

## Functional Enhancements in AVAS V8.0 - BS2000/OSD

Issue May 2007

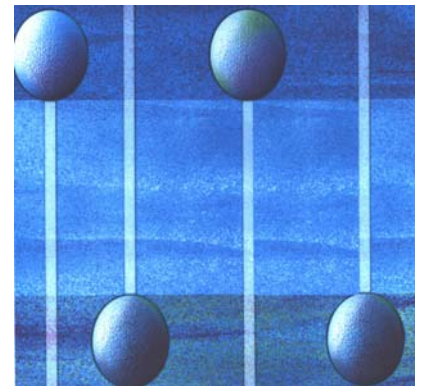
Pages 5

The AVAS job management and handling system automates batch job scheduling and enables all IT production in heterogeneous configurations to be controlled with BS2000/OSD.

Future development of AVAS is aligned to meeting user requirements as identified in customer workshops and in the customer association SICUS, as well as in other forums.

AVAS V8.0 implements the following priority requirements:

- The SYSOUT file of an AVAS job running in a BS2000 system can be displayed during the job's active phase.
- File transfer requests, currently serviced within jobs, can in future be handled via AVAS structure elements, thereby further enhancing the transparency of the runtime logic.
- Integration of subnets into hypernets is improved: The start time of subnets can be linked to the execution of the hypernet, so that the start time for both can be changed to an earlier one in a single step. The USER-PAR-FILE of the hypernet can be inherited by the subnets.



The main functional enhancements for servers in the new version AVAS-SV V8.0 are:

- Additional log files can be transferred in server jobs and thresholds predefined for the scope of the transfer.
- The server monitoring interface will provide more detailed information (JCL, Joblog) as well as the option to halt job distribution. This creates greater transparency on the decentralized system.

Software requirements and release date:

AVAS V8.0 will be released for BS2000/OSD-BC V5.0 or higher.

The general release of AVAS V8.0 is scheduled for October 2007.

### Contents

#### Enhancements in AVAS (BS2000) V8.0

SHOW on running SYSOUT	2
Access to SYSOUT/SYSLST of AVAS processes	2
FILE-TRANSFER functionality via structure element	2
Hypernet/subnet integration	3
Enhancements to AVAS-QUER	4
A separate FHS mask for each command	4

#### Enhancements in AVAS-SV V8.0

Performance improvement for the server monitoring process	4
Signaling additional log files in server jobs	4
Thresholds for server job log files to be transferred	4
Enhancements to the server interface (AVSSURF)	4

## Enhancements in AVAS (BS2000) V8.0

### SHOW on running SYSOUT

The new OUTSYS operation of the NET-CONTROL command can be used during the active phase of an AVAS job in a BS2000 system to access the job's SYSOUT file.

The SYSOUT file for a job can be displayed via EDT using the OUTSYS (#79) operation in masks AVI023, AVI003, AVI027 and AVI029. The operation is only allowed for structure elements (function J or P) with the status RUNNING. The OUTSYS operation invokes mask AVI079, in which the scope of the display (entire SYSOUT file or extracts) can be defined. It is also possible to specify how many PAM pages from the beginning of the file and how many before the end of the file are to be displayed.

The selected part of the SYSOUT file is then read into the EDT editor and displayed.

Access to the SYSOUT file is implemented by the privileged AVAS process SOUT, which, when launched, identifies the file via BS2000 system interfaces, opens it, makes a temporary copy, and transfers it to the AVAS-DIALOG process.

SOUT is a reusable DCAM application. SOUT consists of a primary task and a certain number of secondary tasks. The primary task provides the operating resources of the central DCAM process and controls the application. The secondary tasks handle the requests from the AVAS-DIALOG processes, access the SYSOUT files and transfer them.

The SOUT process must be launched by the AVAS administrator by means of an ENTER call on each BS2000 computer on which AVAS jobs are executed. The SYSOUT file of BS2000 server jobs can also be displayed (AVAS-SV-BS2 product).

The DCAM application name of the SOUT processes must be made known to the AVAS system via new system parameters (SOUT-APPLICATION-NAME or SOUT-APPLICATION-JV).

### Access to SYSOUT/SYSLST of AVAS processes

The SYSOUT and SYSLST files of running AVAS processes can be accessed via corresponding entries in the job variables of the AVAS processes or via the INFORM-PROG command, without the need to terminate the processes.

A copy of the current SYSLST or SYSOUT file is created by entering COPYLST or COPYOUT, respectively.

A new SYSLST or SYSOUT file is assigned by entering NEWLST or NEWOUT, respectively.

The SYSLST and SYSOUT system files of the access processes can be analyzed during live operation after a copy has been created or the assignment changed.

### FILE-TRANSFER functionality via structure element

File transfers to other systems using *openFT* (TRANSFER-FILE command) are supported by a new structure element type FU=F, TYPE=TRA.

The following masks have been newly introduced for processing the new structure type:

- AVN016 (NET-DESCRIPTION, parameter description)
- AVN026 (NET-DESCRIPTION, planning data)
- AVI026 (SHOW-NET-STATUS, parameters)
- AVD026 (MODIFY-SUBMIT-NET, parameters)

The following operands of the *openFT* TRANSFER-FILE command can be specified as parameters on mask AVN016 (and are displayed or offered for modification on masks AVI026 and AVD026):

Parameter	Meaning	Corresponds to operand in TRANSFER-FILE
DIRECTION (TO / FROM)	File transfer direction	TRANSFER-DIRECTION
PARTNER-NAME	Symbolic name of the remote computer, specified by the FT administrator.	PARTNER-NAME
REMOTE {*BS2000 / *MSP / *ANY}	Specifies the type of the remote system.	REMOTE-PARAMETER
LOCAL-FILE	Specifies the name of the file in the local system.	Corresponds to the FILE-NAME operand in the LOCAL-PARAMETER structure
REMOTE-FILE	Specifies the name of the file in the remote system.	Corresponds to the FILE-NAME operand in the REMOTE-PARAMETER structure
REMOTE-TRANSFER-ADMISSION	Access authorization in the remote system.	Corresponds to the TRANSFER-ADMISSION operand in the REMOTE-PARAMETER structure
FT-PARAMETER	Specifies additional operands of the <i>openFT</i> TRANSFER-FILE command. In particular, follow-up processing in the local and remote system can be specified here.	The syntax of the TRANSFER-FILE command must be observed; AVAS does not check this.

From the parameters, the AVAS scheduling function creates a TRANSFER-FILE command with MONJV specification. The command is passed to the operating system via CMD macro and the processing is monitored by the MONJV.

It is recommended to specify UNIX and WINDOWS file name prefixes in particular in the remote system via the FTAC authorization profiles. The same applies to follow-up processing. The link to the FTAC profiles is established via the REMOTE-TRANSFER-ADMISSION operand.

File transfer requests are handled like jobs in the AVAS workflow.

As with a job, a file transfer request can be assigned a short text (FT-TEXT) and a documentation element (FT-DOC, with default name \$bknet\_netname.ftname) in the net description.

A file transfer request can be assigned the same planning data as a job (Select cycle, Symdat with Latest-Start/Delay-Solution/Life-Time).

If the LIFE-TIME parameter is specified in a file transfer request, a condition entry of type JOB is generated when the net is released. Using FU=C, TYPE=JOB in the net description it is now also possible to wait for the condition of a file transfer request to be fulfilled.

The PRINT operation can be used to generate object lists or object descriptions for documentation in list form. The FT parameters can be output via the new list formats L008 (on mask AVN016 NET\_DESCRIPTION) and L026 (on mask AVI026 NET-STATUS).

With CREATE-PLAN-NET, the structure elements FU=F are entered with status PLANNED in the NPRLIB. File transfer requests have no job-describing data (JCL, JCL modules) and are therefore unaffected by the modification.

After the net release (SUBMIT-NET), the individual jobs are processed fully automatically. File transfer requests are executed with /TRANSFER-FILE and monitored, like jobs, by means of a MONJV.

MODIFY-SUBMIT-NET can be used to modify parameters of the structure elements for executing file transfer requests for an already released net. The FT parameters and the specifications for the AVAS session can be modified at the same time (Restart specifications, LATEST-START, DELAY-SOLUTION).

File transfer requests are handled like jobs at RESTART.

Net processing can be interrupted or aborted using the CANCEL-NET statement. CANCEL-NET, KILL-JOBS=YES now also causes file transfer requests in the status RUNNING to be aborted by AVAS in addition to the jobs. For an individual net, an individual job or file transfer request can also be canceled in the display of the net structure.

In the NET-CONTROL command, functions of BS2000 commands can be invoked on the masks for the parameters of the structure elements FU=J/P and FU=C and TYPE=JVA via #7n operations. The BS2000 accesses are performed on the BS2000 system on which the AVAS dialog is running. In AVAS V8.0, file transfer requests can also be accessed from within NET-CONTROL, specifically on mask AVI026 (parameters of the structure element FU=F) using the operations #72 and #73:

- #72 BS2INFO: SHOW-FILE-TRANSFER command on started file transfer requests
- #73 CANCEL: CANCEL-FILE-TRANSFER command on started file transfer requests

Journal record S52 has been extended and a new journal record S60 defined for MODIFY-SUBMIT-NET.

A record is stored in the history file for every file transfer request executed under a net.

The macros AVASJRN, AVSASSAN and AVSCOBAN have been extended for the new structure descriptions, so user programs and exits that use these macros must be recompiled.

openFT versions V9.0 and V10.0 are supported.

## Hypernet/subnet integration

### Linking subnet start times to execution of the hypernet

The start time of subnets can be linked to the execution of the hypernet via the new PLAN-START value \*BY-HYP ("by hypernet"), which is to be entered on subnet Symdats (!symdat) in the net planning data. The subnet will then be started at the time at which its index level is reached in the hypernet.

The link is retained even if hypernet start times are shifted subsequently using MOD-PLAN-NET and MOD-SUBMIT-NET. If the start time of a hypernet is advanced using MOD-PLAN-NET or MOD-SUBMIT-NET, the new start time is brought forward accordingly for the subnets as well.

### Inheritance of the hypernet's USER-PAR-FILE by the subnets

The USER-PAR-FILE of subnets can be supplied via the new USER-PAR-FILE value \*BY-HYPERNET during production from the USER-PAR-FILE value of the hypernet. This enables just one parameter file to be used for the entire hypernet (including subnets).

## Enhancements to AVAS-QUER

The utility routine AVAS-QUER reads the AVAS basic database in the BS2000/OSD system and from it selects data for further processing in relational databases.

More extensive information is output in AVAS V8.0, controllable via the new TABLE-STRUCTURE parameter in the CREATE-FORMAT statement. This enables all the data required for automated creation of an AVAS net to be obtained.

The structures of the following tables have been extended:

- -nettab (assignment of the PVS to nets)
- -netsymtab (assignment of Symdats to nets)
- -ordertab (assignment of subsets to jobs and jobs to nets)

The following new tables will be created:

- -caldaytab (description of calendar days)
- -netformtab (assignment of user masks to nets)
- -netpartab (assignment of the parameter file to nets)
- -orderpartab (assignment of the parameter file to structure elements)

The structure of the following table has been reorganized:

- -ordertexttab (assignment of net description texts to the structure elements of the nets)

## A separate FHS mask for each command

Under control of the new GENPAR parameter "UNIQUE-MASKS", AVAS outputs a unique FHS mask for each command. Hitherto the same masks were used in certain cases for CREATE/MODIFY/SHOW functions, whereas now each of the masks is identified differently. What's more, in addition to mask AVN002 (in future: job description for JOB-TYPE STD) there are separate masks for the job description for JOB-TYPE MOD and EXT, since the parameters for these jobs are input via different interfaces.

This improves the ease with which AVAS workflows can be diagnosed and mapped to the FHS-DOORS interface. With SHOW commands, input fields in which no inputs are possible no longer appear at the AVAS-DOORS interface.

## Enhancements in AVAS-SV V8.0

### Performance improvement for the server monitoring process

To ensure that server jobs are only started on active servers, AVAS provides a monitoring function for the servers entered in the configuration file. This is implemented via the server monitoring process (AVSSVDOG).

In AVAS V8.0, the server monitoring process can be configured to wait for heartbeat messages from the monitored AVAS servers rather than actively querying the status. This can reduce network and CPU utilization.

Under previous versions, a socket connection is set up to each monitored server by AVSSVDOG in accordance with the SV-CONTROL-TIME start parameter and status information is exchanged via this socket. The socket is then cleared down again afterwards. It saves on network resources if, rather than having to set up a complete connection each time, a connectionless UDP message is sent instead.

### Signaling additional log files in server jobs

Up to 96 log files for transfer to the BS2000/OSD system can be signaled in server jobs using the program avslog and the new -file operand. This is done by specifying patterns in file names.

(Upper limit of the CENTRAL application is 99, with two files being reserved for the -stdout and -stderr operands, and one file name space for the log of the proxy job in BS2000).

### Thresholds for server job log files to be transferred

When avslog -file is invoked to signal log files for transfer to BS2000/OSD, the -max-size parameter can be used to specify the size (in KB) up to which the log file is to be transferred. If the size of the log file exceeds this limit, only part of the log file will be transferred. The -head parameter can be used to specify which part of the log file is transferred. The -head parameter defines what percentage of max-size is to be transferred from the start of the file; the remaining percentage is taken from the end of the file.

### Enhancements to the server interface (AVSSURF)

In combination with the server monitoring process (AVSSVDOG) and any web browser, the server interface process (AVSSURF) provides a graphical interface for visualizing the server environment from the AVAS viewpoint. The servers and their status, the jobs started by AVAS on these servers and data associated with these jobs are displayed. In addition, given the appropriate authorization, administrative interventions (stopping/restarting job distribution, modifying the configuration file, canceling jobs, changing log file attributes) are possible in AVAS-SV V8.0.

### Authorization check

AVAS-SURF currently provides only information functions, without performing an authorization check. In AVAS-SV V8.0, an authorization check will be implemented in AVAS-SURF as a requirement for accessing confidential information and action functions. In the definition of users in the AVAS system parameters (USER user definition in GENPAR), the new "MANAGE-SERVER" entry can be used to control whether the user is authorized to manage servers.

The AVSSURF entry mask (AVW001) displays (as previously) the AVAS server configuration. The mask is being extended to include a button which leads to the new mask AVS010, on which the SIGNON information for an AVAS system can be entered. AVSSURF sends the SIGNON information – encrypted if necessary – to the AVSSVDOG addressed by the start parameters SVDOG-HOST/SVDOG-PORT.

This AVSSVDOG connects to the AVAS system via the AVAS batch interface and if the SIGNON is successful, supplies the function authorization table and user group of the AVAS user.

To prevent someone receiving the wrong authorizations via another AVAS system, e.g. a test system, it can be specified, when starting AVSSVDOG, for which SYSTEM-IDs a SIGNON can be performed.

### Management functions for servers

If the AVAS user ID possesses MANAGE-SERVER privilege, radio buttons for the server status \$R (Running), \$I (Ignored) and \$T (Terminated) appear in the 'Status' column of the server table. This enables the server status values to be changed. The following status transitions are permitted:

- \$R -> \$I: No more jobs will be started by AVAS for this server until further notice.
- \$T -> \$I: No jobs will be started by AVAS for this server even if it were to become active in the meantime.
- \$I -> \$R: Job starts are allowed on this server once again.

If the AVAS user ID possesses MANAGE-SERVER privilege, the configuration file can be displayed on mask AVW002 and processed. New entries can be created and existing ones changed or deleted.

### Sorting table outputs

Clicking a button which specifies the name of a server on mask AVW001 causes mask AVW010 to be output with a list of the jobs on this server. All currently active server jobs are listed, along with the server jobs which have already been terminated but whose status has not yet been passed on to AVAS.

By default, the jobs on mask AVW010 are sorted as follows:

- Net name
- Index level
- Job name

The sorting thus corresponds to the sorting order in AVAS/BS2000.

A new option offered is that – as with Windows Explorer – the column headers of the table are clickable.

This causes jobs to be sorted primarily according to this column entry.

The default sorting is used as the second, third and possibly fourth key for the sort.

### Access to confidential information and action functions on servers

On mask AVW010 containing the job list, a user authenticated as an AVAS user and whose AVAS user ID has authorization for the CANCEL-NET function will additionally be offered a 'CANCEL' button for each job. Pressing this button causes the relevant job to be terminated in the background.

If the user's AVAS user ID has authorization to use the SHOW-PROD-JOB function, the job name is presented as a button.

Activating the button causes a branch to be made to mask AVW031, on which the script of the job is displayed.

If the AVAS user ID is authorized to use the SHOW-JOBLOG function, the log file names are represented as buttons. Activating the button causes a branch to be made to mask AVW031, on which the contents of the log file are displayed.

If the AVAS user ID has authorization to use the MODIFY-SUBMIT-JOB function, the attributes of the log files (delete attribute, max. size, header) are displayed as overwritable. These attributes can be modified here and written back to the server by clicking the 'SAVE' button.

### Automatic updating of the status display

In addition to receiving the data for the server table from AVSSVDOG, AVSSURF is also notified when AVSSVDOG is planning to start the next server monitoring cycle (in accordance with the SV-CONTROL-TIME parameter).

This time (plus a contingency reserve of one minute) is incorporated into the web page as an http equivalent so that the browser automatically reloads the page after this time has elapsed.