



Metalogic New Feature Paper

Supervisor Enhancements for Configuration Control (First draft)

Relative to Supervisor version 44.440.09
March 20, 1998

Introduction

The recent Unisys implementation of “1+1” systems whereby a standby system can be automatically switched in the event of a failure on a live system has impacts on the running of Supervisor in such environments. Since the switching procedure involves the movement of multiple disk families and applications from one system to another, Supervisor on the standby must have the capability to automatically continue processing the transferred SCHEDULE file and any OPAL programs (WHENs) that were active prior to the switch.

To make this switching process as painless as possible, Supervisor now has several enhancements to allow the sharing of SCHEDULE and MAGUS configuration data files between the two systems. It is anticipated that the following release levels of the Magus library and Supervisor will support this functionality:

MAGUS	44.440.06
SUPERVISOR	44.440.06

Please note that this paper is very much a draft version; any suggestions or omissions will be gratefully received.

“Critical” files

There are several files which are critical to this implementation. First, Magus utilises a file usually called:

***METALOGIC/MAGUS/CONFIGURATIONDATA**

The name is derived from the SL of the MAGUS library codefile and, by default, will always be searched for under the codefile directory. This file holds licence keys and much Supervisor environmental information, mostly seen in the response to a TT USE command. The presence of valid key information, inserted by the INSTALL utility, is absolutely essential; no Metalogic software will run without valid keys.

Entrypoints in the MAGUS library are used to control all access to this file and its data so Magus must be informed of the presence of an alternate configuration when necessary. When Supervisor initialises, he retrieves this environment information and, in particular, the location of the SCHEDULE file which is also very critical. This file is always called *SCHEDULE and, by default, resides on the DL JOBS family unless this is overridden by the Supervisor command:

TT USE FAMILY <family> FOR SCHEDULE

Since this specification is derived from an environment variable held in the configuration data file, it is apparent that the configuration data file is of considerable importance when switching systems.

These environment variables are given specific names; for example, **the USE FAMILY.. FOR SCHEDULE** is called **SUP_SCHEDULEFAM** and **USE FAMILY ... FOR EXTERNAL** is called **SUP_EXTERNALFAM**. The various configuration data environment variables, applicable to Supervisor and TRIM, are listed in Appendix A of this document.

This implementation handles the automatic handling of these two critical files in the event of a 1+1 standby switch.

Implementation features

An existing Supervisor command is already available:

TT USE FAMILY <shared family> FOR SCHEDULE

This allows the override of the normal DL JOBS assignment of the SCHEDULE file to a user-selected location. Supervisor will always COPY the current SCHEDULE to the new family, overwriting any file already present, and change the default location for any subsequent restarts.

If, for any reason, the SCHEDULE could not be moved the error message:

SCHEDULE COULD NOT BE MOVED

is displayed and the location move will be cancelled.

Further, a new Supervisor command has now been implemented :

TT USE FAMILY <shared family> FOR CONFIG

This permits the override of the SL MAGUS determined location for the configuration data file. When this command is used, Supervisor will check that a valid configuration data exists on the specified file and will respond with various error messages if the file is missing or the family is not on-line.

NO CONFIGURATIONDATA FILE ON SPECIFIED FAMILY

Unlike the **TT USE FAMILY . . FOR SCHEDULE** assignment, Supervisor will **not** automatically copy the existing configuration data file; this must be done manually. Also, it should be noted that the command should only be used after the system switch has taken place and all shared families have been moved from the previous “live” system.

If a good configuration data file is found, then Supervisor will call a new entrypoint in the Magus library to inform him to start using the new configuration. Further, the Magus library will perform checks on the new configuration data file; if these tests fail, the switch will be aborted and the **TT USE** assignment will be cancelled:

SWITCH TO ALTERNATE CONFIG REFUSED

The Magus library will display error messages as to the reason why the file verification processed failed. The variable name used by Magus to hold this family location is called **MAGUS_ALTCONFIG**.

Any program, including Install, Supervisor and TRIM, that wishes to access configuration data variables or OPAL permanent variables will be always directed to use the new configuration. However, some variables are always exempt:

**All licence keys e.g. SUPKEY44.440.05
MAGUS_ALTCONFIG**

The above variables will always be accessed from the default configuration data file which should never be removed. When the Magus library restarts, the keys will be loaded and checked from the usual configuration data file but the **MAGUS_ALTCONFIG** variable will also be checked and, if valid, the alternate configuration will automatically be activated.

After the alternate configuration data file has been successfully loaded, Supervisor will need to reload all his environment variables and the shared SCHEDULE file, which should already have a location assignment in the alternate configuration. This will be done automatically; the SCHEDULE file is closed, all active WHENs are terminated and some clean-up is performed. Supervisor then performs a pseudo-initialise, extracting environment information from the alternate configuration and loading the shared SCHEDULE. The following messages will be seen:

WHEN TERMINATION INITIATED
SUPERVISOR NOW LOADING ALTERNATE CONFIGURATION

All WHENs that had been previously active on the “live” system will be restarted. Note that the processed WHENs from the old SCHEDULE have not really been terminated; once the default SCHEDULE file is reinstated when the system returns to standby, those same running WHENs will be automatically restarted.

It is recommended that the default configuration data file is kept as small as possible. Ideally, it should just hold environmental variables and minimal key information, just to keep Magus overheads to a minimum.

Lastly, it is recommended that the **TT USE** family specifications are different for the default and alternate configurations. In particular, **TT USE FAMILY . . FOR SAVES** indicates where the **SAVED/SCHEDULE** file is copied to during a **SAVE SCHEDULE** and if the setting was the same, could cause confusion if there were problems after a switch.

Initial installation

It is assumed that new Metalogic software releases will, in the future, only be installed on standby systems though, in the initial phase, both the live and standby systems must be installed with a Supervisor and Magus capable of supporting the switching implementation. This section details the procedures necessary for the first installation on both systems.

First, ensure that the Metatapelib DESCRIPTION file is available and the same on both systems.

On the current “**standby**” system:

1. Do a normal install of the Metalogic software including the Metatapelib Tape Library. All family-specific environment variables should be set to non-shared families, e.g. DISK. Only those WHENs needed to run the standby system should be left running.
2. If not, the variables should be set using the **TT USE** command or by the Metalogic Install utility. A list of all relevant variables is available in Appendix A. Alternatively, when running the Install utility, answering NO to the question:

Do you want to keep all installation parameters?

will force you to verify and, if necessary change, all environment configuration variables. Though this may take a little time, it is probably a lot less laborious rather running the Install utility many times.

3. Supervisor can now be run as normal. It would be easier if the Tape Library did not run on the standby system; this is achieved by ensuring the TAPELIB option is reset:

TT SO - TAPELIB

Make sure that you do not have any scheduled OPALs (AFTERS) or OPAL WHENs, of type TAPELABEL or TAPEDB, that will access the Tape Library system. Anyway, it may be a good idea to delete all unwanted OPAL programs and AFTERS to keep the SCHEDULE file compact.

4. COPY the current *METALOGIC/MAGUS/CONFIGURATIONDATA and SCHEDULE files to tape and keep.

On the current “live” system, the following one-time procedure should be followed:

1. All Metalogic software should be installed as normal, including the Metatapelib tape library. All family related environment information, accessible via the **TT USE** command, should be pointing to the correct shared families.

2. If not already done:

TT USE FAMILY <shared> FOR SCHEDULE

will copy the SCHEDULE file from its current location to the family <shared>, if necessary.

3. Manually copy the default MAGUS configuration data file, from wherever Magus is SL-ed, to a user-selected shared family.

4. Next, a “fake” switch needs to be initiated:

TT USE FAMILY <shared> FOR CONFIG

5. This will tell Supervisor and Magus to start using the configuration data file on the <shared> family for all environmental information. Next, Supervisor will perform pseudo restart, reloading his on own environment information, restarting the WHENs and accessing the Metatapelib Tape Library.

This procedure is really be a “no-op” because the SCHEDULE file will be identical but it needs to be done to establish the **MAGUS_ALTCONFIG** setting.

6. Lastly, copy the SCHEDULE and CONFIGURATIONDATA files from the tape produced on the standby system to the DL JOBS family, overloading the existing files. This means that when the live system again becomes a standby, the correct defaults will be available pointing to minimal packs.

7. **Important:** The overloading of the CONFIGURATIONDATA file in (6) will overwrite the Metalogic licence keys for this system with those from the standby. You must re-input the keys after the COPY has completed by

U META/INSTALL KEYS 44.440.6

Where 44.440.06 is the version of Magus library. This means that the next time the Magus library is restarted, the correct keys will be loaded.

The installation of the Metatapelib Tape Library on both systems may cause a problem because the compiled OPALTAPELIB library and tailored software may be incompatible when the CONTROL file is switched from the other system. Since it is envisaged that Metatapelib will not be run on the standby, it is the recovery of the database after the switch that may present problems.

Switching systems

In the event of a system switch because of a live system outage:

1. Once the appropriate shared families are available on the standby,
TT USE FAMILY <shared> FOR CONFIGURATION
2. As before, Supervisor and Magus will load the alternate configuration and Supervisor will then pseudo-restart with the shared SCHEDULE, accessing the alternate environment variables and invoking the live system WHENs.

On the “new” standby system, the Magus library will restart and search for the family assigned to **TT USE FAMILY . . FOR CONFIG**. However, this family will now be unavailable and Magus will automatically delete it from the configuration data file.

This means that all subsequent switches from standby to live will always require **the TT USE... CONFIG** command to be used to not only initiate the configuration switch but also re-establish the **MAGUS_ALTCONFIG** variable for future restarts.

Subsequent installations

Future installations of Metalogic software, as we understand will only occur on standby systems. A new installation should not cause any obvious problems unless there are very significant changes to the way these procedures have been implemented.

However, one outstanding issue mentioned earlier still needs resolution: the Metatapelib database may have problems with the movement of the CONTROL file between systems. and may cause incompatibility problems for the DMSII tailored software and our OPALTAPELIB library after a switch. This problem has still to be addressed.

APPENDIX A: CONFIG variables

This is the list of all family-specific configuration variables that are used by Supervisor, Magus and the Tape Library system. Many of these can be changed using the TT USE command but others can only be changed using the Install utility.

For example, from a privileged usercode:

```
U META/INSTALL SUP_SCHEULEFAM=WORK
```

Will change the current setting. To interrogate the current value use:

```
U META/INSTALL SUP_SCHEDULEFAM
```

MAG_FAMILY **Type: Familyname**

Setting this variable sets the family for the MAGUS library and CONFIGURATIONDATA files. The INSTALL utility will usually ask for the name of the desired family during the first installation.

e.g. **MAG_FAMILY=DISK**

MAGUS_ALTCONFIG **Type: Familyname**

This variable controls the switch to an alternate Magus CONFIGURATIONDATA file, allowing different Supervisor environments to be used. If set by the TT USE FAMILY FOR CONFIG command, Supervisor will force Magus to use the new alternate configuration immediately.

e.g. **MAGUS_ALTCONFIG=WORK**

SUP_BACKUPFAM **Type: Familyname**

Setting this variable determines the family used by SUPERVISOR to SAVE a backup image of the *SCHEDULE file. Also settable via the TT USE FAMILY FOR SAVES command in SUPERVISOR.

e.g. **SUP_BACKUPFAM=DBSPACK**

SUP_CODEFAM **Type: Familyname**

Setting this variable determines the family on which SUPERVISOR's code files should be stored.

e.g. **SUP_CODEFAM=DISK**

SUP_TLJOB **Type: Filetitle**

Setting this variable determines the name and location of the job symbolic files. It may also be set by Supervisor's TT USE command.

e.g. SUP_TLJOB=*TL/HANDLER

SYS_FAMILY **Type: Familyname**

Setting this variable determines the family where SUPERVISOR expects to find system codefiles e.g. SYSTEM/FILEDATA, SYSTEM/LOGANALYZER etc.

e.g. SYS_FAMILY=DISK

TL_ACRFAM **Type: Familyname**

Setting this variable sets the METATAPELIB DASDL family assignment for the SYSTEM/ACCESSROUTINES library and maintained by the INSTALL utility.

e.g. TL_ACRFAM=DISK

TL_AUD2FAM **Type: Familyname**

Setting this variable sets the METATAPELIB DASDL family assignment for SECONDARY audit and maintained by the INSTALL utility.

e.g. TL_AUD2FAM=WORK

TL_AUDFAM **Type: Familyname**

Setting this variable sets the METATAPELIB DASDL family assignment for PRIMARY audit and maintained by the INSTALL utility.

e.g. AUDFAM=DBSPACK

TL_CFFAM **Type: Familyname**

Setting this variable sets the family for the tape library's database CONTROL file, as used by the METATAPELIB DASDL compilation job and maintained by the INSTALL utility.

e.g. TL_CFFAM=DISK

TL_DBFAM**Type: Familyname**

Setting this variable specifies the family used to hold tape library database sets and datasets, as used by the METATATAPELIB DASDL compilation job and maintained by the INSTALL utility.

e.g. TL_DBFAM=PACK

TL_DMSUP**Type: Filetitle**

Setting this variable determines the name and location of the DMSUPPORT library, as used by the METATATAPELIB DASDL compilation job and maintained by the INSTALL utility.

e.g. TL_DUMP=(TAPELIB)DMSUPPORT/METATAPELIB ON PACK

TL_TAILFAM**Type: Familyname**

Setting this variable specifies the family for the tape library database TAILORED software, as used by the METATATAPELIB DASDL compilation job and maintained by the INSTALL utility.

e.g. TL_TAILFAM=PACK

TL_TAPEDB**Type: Filetitle**

Setting this variable determines the name and location of the Tape Library database, METATAPELIB, as used by the METATATAPELIB DASDL compilation job and maintained by the INSTALL utility.

e.g. TL_TAPEDB=(TAPELIB)METATAPELIB ON PACK

TL_TAPELOG**Type: Filetitle**

Setting this variable determines the name and location of the METATAPELIB Tape Log file used to hold all tape event and command information.

e.g. TL_TAPELOG=(TAPELIB)METATAPELIB/TAPELOG ON PACK