Delete

For the Delete file operation, the source is deleted from the execution system, this operation is usually irreversible.

You can enter the following values under source and thus delete:

Source
Directory
Directory/file name
Directory/...Wildcard...

3.2.7. Create directory

The Create dir file operation generates a new directory on the execution system. For this, you have to enter under source the complete path for the directory to be created.

This operation aborts processing if it is not possible to create the directory entered. If this directory already exists, there is no abort and the operation is considered successfully completed.

3.2.8. Create File

The file operation "create file" creates a new empty file (e.g. a trigger file) on the execution system. The complete path for the file to be created must be entered under "Source".

This operation aborts when executed, if it is not possible to create the registered file. If the file already exists, the file will be overwritten - if the "File overwritten" flag is activated, otherwise a termination occurs.

3.2.9. Zip

The Zip file operation generates a zip or a gz file (special case gz) or tar file (special case tar) from the files entered under source.

In the Password field, you can password-protect the zip archive to be generated.

The AES-256 option offers an optional and secure encryption of the password-protected file, but, when unpacking, it needs an archiving program that supports this encryption (Winzip, 7zip, Winrar, and so on).

Under source and target, you can enter the following values:

<u>Source</u>	->	Target
<i>Directory</i>	->	<i>Directory/filename.zip</i>
<i>Directory/file name</i>	->	<i>Directory/filename.zip</i>
<i>Directory/...Wildcard...</i>	->	<i>Directory/filename.zip</i>

Special case gz

Instead of a zip archive, you can also have a gz compressed file made.

Under source and target, you can enter the following values:

<u>Source</u>	->	Target
<i>Directory/file name</i>	->	<i>Directory/filename.gz</i>
<i>Directory/...Wildcard...</i>	->	<i>Directory/*.gz</i>

If you use wildcards for the selection of the source, the system makes a compressed .gz file for each file found. The target file name is formed here always dynamically (target file name = source file name + .gz). To trigger the gz operation, you have to give a target file name that ends with .gz (dummy name such as: *.gz).

Special case tar

Instead of a zip archive, you can also have a tar compressed file made.

Under source and target, you can enter the following values:

<u>Source</u>	->	Target
<i>Directory</i>	->	<i>Directory/filename.tar</i>
<i>Directory/file name</i>	->	<i>Directory/filename.tar</i>
<i>Directory/...Wildcard...</i>	->	<i>Directory/filename.tar</i>

3.2.10. Unzip

With the Unzip file operation, you can unpack zip or gz or tar files.

<u>Source</u>	->	Target
<i>Directory</i>	->	<i>Directory</i>
<i>Directory/filename.zip</i>	->	<i>Directory</i>
<i>Directory/filename.gz</i>	->	<i>Directory/file name</i>
<i>Directory/filename.gz</i>	->	<i>Directory</i>
<i>Directory/...Wildcard...</i>	->	<i>Directory</i>
<i>Directory/filename.tar</i>	->	<i>Directory</i>

3.2.10. Split

With the Split file operation, you can divide a file into several files. The division is done either using the file size, with a certain search key or also page-by-page for PDF files.

As target, specify all target file names with path. Alternatively, you can use count variable <CNT=startvariable>.

In the following example, the file C:/temp/bt_renamed/olaf.txt is split after each blank (RegEx). The number of target files created depends on how many blanks are in the source file. The target files are then called C:/temp/bt_renamed/o1.txt to C:/temp/bt_renamed/oXXX.txt.

The screenshot shows the 'FileInterface - Filehandling (maintenance)' window. It contains several fields and sections for configuring a file operation:

- Name:** FILEINTERFACE_TEST_1
- Description:** (empty)
- Executionsystem:** FILESERVER_P1
- Operation:** SPL Split File
- status independent:** S Steplevel
- further options:**
 - Split by (size/page):** (empty)
 - Split by regex:** (?<=).
 - ☐ overwrite existing file
- Source:** C:/temp/bt_renamed/olaf.txt
- Target:** C:/temp/bt_renamed/o<CNT=1>.txt

At the bottom, there are buttons for 'Continue', 'Cancel', and a help icon.

3.2.12. Merge

The Merge file operation combines all the files specified under source in succession, namely in the target file binarily. If the source files are PDF documents, then they are assembled so that a valid PDF total document is created. For source files with sequential numbers, you can use the count variable <CNT=initial values>.

3.2.13. Convert

The Convert file operation offers an Office interface to convert files.

Prerequisite is an installed LibreOffice (libreoffice.org) or OpenOffice (openoffice.org), where LibreOffice currently offers a wider range of conversion options.

The Office interface offers a number of conversion options (depending on the Office installed), such as: doc->txt, xls->csv, doc->pdf, pdf->txt and so on.

The screenshot shows a software interface for configuring a file operation. The window title is 'BM: FileInterface - Filehandling (maintenance)'. It contains several sections:

- Name:** CONVERT_JOB
- Description:** (empty field)
- Executionsystem:** AGENT_WIN_10.3.21.244
- Operation:** CNVRT Convert
- status independent:** (checkbox, currently unchecked)
- further options:**
 - ☒ overwrite existing file
- Selection Sourcefile: Modification date (Difference to current time):**
 - Diff.to curr.time(min): (empty field) to: (empty field)
 - Changed on: 28.04.2015 17:17:01 to: 28.04.2015 17:17:01
 - ☒ Abort, if no source files found
- Source:** c:/temp/
- Target:** c:/temp/*.pdf
- Buttons:** Continue (green checkmark), Cancel (red X), and a help icon.

Under source and target, you can enter the following values:

Source	->	Target
Directory/file name	->	Directory/file name.[Desired format ending]
Directory/...Wildcard...	->	Directory/file name.[Desired format ending]
Directory	->	Directory/file name.[Desired format ending]

If you install Office in a non-standard directory, the BM agent may not find it. For the corresponding error message, enter the entire directory path of the Office installation into the file
.../batchman_agent/bin/btcmman/profiles/officehome.txt

3.2.14. Execute Command

The cmd operation offers the possibility of starting and controlling scripts or programs via the respective operating system shell in a time-controlled or process-dependent manner. SSH target systems and target systems with an installed BM agent are permitted as target systems for execution.

CMD for launching executable files

Enter the name (possibly with path information) of the executable file in the cmd Command field.
e.g.: c:\script\run.bat

If the executable file requires certain parameters, you can also pass them here.
e.g.: c:\script\run.bat 13051978

You have the option of using dynamic variables (see 3.2.2).
e.g.: c:\script\<T=SCRIPTNAME> <DATE>

Under Windows operating systems, note that shell commands such as "dir" are not executable files. To start shell commands, you must also specify the shell as an executable file.
e.g.: cmd /c dir

CMD for starting script content

For Windows and Linux operating systems you have the option of defining script contents directly in the cmd command field.

Note: To distinguish script content from the call for executable files, script content must be multiline (if necessary, you could also insert a comment line).

Under Linux, cmd content is copied into a sh script and then executed.

Under Windows, this content is copied into a bat script and then executed.

In addition, under Windows you have the option of declaring the script as vbs (@vbs tag) or as powershell (@ps tag). To do this, enter either @vbs or @ps as the first line, without @Tag the scripts run as a bat script

BM: FileInterface - Filehandling (maintenance)

Name: TEST_LOCAL

Description: bat Script

Executionsystem: SFTP_LOCAL

Operation: Execute CMD

status independent

CMD command / script

```
@echo off
set loopcount=5
:loop
echo Hello World!
set /a loopcount=loopcount-1
if %loopcount%==0 goto exitloop
goto loop
```

Continue Cancel

BM: FileInterface - Filehandling (maintenance)

Name: TEST_LOCAL

Description: powershell Script

Executionsystem: SFTP_LOCAL

Operation: Execute CMD

status independent

CMD command / script

```
@ps
$explorerprocesses = @(Get-WmiObject -Query "Select * FROM Win32_Process WHERE Name='explorer.exe'" -ErrorAction SilentlyContinue)
If ($explorerprocesses.Count -eq 0){ "No explorer process found / Nobody interactively logged on" }
Else { ForEach ($i in $explorerprocesses) {
$Username = $i.GetOwner().User
$Domain = $i.GetOwner().Domain
Write-Host "$Domain\$Username logged on since: $($i.ConvertToDateTime($i.CreationDate))"
```

Continue Cancel

3.2.15. Encrypt and decrypt files with PGP

The serious cryptography in the import and export is subject to international restrictions. Due to these restrictions in the standard installation of Java (and thus also in the standard installation of BatchMan FDM), only key lengths are supported that can in general be considered "broken".

For countries where cryptography is not subject to legal restrictions, Oracle offers, as the manufacturer of Java, Policy Files (Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files), which cancel the cryptographic restrictions mentioned.

To be able to use serious cryptography in BatchMan FDM, you have to download from the manufacturer page for Oracle the JCE files that fit your Java version and then copy these to the <java-home>\lib\security directory of your Java installation.

You can find the JCE files for the most current Java 8 at this link:

<http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html>

PGP works with key pairs consisting of a public key and private key.

Following keys are needed:

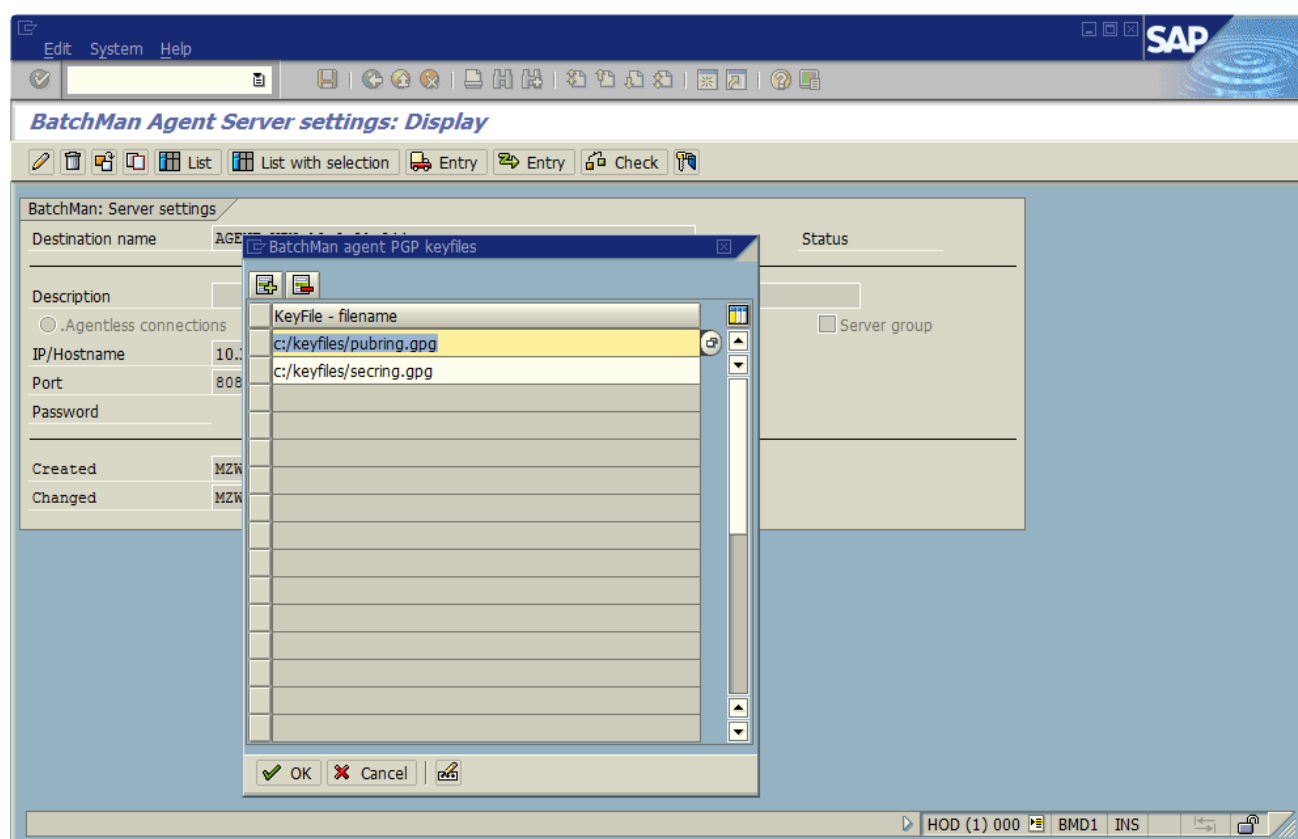
Encryption	- public key of the receiver
Signature	- private key of the sender
Encryption	- private key of the receiver
Signature test	- public key of the sender

Keys are retained in key files, the location of these key files must be made known in the BatchMan server definition of the respective BM agent.

To get to the system set up, choose:

BatchMan® *Function tree* → *Customizing* → *Technical customizing* → *non-SAP server definitions*

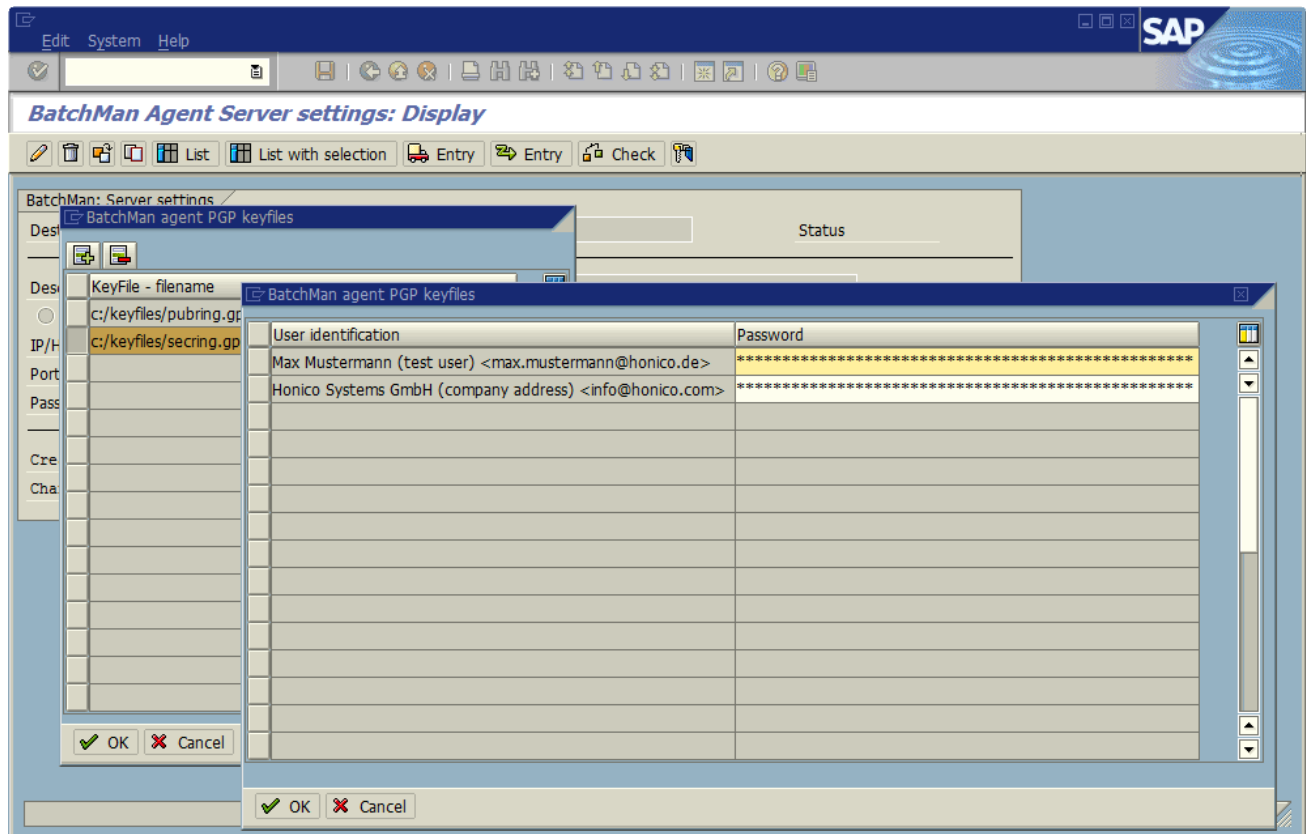
Select the respective BM agent and then choose PGP Keys [CTRL +F8]



Then enter the path for the files with the required public keys and the file with the required private keys into the Key file table.

If individual private keys are password-protected, you also have to make the password known in BatchMan. For this, select the relevant line with the file that has the protected private keys and choose the Password button.

Then the system reads the private keys from the file marked and you can define passwords for the individual keys.



3.2.15.1. Encrypt

The Encrypt file operation offers you the option to use PGP to encrypt files, to sign files or to encrypt and to sign files as a combination.

To encrypt a file, select from the Encrypt selection box a public key from the key files that you defined in the setup.

To sign a file, select from the Signature selection box a private key from the key files that you defined in the setup.

Under source and target, you can enter the following values:

Source	->	Target
Directory/file name	->	Directory/file name.[Desired format ending]
Directory/...Wildcard...	->	Directory/*. [Desired format ending]
Directory/...Wildcard...	->	Directory/...<SF,O,L>.[Desired format ending]

BM: FileInterface - Filehandling (maintenance)

Name: TEST_CIPH

Description:

Executionsystem: AGENT_WIN_10.3.21.244

Operation: CIPH Encrypt

status independent

further options

Encryption:

Signature:

☐ overwrite existing file

Selection Sourcefile: Modification date (Difference to current time)

Diff.to curr.time(min) to: Changed on 17.01.2017 16:53:44 to: 17.01.2017 16:53:44

☒ Abort, if no source files found

Source: c:\Temp*.DAT

Target: c:\temp*.asc

Continue Cancel

3.2.15.2. Decrypt

The Decrypt file operation offers you the option to decrypt files via PGP and validate the sender of the file by checking the contained signature.

To decrypt a file, you only need specifications for source and target.

Under source and target, you can enter the following values:

Source	->	Target
Directory/file name	->	Directory/file name
Directory/file name	->	Directory
Directory/file name	->	Directory/<SF,Offset,Length>
Directory/...Wildcard...	->	Directory
Directory/...Wildcard...	->	Directory/<SF,Offset,Length>

The system selects the correct keys automatically from the key files you defined in the setup. As an additional option, you can abort if the signature is missing.

3.2.16. Synchronization

The Sync file operation offers you the option to synchronize directories among one another.

The synchronization reads the directory list from source and target and compares the files using their last date of change.

Files with newer date of change then overwrite files of the same names with an older date of change (or files that do not exist).

The synchronization of deleted files has priority, however. If the deletion of a file is registered, then the system also deletes this on the opposite side.

The screenshot shows a software window titled "BM: FileInterface - Filehandling (maintenance)". It contains several input fields and checkboxes for configuring a synchronization task. The "Name" field is set to "TEST_SYNC". The "Description" field is empty. The "Executionsystem" is "AGENT_WIN_10.3.21.244", the "Operation" is "SYNC File Sync", and the "Target system" is "HONICO_SFTP_(244)". There is a "status independent" checkbox which is checked. Under the "further options" section, the "duplex synchronization" checkbox is unchecked, and the "synchronize deleted files" checkbox is checked. At the bottom, the "Source" is "d:/test" and the "Target" is "/temp". The window has "Continue", "Cancel", and a help icon at the bottom left.

Name	TEST_SYNC		
Description			
Executionsystem	AGENT_WIN_10.3.21.244		status independent ☒
Operation	SYNC File Sync		
Target system	HONICO_SFTP_(244)		
further options			
☐ duplex synchronization			
☒ synchronize deleted files			
Source	d:/test		
Target	/temp		

Following options are available:

Duplex synchronization

Enables synchronization at both ends Source -> Target and Target -> Source; if this option is deactivated, the synchronization is only done in the direction Source -> Target.

Synchronize deleted files

If you delete a file, then the system also deletes this on the opposite side (the same goes for the Duplex option). If this option is deactivated, a file cannot be deleted in fact because the synchronization would restore this file later with the mapping of the opposite side.

3.2.17. Print

The file operation “print” offers the possibility to send files to a printer. The print layout is not processed, only the native print formats of the respective printer (e.g. postscript, jpg files, etc.) are supported. Please refer to the appropriate printer operating manual for the native print formats supported by your printer.

Under “Source”, select the file you want to print (wildcards are possible). Under “Destination”, specify the appropriate printer, using the F4 value help. In this case, only printers can be selected, which are set up locally in the printer control system in the operating system of the FDM Agent PC.

3.2.18. Mail

With the operation Mail, mails can be sent via server connections of the type SMTP. It is also possible to send files. The mail text can be designed dynamically using BatchMan variables.

BM: FileInterface - Filehandling (maintenance)

Name: MAIL
Description:

Executionsystem: SMTP
Operation: Send Mail
status independent

E-Mail message

to: test@honico.com
Cc:
Subject: Triggerfile

Hello,
attached please find the Triggerfile <TF,0,L>.

Selection Sourcefile: Modification date (Difference to current time)
Diff.to curr.time(min) to: Changed on 09.10.2020 15:46:38 to: 09.10.2020 15:46:38
☒ Abort, if no source files found

Source: /output/<TF,0,L>

Continue Cancel

3.2.19. REST

The REST operation offers the possibility to set up FDM processes that call or use other services via their REST API.

In the Rest Request tab, all input parameters can be specified according to the respective REST API. HEADER, QUERY or PARAM parameters can be defined in the parameter table.

HEADER parameters: Are translated into standard http request header key-value pairs, separated by colons

QUERY parameters: Are standard http url key-value pairs attached to the url

PARAM parameters: Define <VARIABLES> e.g. in the URL of the http connection

A special case of the PARAM parameters is debug: true, which means that the complete http log of the REST call is written out into the job log.
If a REST call ends with an http status greater than 299, the complete http log is also written out in the job log.

BM: FileInterface - Filehandling (maintenance)

Name: REST
Description:

Executionsystem: HTTP
Operation: REST Webservice
status independent:

Rest Request | Rest Response

Method: POST

Parameter

Name	Value	Type
CMD	posts	PARAM
debug	true	PARAM
		PARAM
		HEADER
		QUERY
		PARAM

Rest Content

Media Type: application/json
Pretty Printer

```
{
  "book": {
    "title": "James Bond",
    "author": {
      "name": "Fleming",
      "firstname": "Ian"
    }
  }
}
```

Ln 1, Co 1 | Ln 1 - Ln 9 of 9 lines

Continue Cancel

BatchMan Agent Server settings: Maintain

Destination name: HTTP Status

Description: [Empty field]

☐ .Agentless connections
 ☐ .BatchMan agent
 ☒ .Connections via BM agent

BatchMan Agent: LOCAL

Protocoltype: HTTP protocol

IP/Hostname: https://jsonplaceholder.typicode.com/...

Port: https://jsonplaceholder.typicode.com/<CMD>

User: [Empty field]

Password: [Masked field]

System monitoring and alarm

Alertteam: [Empty field] Alerttool: [Empty field]

Created	MZWIERZ	09.10.2020	09:47:23
Changed	MZWIERZ	09.10.2020	12:05:16

SAP | HE3 (2) 000 | HOSRN083 | INS

The response to the REST call is handled in the Rest Response tab.

Due to the relatively large amount of data, the response to the REST call is not written out in the job log if the http status code is positive.

Via the temporary variable name fields the JSON response can be linked to variables either completely or partially via the maintained xpath.

These variables are written out into the job log

Under xpath you can use different methods to parse out the variables from the response:

- Jsonpath
 - Json Parser, is recognized at the starting „\$“
 - <https://github.com/json-path/JsonPath>
 - Example: \$.runnode.contAttr.contStatus
- Xmlpath
 - XML Parser, is recognized at the starting „/“
 - <https://de.wikipedia.org/wiki/XPath>
 - Example: /html/div[5]
- Response Header complete
 - Output of the complete response header
 - is recognized by the expression “response_header”
 - Example: response_header

- Response Body complete
 - Output of the complete response body
 - is recognized by the expression "response_body"
 - Example: response_body
- regex on response body
 - Output of regex match group n
 - is recognized by the expression „regex.group[n]:“
 - Example:

Xpath	regex.group[1]:filename.*?\=\"([\\w_\\.]*)\\"
response	<Test filename="test.pdf" size="1mb">
var	test.pdf
- regex on response header
 - Output of regex match group n
 - is recognized by the expression „response_header:regex.group[n]:“
 - Example:

Xpath	response_header:regex.group[1]:filename.*?\=\"([\\w_\\.]*)\\"
response	Content-Disposition: attachment; filename="test.zip"
var	test.zip
- Special case: response body is a file
 - Automatic saving of the response body in a temp. file
 - Is recognized by the response header "Content-Disposition: attachment;"
 - In all subsequent steps the complete path of the temp. File is available under the variable <RESTFILE> and can be used to copy the file away. At the end of the job all temp. Files deleted.

Rest Request Rest Response	
Parser Response Object	
temp.variablename	json xpath
json_path	\$.time.test[*]
xml_path	/html/div[5]
response_header	response_header
response_header_regex	response_header:regex.group[1]:Content-Disposition.*\=\"([\\w_\\.]*)\\"
response_body_regex	regex.group[1]:attachment.*\=\"([\\w_\\.]*)\\"

Temp. Variables are valid for all steps within a job - in this way variables can be set and used in different REST steps (Step1 can, for example, generate a sessionID and then pass this on to the following steps).

Variables can be used as input for the rest request of a subsequent step. To do this, they must be specified in angle brackets in the form <variable>.

Rest RequestRest Response

MethodPOST

Parameter

Name	Value	Type
cookie	<cookie-var>	HEADER
authorization	1	HEADER

Rest Content

Media Typeapplication/json

Pretty Printer

```

{
  "method": "login",
  "object": {
    "type": "user",
    "username": "<server_user>",
    "loginpass": "<server_pass>"
  }
}

```

Ln 1, Co 1

Ln 1 - Ln 8 of 8 lines

In addition to the variables that were created in previous steps, there are the additional variables <server_user> and <server_pass> which contain the username and password from the server definition of the used connection.

If user name/password is defined in the server definition, basic authorization is always attempted during a connection by adding the line

Authorization: Basic BaSe64

in the http header. If this is not desired, the basic authorization can be switched off using the HEADER parameter “authorization” with the value “0”.

If the user/password variables <server_user> or <server_pass> are used in a request, the basic authorization is automatically deactivated for this step. If this is not desired, you can set it using the HEADER parameter “authorization” with the value “1”.

With the help of the Periodic (execute until) fields, a REST call can be repeated until a temp. Variable has reached a certain value, the interval is fixed at 1 minute.

With the help of the Abort (if ..) fields, the REST call can generate an abort; this abort would also end any periodicity.

Rest Request
Rest Response

Parser Response Object

temp.variablename	json xpath
message	\$.message
status	\$.status
jobstatus	\$.runnode.contAttr.contStatus

Periodic (execute until...)

temp.variablename		Value
jobstatus	=	finished

Abort (if...)

temp.variablename		Value
status	<>	0
jobstatus	=	aborted

job log

Date	Time	
09.10.2020	15:26:25	Step: 1
09.10.2020	15:26:35	Set variable full_output to {"book":{"title":"James Bond","author":{"name":"Fleming","firstname":"Ian"},"id":101}}
09.10.2020	15:26:35	Set variable author_name to Fleming
09.10.2020	15:26:35	Fleming = Fleming
09.10.2020	15:26:35	Exit-Code: 0

For better visibility of important variables, they can be displayed and edited in an extra job tab “FDM Variables”.

FDM variables have the form <var=name/value>

Name	Wert	Typ
app	job	PARAM

Rest Content
Media Type application/json
Pretty Printer

```
{ "method": "startJob", "object": { "jobname": "<var=jobname/test_job>", "active": true } }
```

Ze 1, Sp 1
Ze 1 - Ze 2 von 2 Zeilen

If such a variable is recognized, it is moved to the “FDM Variables” tab and can be easily displayed and edited there without having to laboriously search for and edit it in the content text.

Job name TESTEWS
Status 🔴 🟢 📄

General data
FDM Variables
Step list
Alerting
Categories

Variable name	Value
jobname	TEST_JOB_BM

3.2.20. Additional file operations

You can maintain additional file operations relatively simply (such as file checks, transformations, and so on). We will gladly make you an offer with no obligation.

3.3. Scheduling dates

You can maintain periodic scheduling using the Scheduling dates tab. The scheduling dates define the time the system monitors the start criteria of a file interface and thus lead to a potential start of the related file interface.

SAP calendar

Using the specifications of a factory calendar, the system can recognize working days and holidays. The factory calendar enables you to make start restrictions specific to countries and factories. The calendar is a required entry.

You can go to the maintenance of factory calendars using these methods:

Transaction SCAL

The SAP calendar button for the maintenance of the periodicity (internal call of transaction SCAL)

Function tree [BatchMan®](#) → [Toolbox](#) → [Calendar tools](#) → [SAP calendar maintenance \(SCAL\)](#)

Valid from – to

The validity period determines from when or to when the system takes a file interface into account during scheduling.

Expected value

In the Expected value field, you can define how often the start criteria trigger is expected within the respective scheduling (such as how often a file comes on a certain day (start file)).

You can later place alarms on these expected values (such as the start file did not arrive in the timeframe) as well as evaluation/monitoring.

Parallel execution

This flag defines whether the system starts the respective file interface immediately, without checking whether or not the related file interface is already active (Parallel execution = X) within an active scheduling when triggering the related start criteria – or whether the system only starts the file

interface once the predecessor is no longer active. If this flag is not set, the system will not start the file interface as long as the predecessor is still active or was aborted without acknowledgement.

Valid from – to

Specify the timeframe (time of day) from when to when a file interface is active with completed scheduling (scheduling day). Only within this timeframe will the system check the start criteria. A file interface can also remain active beyond a change of day even if no scheduling is defined for this following day.

Here it is also possible that several schedulings are active at the same time for a file interface (a current one and one from the day before). If a file interface start occurs here, the system always calculates this for the expected value of the oldest active scheduling. Also in monitoring, the system lists the scheduling under the oldest active scheduling.

In the event the expected value is not attained, the system always issues an alarm upon expiration of the validity period.

Periodicity of scheduling

For the periodicity of scheduling, you have several options:

MONTHLY EXCEPT JA, FB, MR. . . .

If you select the "Monthly" field, the system executes the job monthly. You can also make a more finely tuned selection by marking one of the four fields below, such as Calendar day, Working day, Month-end or the Last calendar day of the monthly execution date for the job.

If you select the Ja, Fb, Mr ... fields, then the system will NOT execute the job in the months selected. These fields are valid for all the following four options:

1. If you enter "Calendar day X" and the corresponding day, the system will carry out this the job monthly according to the day entered. With these restrictions, you can further impact the execution.
2. Working day: X

The "Working day: X" checkbox means that the job is executed each month on the nth working day.

Example: June 1996

01. Working day = June 3, 1996	Scheduling on ⇒ June 3, 1996
20. Working day = June 28, 1996	Scheduling on ⇒ June 28, 1996

3. Month-end – Move up by X days

Under "Month-end" the system understands the last working day in the month. If you select the "Month-end – Move up by X days" field, the system always executes the job on the last working day in the month or moved up by X calendar days before the month-end. If you also select the "Working days" field, then the previous entry "Move up by X days" only refers to working days.

Example: June 1996

☒ Month-end	⇒ Scheduling on June 28, 1996 (= last working day in June 1996)
☒ Month-end - Move up by 7 days	⇒ Scheduling on June 21, 1996
☒ Month-end - Move up by 7 days X Working day	⇒ Scheduling on June 19, 1996

4. Last calendar day – Move up by X days

The system executes the job on the last calendar day or on the day moved up by X days of the month.

DAILY . . . EXCEPT MO, TU, WE

If you select the "Daily" field, the system executes the job each day.

EXCEPTION: If you select fields in the adjacent list of weekdays, then the system does not execute the job on weekdays selected.

PERIODICAL EVERY X DAYS, beginning on:

If you select this field, the system executes the job periodically, according to the number of calendar days entered in the adjacent field. To determine the day of execution, the system uses the reference date (beginning on). Thus, the scheduling date results from the reference date and the periodicity.

Example:

Execute every day	Entry ⇒ '1'
Execute every 2nd day	Entry ⇒ '2'
Execute every 3rd day	Entry ⇒ '3'

Note: The "Periodically every X weeks" field has been omitted. If the system is to execute a job weekly, then simply use "Periodically every X days" and multiply by 7.

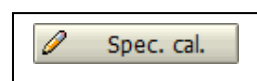
Example:

Execute every week	Entry ⇒ '7'
Execute every 2nd week	Entry ⇒ '14'
Execute every 3rd week	Entry ⇒ '21'

Special calendar/Special rule

The special calendars provide a tool you can use to define separate executions in special calendars based on the SAP factory calendars/ logistics calendars.

The BatchMan user manual describes how you can create and maintain special calendars.



Using the Special rule button, you can go directly to the special calendar maintenance.

If, aside from the periodicity, schedulings are required on other days, then you can define these dates in the "Other schedulings" table.

Note: However, the prerequisite is a selected periodicity that is not equal to "No scheduling".

If a scheduling date was maintained, then the Restrictions button becomes active. This calls another screen (instead of the scheduling dates), where you can make restrictions for the start date.

BM: Fileinterface maintain

Name: FILEINTERFACE_TEST_1 Status: inactive

General data | **Fileprocessing** | Scheduling dates | Start criteria | Alerting

SAP calendar: 01
Valid from: 01.01.2004 To: 31.12.9999 [Scheduling preview]

Expected value: 5
Parallel Execution: ☐
Valid from: 04:00:00 To: 08:10:59 Date + Days ☐ WD

Restrictions (if one restriction is true -> no scheduling) [Scheduling data]

Behavior on non workdays

- ☒ Repeat always (no restrictions)
- ☐ Skip
- ☐ Forward Working day
- ☐ Move on next working day

periodicly restrictions

- ☒ No restrictions
- not on: ☐ CD of month ☐ WD of month
- not on last: ☐ CD of month ☐ WD of month

other restrictions (no scheduling)

From date	To date

Created: MZWIERZ 27.10.2014 15:52:26
Changed: MZWIERZ 30.10.2014 10:49:40

HOD (1) 000 BMD1 INS

This is where you specify whether the system is always to execute a file interface on non-working days, or the system is to omit it, or move it up to a working day or move it up to the next working day.

In periodic restrictions, you can also enter that the file interface should not run on certain days in the month (CD – calendar day; WD – working day).

Example: Processing is to run monthly. Only on the last working day is the system to run an extended processing, which carries out month-end closing. Then “Not on the last WD of the month” is activated.

If other restrictions are required in addition to those chosen, then you can define these in the “Other restrictions” table.

Note: Prerequisite is a selected periodicity that is not equal to “no scheduling” so that the system takes values into consideration.

Using the active Scheduling dates button, you can switch back to the schedulings. With maintained schedulings and restrictions, the Scheduling preview button is also active. You can use this to display a preview of the period values set. You should always execute this function to be safe, especially for special schedulings.

3.4. Start criteria

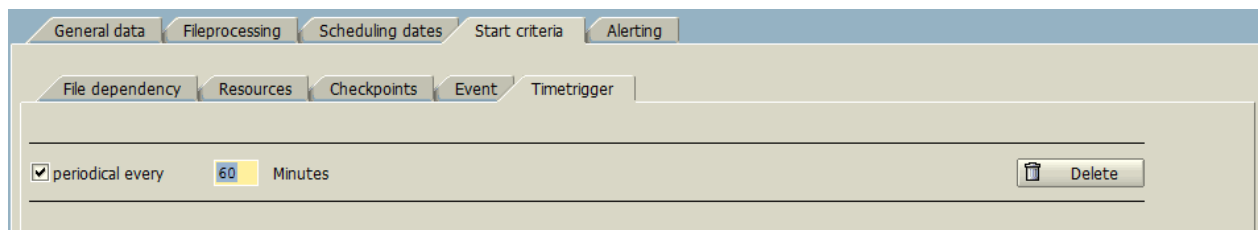
If an active scheduling is available for a file interface, the system will start it as soon as all defined start criteria are met.

There are 5 various start criteria that you can combine as needed:

File, Resource, Checkpoint, Event, and Time

Time trigger

The time trigger defines when and how often the file interface starts, after scheduling is completed (valid from time). If there are several start criteria, time trigger defines when and how often the system check the other start criteria.



For instance: Start criteria: Time-periodic every 60 min.

If the scheduling dates define a scheduling, the file interface starts from the validity time and then every 60 minutes until the end of the validity time.

For instance: Start criteria: Time-periodic every 60 min. + file dependency

If the scheduling dates define a scheduling, the file interface checks whether or not the related start file exists as of the validity time and then every 60 minutes until the end of the validity time. If so, then the system starts the file interface.

File dependency

If a file dependency is defined, the file interface starts only if the defined start file (trigger file) is on the defined system.

If the system finds a start file causing a file interface to start, then the system registers this start file in table /BTCMAN/M_FILFRG. A start file will not trigger any (more) starts if it is already registered in table /BTCMAN/M_FILFRG with the same name, same storage location (destination), and same modification time.

Enter the name of the start file completely with the path. Variables are also allowed (see section: Dynamic file names) and *- Wildcards.

After starting the file interface, the name of start file the system found forms the basis for the variable <TF,...> in the file operations to be executed (see section: Dynamic file names).

A typical problem when using file dependencies can arise if the start file is the same file that you want to process with the file interface.

For instance: You want to send a file A. To ensure the sending process is started once file A available, define a file dependency for file A.

The problem here is that the file trigger immediately starts the file interface as soon as a file is available, and the file trigger cannot tell that this file might not be finished yet. So the system may process a file that is not (yet) complete.

One way you can avoid this issue is to use the Test write lock flag. For this you need, however, write authorization for the relevant system/directory.

If the Test write lock flag is active, the system then tries to open the related start file in the write mode, system assumes that if this works then no other process can hold a write lock on this file and thus the file has been completely written.

Another option is to use the Change date field.

Here you can test when the last file change occurred (assuming: file is older than 5 min. = finished being written). Here you have to be aware of how the respective operating system writes a file, for instance in Microsoft Windows the system sets the change date at the beginning of the file generation (file = 0 bytes) and again at the end of the write process.

So to ensure that the file is no longer being accessed, you would have to define the change date to be before the maximum duration of the write process.

Search file contents

Here the system only triggers the defined file trigger if the search term can be found in the file. This search mode is possible on the related BM agent system because protocols like SFTP do not provide any text search within files.

File >

Here the system triggers the defined file trigger only if the found file is greater than the file size indicated here.

The screenshot shows the 'File dependency' configuration window. It has tabs for 'General data', 'Fileprocessing', 'Scheduling dates', 'Start criteria', and 'Alerting'. Under the 'Fileprocessing' tab, there are sub-tabs for 'File dependency', 'Resources', 'Checkpoints', 'Event', and 'Timetrigger'. The 'File dependency' sub-tab is active. It contains several sections: 1. A checked 'File dependency' checkbox with a text field containing 'c:/temp/bt_renamed/m1.txt' and a 'Destination' field with '10.3.21.75_HTTP'. There are 'Test File lock' and 'Delete' buttons. 2. An unchecked 'Search for file content' checkbox with a 'Search item (regex)' text field and a 'Delete' button. 3. An unchecked 'File > than' checkbox with a 'Filesize (byte)' text field and a 'Delete' button. 4. An unchecked 'Mod.time' checkbox with 'from' and 'to' date/time pickers set to '18.12.2014 12:12:08' and a 'Diff. to current time (min)' text field. There is a 'Delete' button.

Resources

A resource is a symbolic field that job control uses to determine start properties. In contrast to the checkpoint, resources have to be decreased at the start and increased at the end. For load restriction, resources can be used for instance to manage tape drives or also for batch processes. Items 6.6 Checkpoints and 6.7. Resources in the BatchMan user manual have more details on checkpoints and resources.

The screenshot shows the 'Resources' configuration window. It has the same tabs as the previous window. Under the 'Fileprocessing' tab, the 'Resources' sub-tab is active. It contains a checked 'Resource' checkbox with a text field containing 'BREAK' and a 'Delete' button. Below it, there is an 'Amount' text field with the value '1'.

Checkpoints

A checkpoint is a symbolic field that the control uses to determine start properties.

Checkpoints are used mainly to:

- trigger dependencies between various file interfaces
- define certain states in the system (use as milestone with the option for alarms)

Wait for checkpoint before start: You can set parameters for a file interface so that it starts only once a checkpoint has the value that is correct or larger.

A file interface can alter the value of checkpoints.

Set checkpoint BEFORE execution: Before the scheduling or the start of the file interface, a value is reassigned (fixed value) to the checkpoint or the value is increased (negative value lowers the value of a checkpoint).

Set checkpoint AFTER execution: After a file interface was ended and depending on whether or not the file interface was successful, you have the option to reassign a value to checkpoint (fixed value) or increase a value (negative value lowers the value of a checkpoint).

The screenshot shows a software interface with a tabbed menu at the top: 'General data', 'Fileprocessing', 'Scheduling dates', 'Start criteria', and 'Alerting'. The 'Start criteria' tab is active, and within it, the 'Checkpoints' sub-tab is selected. The main area contains three sections, each with a checkbox and a 'Delete' button. The first section, 'Wait for job start at checkpoint value', has a checkbox, a 'Checkpoint' field with a dropdown arrow, and a 'Par.value >=' field. The second section, 'Set checkpoint before job starts', has a checkbox, a 'Checkpoint' field, a 'to value' field, and a 'Fixed value' checkbox. The third section, 'Set checkpoint after job end', has a checkbox, a 'Checkpoint' field, an 'RFC destination' field, and two rows for 'if OK' and 'if abort', each with a 'Fixed value' checkbox.

Event

If an external system cannot communicate with an SAP System, it is possible to trigger an event in a SAP System the using the operating system program SAPEVT. With this start condition, the respective event is specified with its related parameter. The parameter is not required; but you should use it so that an event need not be defined for each incident.

The system reads the event with the parameter from the event history (available with XBP 3.0), for this, the corresponding event must be recorded in the event history before (SM64 -> Criteria Manager). After the start criteria were triggered, the triggering event in the event history is confirmed so that the system does not trigger the start criteria later again.

SAP

Goto System Help

SM64

Event History: Profiles and Criteria

Event History Criteria Reorganization BckProcEvnts

Criteria Profiles

Type	Profile ID	Text	Status	Author	Changed By	Changed On	Changed At
Event History	1	Events	Green	GMEYER	GMEYER	19.09.2013	16:13:36
	2	Test		GMEYER	GMEYER	07.03.2013	17:30:47

Criteria Hierarchy of the Profile Events

AND

OR

BM_TEST_GM

Change

Description	Fld	Option	Lower Limit	Upper Lim.
BM_TEST_GM	EVENTID	=	BM_TEST_GM	
		=	J_SH1_START_OF_JOB	
		=	EVENT_AN4_TEST	
	EVENTPARM	>		

Scripting support is dis

AN4 (1) 000 HOSR0244 INS

SAP

Goto System Help

SM64

Event History: Overview

Event History Criteria Reorganization BckProcEvnts

Event History: 2 Entries

Current Date	Time	Event	Event par.	Number	Status	Status	Background Server
08.01.2015	10:54:17	BM_TEST_GM	MAREK	0	Confirmed	OK	HOSR0244_AN4_00
08.01.2015	10:56:29	BM_TEST_GM	MAREK	0	New	OK	HOSR0244_AN4_00

AN4 (1) 000 HOSR0244 INS

By maintaining an RFC destination, you can trigger events that were triggered on other SAP Systems.

Trigger event after execution: This function is for starting subsequent processes after the file interface has ended. Here, you can trigger various event parameters depending on the status after the file interface has ended.

The screenshot shows the 'Alerting' tab in the SAP BatchMan configuration interface. It contains two main sections for defining alert events.

Event (Historie)

- ☐ Event (Historie)
- Parameter: [Text Field]
- RFC-Destination: [Text Field]
- ☐ alle früheren Events bestätigen
- [Löschen]

Auslösen Event NACH Ausführung

- ☐ Auslösen Event NACH Ausführung
- Event: [Text Field]
- RFC-Destination: [Text Field]
- Parameter:
 - wenn OK: [Text Field]
 - bei Abbruch: [Text Field]
- [Löschen]

3.5. Alerts

The alert can be triggered for various system states. If you enter a **BatchMan** team and alert tool, the system sends messages for the jobs.

The teams are maintained via [BatchMan® Function tree → Customizing → Basic customizing → Team/Recipient definitions](#).

The documentation **BM4_d_AlertHB_JJMM.doc** describes the Customizing of the various alert types (alert tools) using examples.

The screenshot shows the 'Alerting' tab in the 'Fileinterface maintain' window. The window title is 'BM: Fileinterface maintain'. The name of the interface is 'FILEINTERFACE_TEST_1' and its status is 'inactive'.

Alerting Tab

	BatchMan team	Table	Alert tool
Finished processing	[Text Field]	☐	[Text Field]
Processing stopped	[Text Field]	☐	[Text Field]
Expected value filetrigger below	[Text Field]	☐	[Text Field]
Expected value filetrigger exceeded	[Text Field]	☐	[Text Field]
Max. runtime (min)	[Text Field]	☐	[Text Field]
☐ Abort if run time exceeded			
☐ Retrigger alert			
To (min)	[Text Field]	☐	[Text Field]

Metadata

Created	MZWIERZ	27.10.2014 15:52:26
Changed	MZWIERZ	07.11.2014 14:59:42

Status Bar

HOD (1) 000 BMD1 INS

Job has ended (positive alert): This generally serves to notify individual users or user groups that processing was a success.

However, if certain processing is to be ended by a specific time, a positive alert may also be useful. However, the checkpoints as milestones are to be taken into account here. Both a positive and a negative alert can be defined at a checkpoint as a milestone.

Processing aborted (negative alert): This function is to alert you to an aborted processing.

Expected value for file trigger not reached (NoExec alert): This is triggered if the file interface was started (such as start file not available) within the respective scheduling day (and time from-to) less than is defined in the expected value. The alert always happens at the end of the scheduling day (time to).

Expected value for file trigger exceeded (Recovery alert): This is triggered if the file interface was started (such as start file comes too often) within the respective scheduling day (and time from-to) more than is defined in the expected value. The alert happens as soon as the expected value is exceeded.

The alert for **maximum runtime** is triggered when the maximum runtime entered is exceeded. If you set the Abort if exceeded flag, the system ends the file interface.

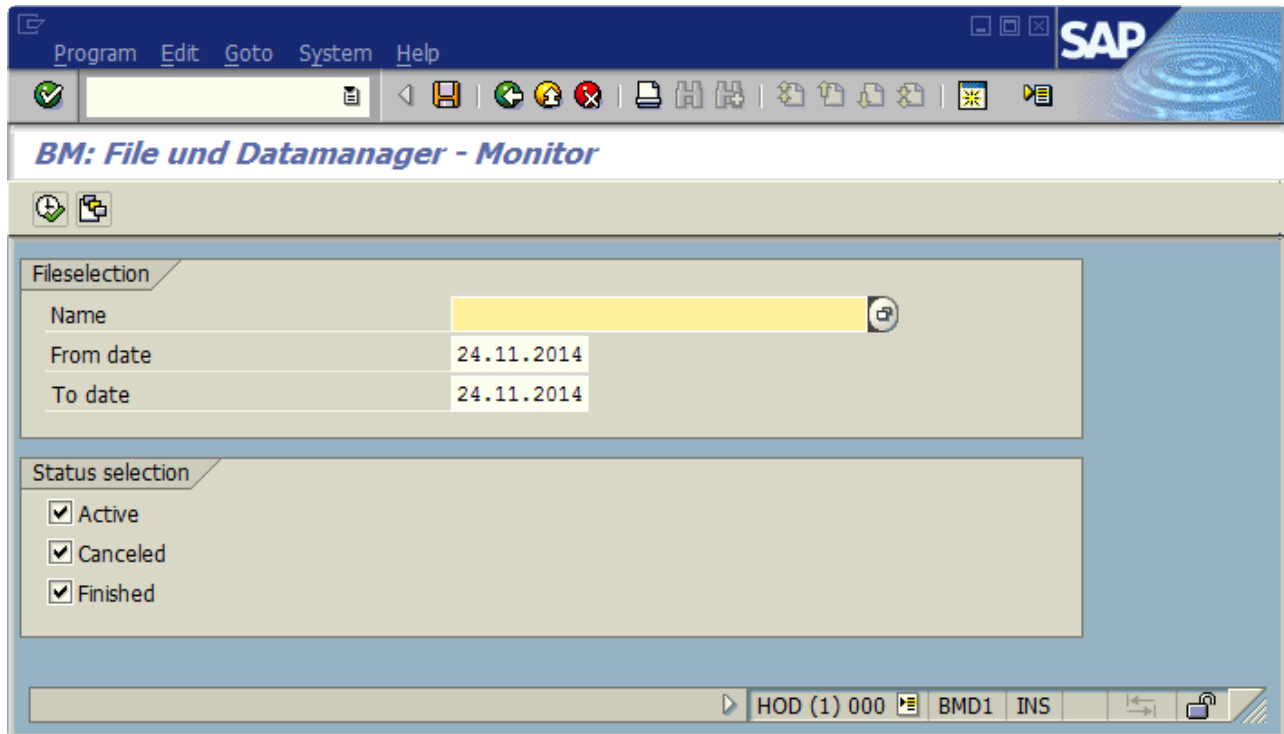
If the Trigger alarm again flag is not activated, the system only triggers the alert. This alert can be triggered again after a second time period (also with various alert tools). Whereby the alert is then triggered n times every x minutes.

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

4. Monitoring

To access the monitoring of the file interface, choose:

BatchMan®-Function tree → File interface → Monitoring → File interface monitoring



On the selection screen of monitoring, you have the option to select a file interface by name and scheduling date. For recurrent selections, you have the option to save selection variants.

If you select by status, the system filters out all processes that do not match the status selected.

Monitoring is displayed as an expandable list. The individual file interfaces are on the uppermost level.

If you expand the uppermost level, you will find the respective scheduling days that result from the Scheduling tab of the concerned file interface. By default, one scheduling day goes from 12am-12am; it is also possible for schedulings to go more than one day (validity + number of days). For schedulings that go more than one day, theoretically schedulings may also overlap. If several schedulings are active, the system always assigns the file interface to be started to the oldest active scheduling.

On the lowest level, below the individual scheduling days, you will find the started file-interfaces (job level) with their step lists.

Monitoring consists of these columns:

File interface	Name of the file interface or the different levels (scheduling day, and so on.)
Stop	The stop flag signals a processing stop on all subordinate levels. The stop flag on the file interface level (uppermost level) corresponds to the active flag in the related file interface definition. The stop flag on the scheduling day level locks this special scheduling day.

Status Shows the status of the respective level. The status of the upper levels results in part from the status of the lower levels. (The system always passes the Aborted status upward).

Following statuses are possible:

active
 finished = successfully ended
 aborted = ended with abort
 faulty = abort in subordinate level
 aborted/confirmed = ended with abort and abort confirmed

EV target Expected value of the file interface defined in the scheduling criteria

EV actual (Total) value of the number of schedulings, at the end of a scheduling day, the EV target should be equal to the EV actual (otherwise the system issues an alarm that EV fell short/exceeded).

Start date/time Local start date and time

End date/time Local end date and time or empty if still active

Manual scheduling User name of the manual starter when scheduling without triggering the start criteria such as manual start, repeat, and so on.

Fileinterface	Stop	Status	EV Soll	EV Ist	Startdatum	Startzeit	Enddatum	Endzeit	Manuelle Einplanung	Beschreibung
FILEINTERFACE 1		active								b
24.11.2014		active	5	4	24.11.2014	12:00:00	24.11.2014	19:00:00		
24.11.2014 15:58:26		active			1 24.11.2014	15:58:26			MINIERZ	
24.11.2014 15:21:18		finished			1 24.11.2014	15:21:18	24.11.2014	15:23:50		
AGENT_LINUX_10.3.21.178 ZIP /tmp/marek/pure-ftpd-1.0.30/*.* /tmp/marek/f		finished			24.11.2014	15:23:48	24.11.2014	15:23:48		zippen
AGENT_LINUX_10.3.21.178 OZIF /tmp/marek/ftp.zip /tmp/marek/pure-ftpd-1.0		finished			24.11.2014	15:23:48	24.11.2014	15:23:48		
AGENT_WIN_10.3.21.244 MDIR C:/Temp/copy_from_linux		finished			24.11.2014	15:23:48	24.11.2014	15:23:48		
AGENT_LINUX_10.3.21.178 COPY /tmp/marek/ftp.zip AGENT_WIN_10.3.21.244 C:		finished			24.11.2014	15:23:48	24.11.2014	15:23:48		
SFTP_10.3.21.178 MDIR /tmp/marek3		finished			24.11.2014	15:23:48	24.11.2014	15:23:49		
AGENT_WIN_10.3.21.244 COPY C:/Temp/copy_from_linux SFTP_10.3.21.178 /tmp		finished			24.11.2014	15:23:49	24.11.2014	15:23:49		
SWH_10.3.21.71 MDIR C:/Temp/copy_from_linux		finished			24.11.2014	15:23:49	24.11.2014	15:23:49		
SFTP_10.3.21.178 COPY /tmp/marek3 SWH_10.3.21.71 C:/Temp/copy_from_linux		finished			24.11.2014	15:23:49	24.11.2014	15:23:50		
SFTP_10.3.21.178 COPY /tmp/marek3 ftp.zip FTP_10.3.21.165 /TEST/test/<SF		finished			24.11.2014	15:23:50	24.11.2014	15:23:50		
24.11.2014 12:00:21		finished			1 24.11.2014	12:00:21	24.11.2014	12:02:52		
24.11.2014 09:54:03		finished			1 24.11.2014	09:54:03	24.11.2014	09:56:36	MINIERZ	
FILETRIGGER_SET_CHKP		active								
FILETRIGGER_TEST		active								
TEST_CHKCP_AUSLOESEN		active								
TEST_CHKCP_WARTEN		active								
TEST_COPY6		aborted								
24.11.2014		aborted	9	24.11.2014	00:00:00	24.11.2014	00:00:00			
24.11.2014 14:51:44		finished			1 24.11.2014	14:51:44	24.11.2014	14:51:46	MINIERZ	
24.11.2014 14:01:07		finished			1 24.11.2014	14:01:07	24.11.2014	14:01:07	MINIERZ	
24.11.2014 13:46:14		finished			1 24.11.2014	13:46:14	24.11.2014	13:46:16	MINIERZ	
24.11.2014 13:45:26		finished			1 24.11.2014	13:45:26	24.11.2014	13:45:27	MINIERZ	
24.11.2014 13:44:24		aborted			1 24.11.2014	13:44:24	24.11.2014	13:44:25	MINIERZ	
24.11.2014 13:35:05		finished			1 24.11.2014	13:35:05	24.11.2014	13:35:05	MINIERZ	
24.11.2014 13:29:36		finished			1 24.11.2014	13:29:36	24.11.2014	13:29:37	MINIERZ	
24.11.2014 11:40:09		finished			1 24.11.2014	11:40:09	24.11.2014	11:40:10	MINIERZ	
24.11.2014 11:39:10		finished			1 24.11.2014	11:39:10	24.11.2014	11:39:11	MINIERZ	
TEST_EVENT_AUSLOESEN		active								
TEST_EVENT_WARTEN		active								
24.11.2014		active	182	24.11.2014	00:00:00	24.11.2014	23:59:59			
TEST_VIELE_STEPS		finished								

You can carry out the following actions using the application toolbar



Update re-import the monitoring data



Expand subordinate levels



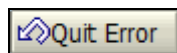
Collapse subordinate levels



Manually start the file interface without taking start criteria into account



Abort file interface processing (takes about 30 sec. + master job period)



Confirm abort (unblock file interface if it is blocked by abort)



Display job log (the whole Job on job level, individual step on step level)



Repeat scheduling (with selection: From step number)



Block file interface (set to inactive in the master data), or block scheduling day



Unblock file interface (set to active in the master data), or unblock scheduling day



Display file interface or on the job level: Display of the started object (file interface at the time of the start)

Individual actions are only available on certain levels or for a certain status.

Individual actions are also available using the context menu of an object (right mouse key).

4.