

UTILITIES

Reference Manual

Relative to Jampack version 530.05

WaitWatcher version 530.02

Contents

Utilities overview.....	5
Installation.....	5
JAMPACK Utility	7
Introduction.....	7
Configuring JAMPACK.....	7
JAM_NOAX.....	8
Command syntax.....	8
HELP.....	9
<family name>.....	9
Capabilities.....	10
SQUASH.....	10
Simulation request (space consolidation).....	11
'SECTORS REQUIRED' hangs.....	12
How JAMPACK works.....	13
Example JAMPACK run.....	14
File Exceptions.....	15
JAMPACK messages.....	15
Error messages.....	15
Warning and Information messages.....	17
Running JAMPACK from SUPERVISOR.....	18
Running JAMPACK from WAITWATCHER.....	18
Summary.....	19
WAITWATCHER utility	20
Introduction.....	20
WAITWATCHER configuration.....	20
Running WAITWATCHER.....	21
Operator interface.....	21
FAMILY.....	22
FP.....	23
HO.....	25
IGNORE.....	26
NOFILEDS.....	27

QUIT.....	28
SPARE.....	28
Copyright.....	30
ACKNOWLEDGEMENTS.....	30

Recent Changes

JamPack

May 2008 530.03:CheckerBoard Report

JamPack can now produce a readable CheckerBoard report similar to the one in *SYSTEM/FILEDATA. However, the report is produced typically in just 5% of the Elapsed time that FileData uses and only 2% of the CPU time. Also, unlike FileData it reports not only on Permanent file in the SYSTEMDIRECTORY, but on temporary files as well as many other types of disk usage.

The syntax is:

```
<Family> # <Inx> SEG <Address> FOR <length>
```

For example:

```
DISK #2 SEG 1000000 FOR 10000
```

If # <inx> is omitted #1 is assumed.

The report is produced in a permanent disk file called CHKBD, which can be Label-Equated. Each contiguous area has a line in the file of the format:

```
<start> <hyphen> <end> <fixed> <flags> <row type>
```

where <start>. and <end> are the start and end addresses

- <hyphen> is a space id this area is a single row, or a '-' if it is made up of multiple contiguous rows of the same file.
- <fixed> is normally a space, but is 'F' if the row would block or otherwise give problems to a RESERVE.
- <flags> is normally space, but is 'T' if the file is temporary, and 'O' if the file is in use.
- <row type> is either a FILENAME or one of the following:
 - >> Label << - The unit label area (at address 0 to 27)
 - >> Olay Row << - An overlay row (only on DL OLAY family)
 - >> MCP Info << - The Mem Dump workspace (only CM'ed Families)
 - >> Unavailable << - Areas that are in use, but which are not known to RESERVE.
 - >> Job File << - A job in the live JOBDESC (on DL JOBS family)

>> Obsolete Job << - Rows in an obsolete JOBDESC, which should be removed.

For example:

```
0 27      F  >> Label <<
28-4027    O  *SYSTEMDIRECTORY/001
4093 4597  (SHAFT)METALOGIC/MG
```

Utilities overview

This manual discusses the capabilities and operations of two Metalogic software utilities called JAMPACK and WAITWATCHER that can help improve the operability and day-to-day running of Unisys Clearpath mainframe systems.

JAMPACK is a powerful, standalone utility that can rapidly resolve SECTORS REQUIRED waiting entries or relieve disk fragmentation problems that frequently occur on high-usage disk subsystems. Similar to the disk fragmentation utilities in the PC environment, even systems with large disk subsystems can benefit from improved performance with regular maintenance.

WAITWATCHER is a continually running utility that monitors new waiting entries in the system, searching for events such as SECTORS REQUIRED, NO FILE and REQUIRES PK conditions. In each of these situations, WAITWATCHER can perform pre-programmed actions to resolve the waiting entry. If JAMPACK is available, SECTORS REQUIRED conditions can be handled by WAITWATCHER by running JAMPACK upon detection of the waiting entry.

JAMPACK and WAITWATCHER are described in more detail in the remainder of this document.

Installation

Like all other METALOGIC products, the installation is prepared by running the INSTALL utility. You should refer to the Metalogic INSTALL manual documentation for more detailed information, but the basic process is extremely simple.

All Metalogic releases are now held in zipped, Unisys wrapped containers distributed over the Internet. Customers are notified of a new release by email that will include a download URL.

The container is moved to the mainframe using drag-and-drop to a mapped MCP disk share or BINARY FTP. The container name consists of a site id (up to 12 characters) followed by an internal Metalogic release identity.

The INSTALL software should be unwrapped from the container, running from a MARC or CANDE station running under a privileged usercode:

Once the container is available, the INSTALL utility, named *OBJECT/META/INSTALL, should be unwrapped:

```
UNWRAP *OBJECT/META/= OUTF META_1521 TO DISK(RESTRICTED=FALSE)
```

From a Cande station, running under a privileged usercode with visibility to the *OBJECT/META utilities, enter:

U META/INSTALL

Most Metalogic software requires a valid licence key to run; all keys are automatically loaded by the INSTALL utility and verified by a Metalogic system library function called MAGUS that is loaded during install.

In the case of JAMPACK and WAITWATCHER, some configuration is required and the INSTALL program will prompt the user if these settings require adjustment. Please refer to the individual chapters on each product and the **Metalogic INSTALL Reference manual** for more information.

In practice, the Metalogic software environment is maintained by a series of “configuration variables” which are accessible using the INSTALL utility. These variables are held in a MAGUS controlled file called:

***METALOGIC/MAGUS/CONFIGURATIONDATA**

The INSTALL utility allows some minor configuration of both the JAMPACK and WAITWATCHER utilities; these features are discussed in the following chapters.

Introduction

The MCP allocates space on packs in variable sized areas and as the files occupying these areas on a disk are being created, expanded or removed, the numbers and sizes of the available areas on the disk will change considerably over time.

This leads to an effect known as "checkerboarding", the result of which is very visible on most systems when programs hang with 'SECTORS REQUIRED' messages, even though there is much more available space than is needed. The problem is that the available space will usually become distributed in small chunks scattered over the entire disk, especially if there are large numbers of small, volatile files on the family. If nothing is done to alleviate checkerboarding, ideally 17% of a pack must be kept un-used to avoid a 'SECTORS REQUIRED' hang.

JAMPAK is an essentially "intelligent" RESERVEDISK, performing multiple ODT RESERVE commands to make the MCP dynamically move file rows to optimise and consolidate available disk. For fast responses to disk shortage problems, JAMPAK will only move those rows that will release the maximum available space in the shortest time. Please note that JAMPAK can "squash" any family including DISK.

JAMPAK'S greatest short-term benefit is its fast and reliable resolution of SECTORS REQUIRED waiting situations, usually within 30 seconds. This helps to eliminate production delays and wasted machine time plus a great deal of operator frustration. If JAMPAK is used as a prevention mechanism rather than cure, its long-term benefits enable even the most heavily occupied pack families to optimise their available disk space by reclaiming areas made useless because of checkerboarding effects.

As a consequence, JAMPAK can also improve I/O performance by reducing head travel and therefore seek time, by balancing I/O loads between all members of a pack family.

Configuring JAMPAK

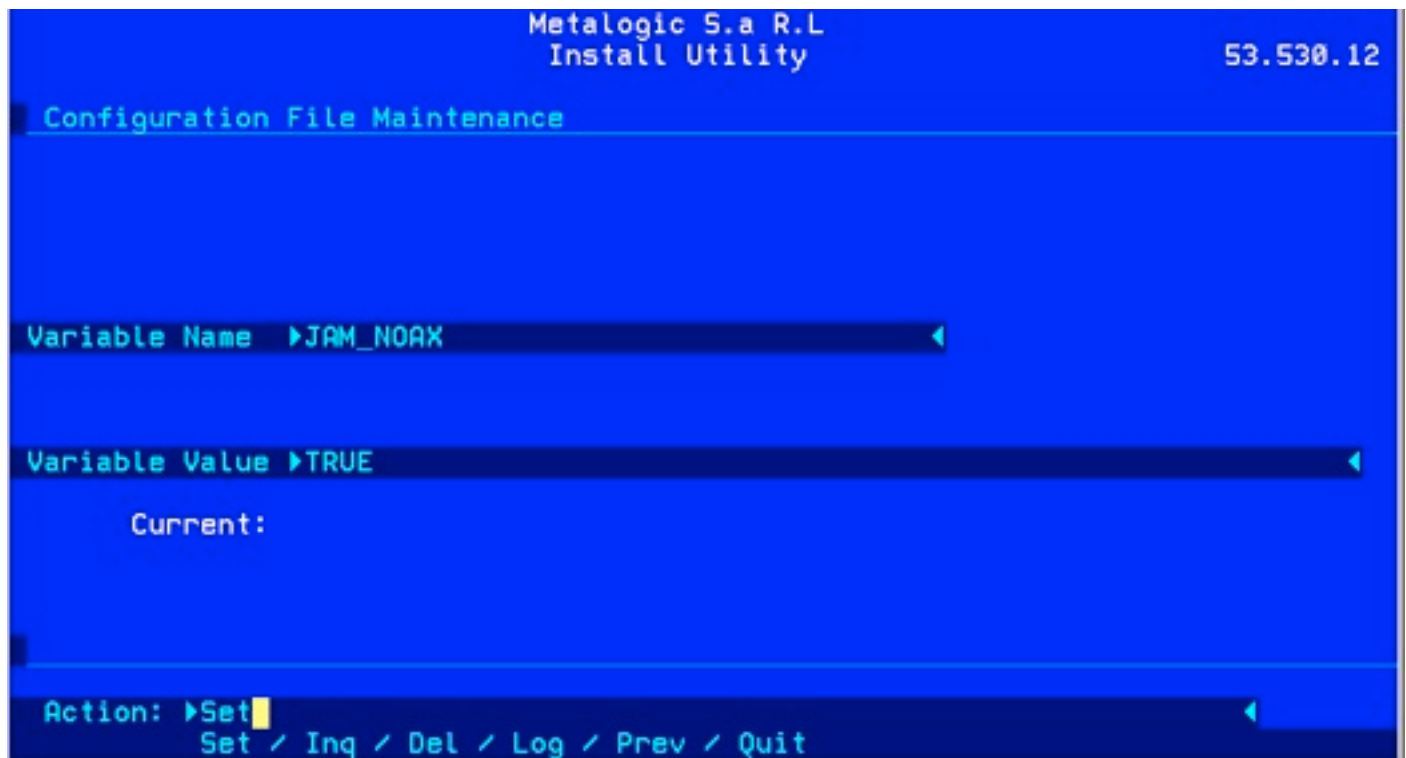
There is no standard configuration available for JAMPAK using the INSTALL utility. However, via the MAINT menu, INSTALL will permit changes to non-standard configurations items.

For JAMPACK, one such setting exists:

JAM_NOAX

For sites running JAMPACK from MARC stations, the regular appearance of the normal JAMPACK 'QUIT' waiting entry seen during each RESERVE run can cause consistent screen updates. The behaviour of this waiting entry can now be controlled setting the JAM_NOAX configuration variable. If this variable is set to 'TRUE', the waiting entry will be suppressed. If the variable is 'FALSE' or not present, JAMPACK will operate as before.

U META/INSTALL MAINT



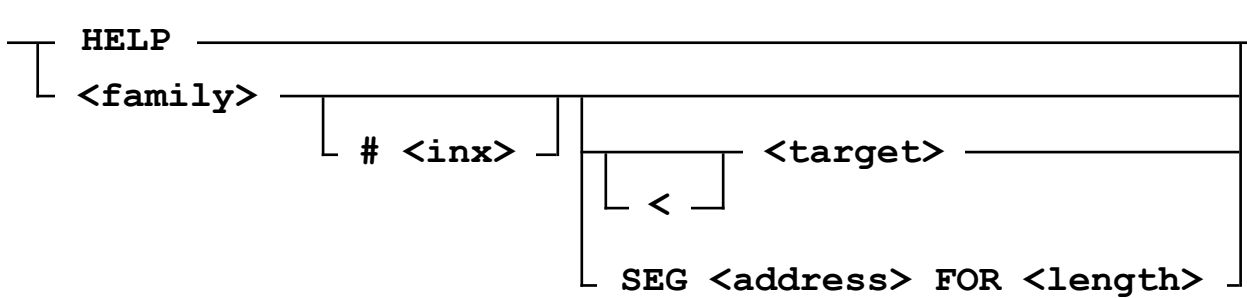
After change, any new invocation of JAMPACK will use this setting.

Command syntax

JAMPACK has various modes of running, allowing responses to immediate disk shortages or revitalising pack families that have severe fragmentation problems. The user can select the following options by changing the parameter passed to JAMPACK.

```
RUN *METALOGIC/JAMPACK ("<STRING>")
```


Where <string> may be:



HELP

Provides a printed description of the commands described here. The logical file LINE has a KIND of PRINTER and may be file-equated.

<family>

The <family name> modifier is the name of a user-selected, on-line, write-enabled pack family. Using the # <inx> modifier allows the selection of an individual familyindex member of the <family name>. An overview of each of the three ways of running JAMPACK is shown in the Capabilities section at the end of this chapter. The <target> modifier is an unsigned integer that is used as follows:

1. If no <target> is specified, a full squash is attempted by performing multiple RESERVES. If allowed to run to completion, JAMPACK will effectively optimise and compress the areas into available and non-available. Note that this can be a very lengthy process if the pack is in-use because any active in-use rows will prevent effective partitioning.

```
RUN *METALOGIC/JAMPACK ("MYPACK #2")
```

Each RESERVE increases the usable space on the pack, so the squash does not necessarily have to complete to reduce fragmentation. It may be terminated before completion by an "AX QUIT"

2. If '<' precedes the <TARGET> modifier, a careful reservability analysis is conducted (known as the SIMULATION run), then a single reserve is done. If <TARGET> cannot be found, the largest possible <TARGET> is used instead. This form is suitable for making very large areas in which case the pack should be lightly loaded. It is also useful if the pack is very fragmented. (See section 2.2.2 SPACE CONSOLIDATION).

```
RUN *METALOGIC/JAMPACK ("MYPACK < 10000")
```

3. A <target> modifier specified on its own uses a much faster analysis to solve a problem. This is usually the best mechanism for handling 'SECTORS REQUIRED' situations. If the RESERVE generated itself gets a problem, JAMPACK converts the request to the '<' case.

When JAMPACK finishes calculations, a RESERVE is invoked and JAMPACK will wait for it to complete. If the RESERVE completes, JAMPACK will automatically restart and remove the BADDISK file created by the RESERVE. If the RESERVE hangs, JAMPACK will DS it and retry as appropriate.

JAMPACK may be stopped prematurely by entering a QUIT in the response to the AX, in which case JAMPACK will then DS the current RESERVE and then stop. If JAMPACK is performing a full squash, it will wait for the end of the current RESERVE before quitting. If a second QUIT is input, then JAMPACK will DS the current RESERVE and quit immediately.

```
RUN *METALOGIC/JAMPACK ("MYPACK 30000")
```

4. If SEG <address> FOR <length> is specified then a checkerboard report is produced

The different modes of operation discussed here are described in more detail in the next section, 'Capabilities'.

Capabilities

JAMPACK has three different modes of operation that are used to resolve long-term fragmentation problems or rapid resolution of SECTORS REQUIRED situations. In addition a checkerboard report may be generated.

SQUASH

If a pack is not currently in use, JAMPACK can "squash" all the available space into one contiguous area at the end of the pack. This is quite a long operation, as a

large proportion of the data on the pack must be moved, so it would normally be scheduled at night or over a weekend. This mode is very similar to the Unisys ODT command SQUASH but a JAMPACK 'squash' does not require repeated operator input (you have to input multiple ODT SQUASH commands to get optimal results).

For example, if the ODT command DU was used, the following response might be seen (on 53.1 MCP):

```
du on dev
Disk Usage Report for DEV (Entire Family)

-- Sectors --      Bytes      Percent
Available: 11,881,355 2,138,643,900 49 %
In Use : 12,230,096 2,201,417,280 51 %
Capacity : 24,111,451 4,340,061,180 100 %

Available Space Analysis

Areasize -- Sectors --      Bytes      Percent
Largest : 10,167,404 1,830,132,720 42 %
< 504: 346,156 62,308,080 1 % (5,572 Areas)
> 503: 11,535,199 2,076,335,820 48 % (81 Areas)

22,857 areas of 504 sectors could be allocated
```

The key values in the above display are the 'Available' and 'Largest AreaSize' sector values. In the above example, the largest area is about 85% the size of the total available space. This is an example of a pack where an overnight run of JAMPACK with the parameter "DEV" is done on a regular basis.

```
RUN *METALOGIC/JAMPACK ("DEV")
```

Checkerboarding problems, causing file fragmentation and slowing down file access times would be minimised.

JAMPACK will 'SQUASH' the pack as far as it can and will stop either at operator request or at the point when no further space consolidation is possible or practical.

Another DU ON DISK, input after a full JAMPACK squash, might show the 'Largest Areasize' very close to that of 'Available' sectors. In practice, this is not usually possible if the family is in-use during the duration of the squash - especially if there are multiple data files, open by applications, since disk rows belonging to these files cannot be moved.

Simulation request (space consolidation)

Although a JAMPACK "squash" can completely eliminate checkerboarding, a run for a large family may take significant time to run. In practice, it is often sufficient to

ensure that a much smaller contiguous area is available to reduce the likelihood of 'SECTORS REQUIRED' hangs.

To allow for this, 'simulation' request' mode allows JAMPACK to create a single large available area (not a maximum) on a lightly loaded pack. This mode is enabled by using the '<' modifier prior to a target sector size.

```
RUN *METALOGIC/JAMPACK ("DEV <300000")
```

The run time is typically a few minutes for small disk families but can take longer for larger families, though this is practical outside peak periods.

The amount of contiguous disk space requested is specified in the parameter, and if that amount cannot be easily created without a full SQUASH, the target figure is optimised and reduced to a more economical value.

'SECTORS REQUIRED' hangs

'SECTORS REQUIRED' is one of the most intractable of operational problems. It can seriously affect system throughput and will usually occur during the busiest periods in the day and on the most heavily used packs. JAMPACK is engineered to solve 'SECTORS REQUIRED' in SECONDS, even on a heavily used family or pack, and even on the halt-load unit.

For example, if the following typical waiting entry was generated from a LIBRARY/ MAINTENANCE job

```
#1234:PK45 504 SECTORS REQUIRED ON MYPACK *SYMBOL/MCP
```

Run JAMPACK with the parameter

```
RUN *METALOGIC/JAMPACK ("MYPACK 504")
```

Typically, the above command would solve the waiting entry within 30 seconds.

This mode is particularly powerful when used in conjunction with METALOGIC/ SUPERVISOR or the WAITWATCHER utility, so that 'SECTORS REQUIRED' is rapidly detected, and JAMPACK is run automatically.

Checkerboard Report

JamPack can produce a readable CheckerBoard report similar to the one in *SYSTEM/FILEDATA. However, the report is produced typically in just 5% of the Elapsed time that FileData uses and only 2% of the CPU time. Also, unlike FileData it reports not only on Permanent files in the SYSTEMDIRECTORY, but on temporary files as well as many other types of disk usage.

The syntax is:

```
<Family> # <Inx> SEG <Address> FOR <length>
```

For example:

```
DISK #2 SEG 1000000 FOR 10000
```

If # <inx> is omitted #1 is assumed.

The report is produced in a permanent disk file called CHKBD, which can be Label-Equated. Each contiguous area has a line in the file of the format:

```
<start> <hyphen> <end> <fixed> <flags> <row type>
```

where <start>. and <end> are the start and end addresses

- <hyphen> is a space id this area is a single row, or a '-' if it is made up of multiple contiguous rows of the same file.
- <fixed> is normally a space, but is 'F' if the row would block or otherwise give problems to a RESERVE.
- <flags> is normally space, but is 'T' if the file is temporary, and 'O' if the file is in use.
- <row type> is either a FILENAME or one of the following:
 - >> Label << - The unit label area (at address 0 to 27)
 - >> Olay Row << - An overlay row (only on DL OLAY family)
 - >> MCP Info << - The Mem Dump workspace (only CM'ed Families)
 - >> Unavailable << - Areas that are in use, but which are not known to RESERVE.
 - >> Job File << - A job in the live JOBDESC (on DL JOBS family)

How JAMPACK works....

Depending on the mode of operation, JAMPACK calculates the optimal RESERVE command that is needed to resolve the problem. In SQUASH mode, this process may be repeated many times whereas tackling a SECTORS REQUIRED situation may only require a single calculation.

Once JAMPACK has processed the RESERVE command, JAMPACK watches the RESERVEDISK as it executes and waits for the process to terminate or stop in the waiting entries. If RESERVEDISK enters the waiting entries, JAMPACK will immediately DS the task and issue a new RESERVE command. Typically,

RESERVEDISK enters the waiting entries if a disk row movement is blocked because the file is in-use – this is usually a timing problem because JAMPACK will automatically exclude rows belonging to in-use files from its analysis.

JAMPACK tracks BOJ, WAITING and EOJ events to make sure that the RESERVEDISK is handled correctly. It is possible for the MCP to schedule a RESERVEDISK under a variety of circumstances (e.g. HS is enabled) and JAMPACK will automatically FS the schedule entry unless the system schedule reason is due to DL LOG disk space shortages.

Whilst JAMPACK is waiting for RESERVEDISK to terminate, it is possible to force early termination of JAMPACK, using the AX QUIT command:

```
<MIXNO JAMPACK>AX QUIT
```

As stated elsewhere, for the space consolidation simulation and SECTORS REQUIRED modes, a QUIT will cause an immediate DS of the RESERVEDISK and JAMPACK will go to EOT.

In the case of a full SQUASH, JAMPACK will go to EOJ after the current RESERVEDISK has finished unless a second AX QUIT is entered. This process slightly speeds up the response to the RESERVE completion and reduces the resources JAMPACK takes as it waits.

Example JAMPACK run

Consider JAMPACK running in 'SQUASH' mode to de-fragment the family DEV:

```
RUN *METALOGIC/JAMPACK ("DEV")
```

The following messages might be displayed as JAMPACK issues the first RESERVE command:

```
3210 12:53 JAMPACK:RES PK502 SEG 51595 FOR 2164958
3210 12:53 ACCEPT:'3210 AX QUIT' TO QUIT AFTER THE RESERVE
COMPLETES
3212 12:53 PK502 SURVEYING FILES
FOR 'RES' OR 'SQUASH'
13% IN *SYSTEMDIRECTORY/001 ON DEV
3212 12:53 PK502 SURVEYING FILES FOR 'RES' OR 'SQUASH'
33% IN *SYSTEMDIRECTORY/001 ON DEV
3212 12:53 PK502 DATA MOVED IN *SYSDIR/1
3212 12:53 PK502 DATA MOVED IN (IPP)ALLIANCE/SYSDIR
3212 12:53 PK502 DATA MOVED IN *RELEASE/VERSIONS
```

If an AX QUIT is issued to JAMPACK (mix number 3210) i.e.

```
3210 AX QUIT
```


Then JAMPACK immediately

```
3212 12:54 PK502 DATA MOVED IN (IPP) "CCF.CON"  
3210 12:54 JAMPACK: GOING AWAY AFTER RESERVE COMPLETES  
3210 12:54 ACCEPT: '3210 AX QUIT' TO QUIT AND DS THE RESERVE  
3212 12:54 PK502 DATA MOVED IN *MAGUS/PDUMP  
3212 12:54 PK502 DATA MOVED IN *META/KEYS
```

If a second AX QUIT is now entered:

```
3212 12:54 PK502 DATA MOVED IN (BOB) "NEWMAGUS.CON"  
3212 12:54 PK502 DATA MOVED IN (BOB) "MACHINES.TXT"  
3212 12:54 PK502 RES, XD, OR SQUASH HAS BEEN DS'ED  
3210 12:54 PK502 *BADDISK/FMLYINX1/UNIT502/AD0C98BH  
NOT REMOVED (NOT PRESENT) ON DEV
```

The RESERVEDISK has been DS-ed and JAMPACK terminates.

In a SECTORS REQUIRED mode, JAMPACK will display an informational message about the maximum area created by the run:

```
1234 JAMPACK:AN AREA OF 399292 SECTORS MADE ON DEV
```

File Exceptions

During JAMPACK analysis, there are certain files that JAMPACK will not consider moving. These include temporary files, running job code, certain system files and most in-use files (unless the file is an active code file). Such files will not be moved by RESERVEDISK so they cannot be considered in pack space optimisation.

Occasionally, though, RESERVEDISK will encounter areas, requested by JAMPACK that are blocked by the MCP for some reason usually because the file owning the selected disk rows has become in-use since JAMPACK's analysis. In such cases, JAMPACK will automatically DS the waiting RESERVEDISK and continue to the next analysis.

JAMPACK messages

This section documents some of the various errors and information displays which may be generated during a JAMPACK run.

Error messages

These usually indicate an internal program fault or a system problem of some kind.

JAMPACK will usually terminate after issuing these error messages.

FLAT READ ERROR

I/O ERROR IN FLAT READ

All the above indicate a serious problem trying to read *SYSTEMDIRECTORY/001 on the target family to retrieve file information. This possibly indicates system hardware errors or a corrupt directory; it is advisable to attempt a RB ON <family> so that the MCP can check the directory. It is not possible for JAMPACK to continue.

VANISHING JOBDESC FILE: *JOBDESC ON DLFAMILY

If a pending DL JOBS assignment has been made i.e. the DL JOBS setting has been changed but no halt-load has been performed, then if JAMPACK is running against the new DL JOBS family and needs to process the job description file (JOBDESC), the above message is displayed.

INCOMPATIBLE UINFO STRUCTURE

This message indicates a probable incompatibility between JAMPACK and the current MCP level. It is likely that old, unqualified Metalogic software is running on a newer MCP; you should contact Metalogic for further assistance.

UNREADABLE SYSTEMDIRECTORY DUE TO UNKNOWN BLOCKING

The SYSTEMDIRECTORY is inaccessible because the file blocking is not recognized (should be BLOCKSIZE=30 or BLOCKSIZE=60 for VSS-1 disk).

I/O ERROR IN JOBDESC READ

JAMPACK has special handling for any JOBDESC file found on the target family; any I/O error reading that file will cause JAMPACK to terminate.

JAMPACK NOT INSTALLED - CALL METALOGIC

Metalogic licence keys for JAMPACK have expired or are not installed. Please contact Metalogic for assistance.

ALREADY RUNNING, THAW MAGUS IF JAMPACK WAS DS'ED

Only one JAMPACK can be running at any time; if a JAMPACK has been DS-ed, it is possible for the MAGUS library to believe that it has not been terminated. If another copy of JAMPACK runs in these circumstances, the above error will occur.

It is necessary to THAW the MAGUS library and restart all other Metalogic software. Any old copies of MAGUS should be terminated before running JAMPACK.

JAMPACK OLDER THAN MAGUS - COPY & SL MAGUS

This error may occur if an old JAMPACK codefile runs against a new MAGUS; this may occur because MAGUS and JAMPACK are very closely integrated and it may not be possible to run older versions.

SCHEDULE REASON IS LOG, JAMPACK ABORTED

JAMPACK will automatically FS any RESERVEDISK that is scheduled by the system – unless the schedule reason is down to disk space shortages on the DL LOG family. However, if the target family is the DL LOG family, JAMPACK will override this restriction and perform the FS.

NO RESERVE GREATER THAN DUMAX CAN BE DONE ON <family>

The value specified as a target is greater than the >current, available space as seen in a DU ON <family> response. No further action by JAMPACK is possible.

MAXIMUM NUMBER OF ERROR RETRIES EXCEEDED

This indicates that the retry limit (3) for optimising a simulation request has been reached. Typically, this message appears if the target sectors, or its optimised value, cannot be attained.

Warning and Information messages

PACK IS ALREADY FULLY SQUASHED

This message is displayed if JAMPACK is unable to optimise the pack family any further, that is, the largest contiguous area is very close to that of the maximum area (according to a DU).

**IO ERROR IN SEGMENTED HEADER READ
SHORT BLOCK ON LAST JOBDESC READ
LAST JOB HDR NOT COMPLETE
SHORT BLOCK ON LAST DIRECTORY READ
LAST HDR NOT COMPLETE**

All the above errors indicate a problem with the handling of segmented headers in the directory or job headers in a JOBDESC file. This may be indicative of an incompatibility with a new MCP or a corrupted directory/JOBDESC.

PROCESSING OBSOLETE JOBDESC FILE: *JOBDESC ON OLDLJOBFAMILY

Similar to the 'vanishing' JOBDESC described above, if JAMPACK finds an old JOBDESC file that is not in-use by the system, this message may be seen.

NO HOLE OF xxxx SECTORS

When doing a simulation run (i.e. with a "< target" specification), it may be likely that JAMPACK will not be able to find a suitable area to reserve, in his first pass"

NO xxxx SECTOR RESERVE

Followed by:

**SEARCHING FOR NEW TARGET SIZE
OPTIMISING TARGET MOVE**

This process may be repeated as JAMPACK attempts to find an area of disk that can be reserves to best satisfy the original request. This optimisation process continues until a search of all the available disk areas has been exhausted.

Running JAMPACK from SUPERVISOR

User's of Metalogic's SUPERVISOR may run JAMPACK automatically by using example supplied OPAL programs. These are always available within the release container supplied by Metalogic and automatically loaded at installation time. These Opals allow Supervisor to detect 'SECTORS REQUIRED' waiting entries and to monitor pack checkerboarding as well as other automatic exception handling.

These standard OPALS are included in the Metalogic release container held in the symbolic file: found in the file, OPALS/SUPERVISOR/EXAMPLES,

All OPAL examples are automatically loaded during installation. The OPALs within that file are called EX_JAMCONTROL and may be loaded /invoked directly from a SUPERVISOR COMS window or ODT:

**TT ENTER FROM OPALS/SUPERVISOR/EXAMPLES
TT WHEN EX_JAMCONTROL DO EX_JAMCONTROL**

The WHEN will monitor for all occurrences of any waiting entry with "SECTORS REQUIRED" displayed in the RSVP. SUPERVISOR will automatically invoke and monitor JAMPACK, taking away any need for operator involvement.

Please refer to the **Metalogic SUPERVISOR Reference Manual** for more information on the features of SUPERVISOR.

Running JAMPACK from WAITWATCHER

Similar to the mechanism employed by SUPERVISOR, the WAITWATCHER product (see [Waitwatcher](#)) can also monitor the occurrence of "SECTORS REQUIRED" waiting entries.

Upon detecting a waiting entry, WAITWATCHER can automatically invoke

JAMPACK to immediately handle the problem.

Further, WAITWATCHER has the added capability of placing large, dummy file, called *GOODISK, onto specific families where disk shortages may be prevalent. When a SECTORS REQUIRED is seen, WAITWATCHER can remove this file as a “quick fix” especially where JAMPACK may be unable to resolve the problem. This feature is particularly recommended for the DL families, especially DL OVERLAY.

Summary

JAMPACK increases the effective size of a pack, by reducing the impact of checkerboarding, and it can eliminate many of the throughput problems associated with 'SECTORS REQUIRED' hangs. JAMPACK requires no software patches, and has been tested on a wide variety of systems and different MCP levels.

JAMPACK has been successfully run on a system with 110 packs on-line, and on a halt-load family with nine members. METALOGIC/SUPERVISOR & FLEX products can be used to further improve automated operation, and pack utilisation.

WAITWATCHER utility

Introduction

WAITWATCHER is a simple, continuously running program that monitors system waiting mix entries and takes appropriate action for certain, common RSVPs. In its current implementation, WAITWATCHER specifically detects all variants of the following:

NO FILE
REQUIRES PK
SECTORS REQUIRED

During normal operation, WAITWATCHER wakes up every 12 seconds and checks the waiting entries. If a 'SECTORS REQUIRED' entry is detected, for a FAMILY that is not marked as 'ignored' (see AX IGNORE), the JAMPACK utility will be run. If JAMPACK is unavailable, then WAITWATCHER will execute the Unisys SQUASH ODT command.

If a 'NO FILE' or 'REQUIRES PK' condition is found, WAITWATCHER will take actions determined by the settings of the internal NOFILEDS and FAMILY options.

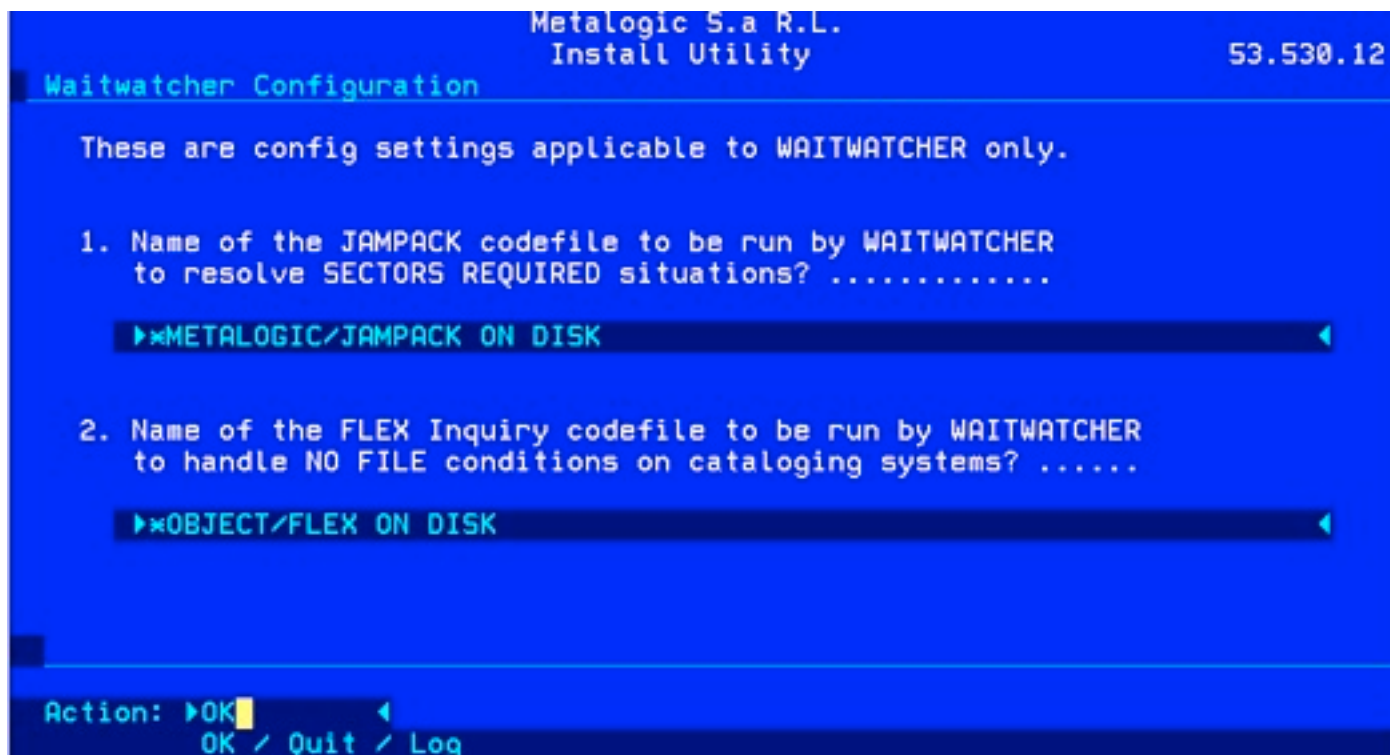
On CATALOGING systems, if the RSVP is a "NO FILE, FIND ON" and the Metalogic FLEX Inquiry utility is available, FLEX Inquiry will be run to perform a LOAD to bring back the non-resident file from the appropriate backup media.

WAITWATCHER assumes that the JAMPACK codefile is called *METALOGIC/JAMPACK and FLEX is called *OBJECT/FLEX, and that they are on the same pack as the WAITWATCHER code file (these defaults can be changed during the installation process).

WAITWATCHER also has two reports which can be requested using an 'AX' message. 'FP' gives a summary of the status of the disk pack subsystem, and 'HO' gives a report on the files open on a family. Both these commands are similar to the operator variants available to the Metalogic SUPERVISOR product.

WAITWATCHER configuration

During the first installation, if Metalogic's FLEX Inquiry and JAMPACK utility programs are available, the INSTALL utility will request confirmation of each code file title.

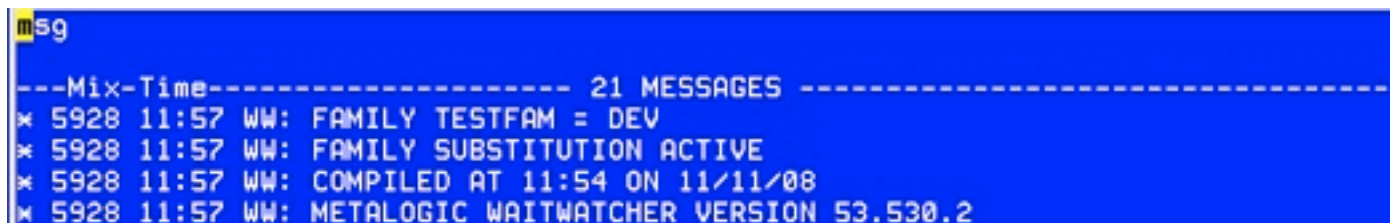


Running WAITWATCHER

The program should be run by from the ODT by:

```
??RUN *METALOGIC/SUPERVISOR/WAITWATCHER
```

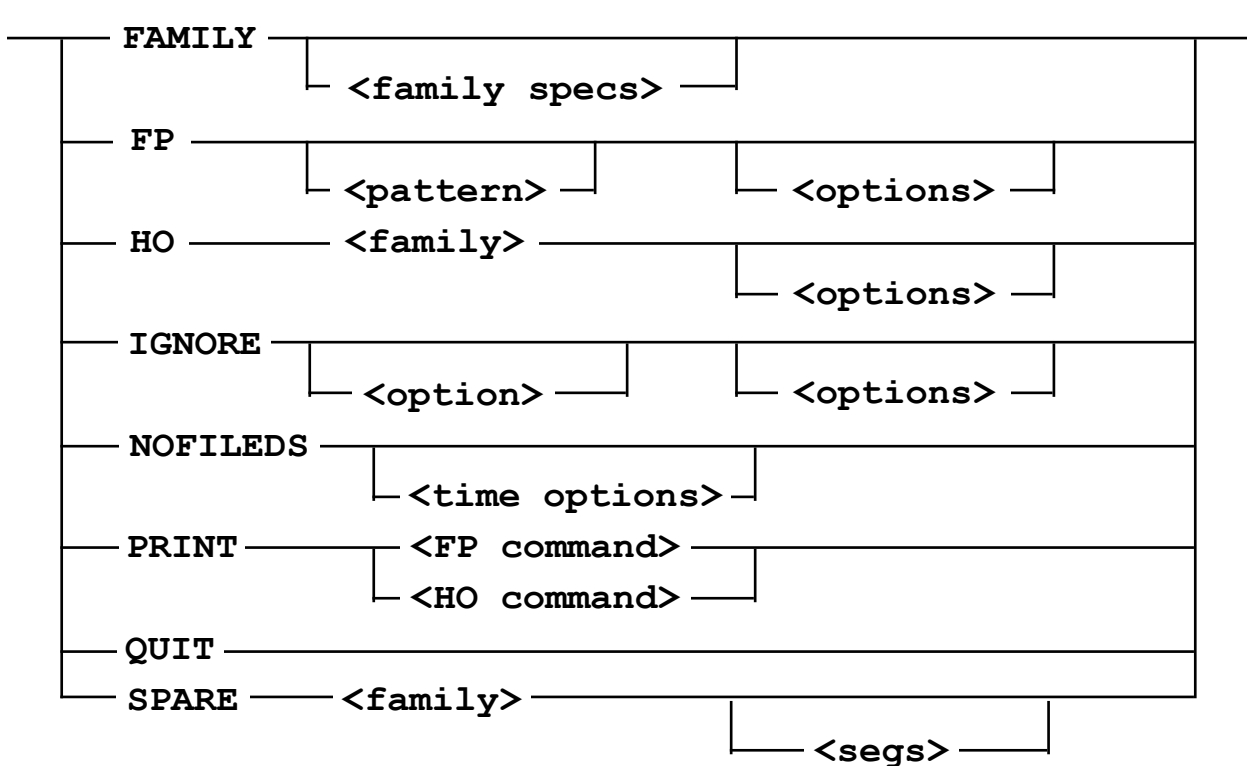
The following might be a typical series of displays:



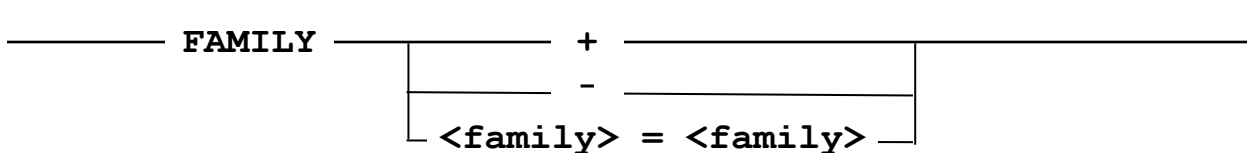
Operator interface

In the current implementation, WAITWATCHER commands are issued as operator AX actions to the active WAITWATCHER program.

The responses from these AX commands appear as system display messages. The command subset is shown below:



FAMILY



The FAMILY command allows automatic re-direction of 'NO FILE' and 'REQUIRES PK' messages for a chosen family, providing an extra level beyond that normally available from family substitution in the MCP.

FAMILY is useful when a pack that is present is unavailable. The command may also be used to move files around the pack subsystem depending on where space is available.

For example, a file, say BIGFILE, might be too large to be kept on-line all the time. If the application that uses this file has a non-existent pack name built-in, say PSEUDOPK, then the operators could load the BIGF file by entering a 'FP' command and finding the current most suitable pack, for example: EMPTYPK.

The operator could then enter the following AX to WAITWATCHER:

```
<mixno>AX FAMILY PSEUDOPK = EMPTYPK
```

WAITWATCHER will respond with:

```
#<mixno> WW:FAMILY SUBSTITUTION ACTIVE  
#<mixno> WW:FAMILY PSEUDOPACK = EMPTYPK
```

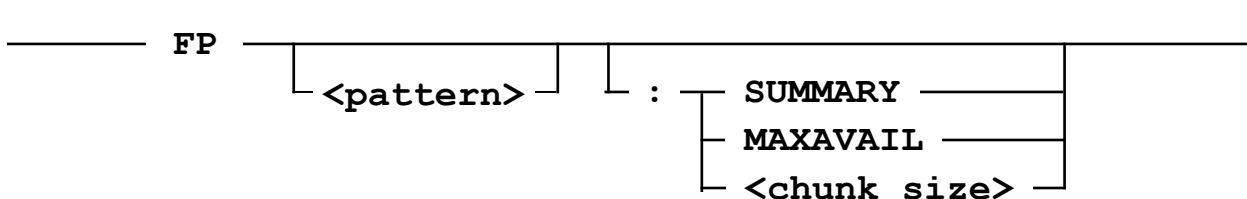
where<mixno> is the mix number of WAITWATCHER.

Any NO FILE or REQUIRES PK waiting entries that request the family PSEUDOPK will be automatically substituted with the family EMPTYPK.

The substitution can be suspended by using the command 'FAMILY -', and WAITWATCHER will retain the setting. The setting can later be restored by the 'FAMILY +' command. The FAMILY settings are preserved across Halt/Loads and program restarts.

The specification in effect, when a 'FAMILY -' is actioned, will NOT be remembered in subsequent program executions. Only one FAMILY substitution may be in force at any time.

FP



FP (File Placement) gives a report on the status of the entire pack subsystem to assist with the problem of where large files can be placed. Much of the information in the report is not easily obtained by conventional operator commands.

Each disk unit has a line in the display messages, and for multi-pack families each family has an additional 'totals' line. By default, output appears as a series of display messages unless the PRINT modifier is used, when the output will divert to a printer backup file.

By default, the report shows all disks, with a summary totals line for each family, and includes the numbers of sectors in available areas more than 504 sectors. By specifying a wildcard <pattern>, the report can be restricted to include specific families.

The SUMMARY modifier restricts the report to show only summary information for each family.

The <chunk size> modifier allows the report to show available disk area information using an area size other than 504 (same as the default in a DU response). The MAXAVAIL modifier will report the largest, contiguous available area for each disk

pack.

A typical response to FP might be:

```
MSG
---Mix-Time----- 21 MESSAGES -----
* 5928 10:13 WW: ----- Disk Status Summary -----
* 5928 10:13 WW: Family Name  DiskSize TotAvail      >504    Reads    Writes Mbytes
* 5928 10:13 WW: CDIMAGE      24437686 16692446    16630K    42468     54725    1813
* 5928 10:13 WW: DEV          24111451 11884472    11534K    100291    128861    3704
* 5928 10:13 WW: DISK         23301733 7999315     7354845   306715    214030    5667
* 5928 10:13 WW: PRIVATE      58331     54135      54121     12         0         0
* 5928 10:13 WW: SPARE        23301733 3345216     3247881   271660    46096     10060
* 5928 10:13 WW: Family Name  DiskSize TotAvail      >504    Reads    Writes Mbytes
* 5928 10:13 WW: ----- Disk Status Summary -----
```

The reported items are

SEGS	The total number of segments. This can be useful in determining percentage availabilities etc.
DU	The amount of available space. By juxtaposing the DU for all packs, it can be used to give a quick answer to the question of where a large file can be placed. The + or - which can follow the figure indicates the trend since the last FP request. This allows the trend of usage on the pack to be monitored easily.
DUMAX	The largest area of available space.
READS	The number of reads since the last halt/load or UR-.
WRITES	The number of writes since the last halt/load or UR-.
MBYTES	The number of megabytes of data in I/Os since the last halt load. In combination with the two preceding numbers, this can give a good picture of the usage of the packs and facilitate balancing. They also give an insight into the possible competition from other I/Os depending on where the file is placed.

To print a standard report:

PRINT FP

To display a report of all families, whose label starts with 'WORK' and show space greater than 1000 sectors:

FP WORK= : >1000

To show a family summary report:

FP : SUMMARY

HO

_____ HO _____ <family> _____
 └─ # <inx> ─┘

The HO (Headers Open) command reports on disk file headers currently open on the specified family, including both temporary and job files (if the DL JOBS family). The <family> specification must be an available, on-line, disk pack family.

If the "#" variant is used for a multi-member disk family, the report is restricted to those disk file headers with rows active on the specified family index.

This command has two main uses. Firstly, if a program goes into an infinite loop writing to a temporary file, the programs using the pack which contains that file, may start to get "SECTORS REQUIRED" RSVP's. This command can be used to identify which file is causing the problem.

Secondly, it is sometimes necessary to CLOSE a pack for maintenance reasons. The MCP will not allow the disk to be closed unless there are no in-use files open (the *SYSTEMDIRECTORY/001 file is **always** open). HO can help identify which files need to be released to CLOSE the pack.

A typical response might look like:

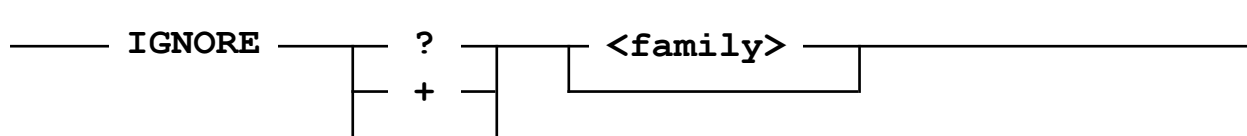
```
MSG
---Mix-Time----- 21 MESSAGES -----
* 5928 10:23 WW:      *DNDISK (1164800)
* 5928 10:23 WW:      *SYSTEMDIRECTORY/001 (14000)
* 5928 10:23 WW:      *METALOGIC/SUPERVISOR/LOG (1530)
* 5928 10:23 WW:      *METALOGIC/TRIM/LOG (6630)
* 5928 10:23 WW:      *METALOGIC/MAILLIB/LOG (7140)
* 5928 10:23 WW: REPORT OF HEADERS OPEN ON CDIMAGE
```

The 'T' marker, prior to the file title, would indicate a temporary file; 'J' a job or session whereas no marker means the header is a permanent file. The parenthesised figure after the name indicates the total amount of sectors occupied by the file.

The SYSTEMDIRECTORY/001 file will always appear if one exists, but it does not count in the open count given in the PER PK ODT command.

Since the release of MCP 3.9 and above, the HO does NOT return the mixnumber of the task that originally caused the disk file header to be loaded into memory. The reason for this omission is that this attribute is not easily retrievable any more and too much overhead would be needed to extract it. Further, the Unisys equivalent command, SHOWOPEN, does provide this information but takes a considerably longer time to get it.

IGNORE



The IGNORE command is used to maintain a list of families where SECTORS REQUIRED' waiting entries should be ignored and, therefore, excluded from action by JAMPACK. This list is preserved across Halt/Loads and program restarts.

The 'IGNORE ?' variant interrogates the current list of families for which JAMPACK will not be invoked.

'IGNORE +' or 'IGNORE +' are used to add families to the list.

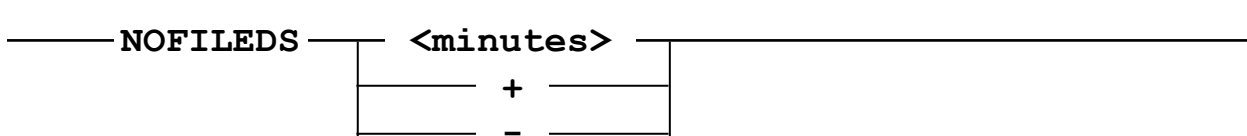
'IGNORE -' will delete individual families from the list.

For example:

```
3027AX IGNORE + DISK
#3027 WW: FAMILY DISK ADDED TO IGNORE LIST

3027AX IGNORE ?
#3027 WW: IGNORED FAMILIES
#3027 WW: DISK
#3027 WW: IGNORED FAMILIES
```

NOFILEDS



Programs waiting on 'NO FILE' or 'REQUIRES PK' RSVPs are quite a common annoyance to operations. They are in fact relatively expensive in resource consumption, taking memory, processor and I/O time as the MCP periodically re-evaluates the condition. Operators are wary of DS-ing such programs because they feel they may be blamed

Whereas many users would object to operations DS-ing their programs they will often accept it if it is automatic. As a result numerous sites have implemented automatic DS methods, either by writing programs to do it, or by patching the MCP. Most users of Metalogic's SUPERVISOR use an OPAL program to enforce the DS. The NOFILEDS command in WAITWATCHER is a simplified version of that OPAL. The 'NOFILEDS' setting is retained across Halt/Loads and program restarts.

When NOFILEDS is active, any program waiting on a DISK or PACK file with a NO FILE or REQUIRES PK RSVP will be DS-ed after the specified number of minutes if the RSVP is not responded to.

There are two main exceptions. If the file is a non-resident file on a Cataloging system, WAITWATCHER will try to use FLEX Inquiry to LOAD the file. Also, if the family requested is the one specified by the FAMILY command, the FAMILY substitution will be applied first.

The <minutes> modifier is the number of minutes WAITWATCHER will wait before DS-ing the offender. It must be greater than or equal to 0. The '-' modifier turns off NOFILEDS, '+' turns it on with the previous setting, and NOFILEDS on its own gives a display of the current state of the option.

```
3027AX NOFILEDS
```

```
#3027 WW: NOFILEDS DELAY IS 5 MINUTES
```

When an eligible NO FILE is seen, WAITWATCHER attaches a message on the waiting entry to indicate that the mixnumber will be DS-ed e.g.

```
#3035 NO FILE ZZZ ON DEV (PK)
```

```
#3035 WW: WILL BE DSED @ 1450 FOR NO FILE
```

After 5 minutes, the waiting entry will be DS-ed:

```
#3035 WW: DSED FOR NO FILE
#3035 GOING
#3035 OPERATOR DSED @ (06488090)
#O-DS @ 06488090, 08966000, 04477000, 06170000, 02576000, 90000375.
#ET=4:37.6 PT=0.0 IO=0.0
```

QUIT

— QUIT —————|

QUIT causes an orderly end of task at the next cycle. This is preferred to DS-ing the program.

SPARE

— SPARE — <family> —————|
 |—————|
 <segments> ———

WAITWATCHER and JAMPACK cannot fix certain SECTORS REQUIRED hangs. In particular, those involving MCP processes such as OLAYSCOUT, CHANGEMCP or RESERVEDISK cannot be programmed.

Because of this, WAITWATCHER automatically keeps an amount of “spare” disk space on packs that have a DL OLAY on them. The file is called *GOODDISK and has an odd FILEKIND (123). A GOODDISK file can be placed on other families manually using the SPARE command. If a GOODDISK file is already resident on the pack, the command will be ignored.

If WAITWATCHER gets into a situation where JAMPACK or SQUASH cannot handle the SECTORS REQUIRED, it will remove the GOODDISK file, thus providing some time to decide what further action can be taken. The list of families that have had spare areas created on them (but not the size of the area) is preserved across Halt/Loads and program restarts.

The optional <segments> parameter allows the size of the GOODDISK file to be defined (the default value is 3024, maximum value is 999999).

```
3036AX SPARE DISK
#3036 WW: FAMILY DISK ADDED TO SPARE LIST
```

When WAITWATCHER is required to remove a *GOODISK file because of a SECTORS REQUIRED situation on DL OVERLAY or SPARE family:

#3036 WW: EMERGENCY SPACE REMOVED ON DISK

Copyright

Copyright © 1979-2009 Metalogic S.à R.L., Luxembourg.
All rights reserved.

Last revision: November 11 2009

METALOGIC believes that the information described in this manual is accurate and reliable, and much care has been taken in its preparation. However, no responsibility, financial or otherwise, is accepted for any consequences arising out of the use of this material.

The information contained herein is subject to change. Revisions may be used to advise of such changes and/or additions.

ACKNOWLEDGEMENTS

To John Hurley, for conceiving FLEX Inquiry and backing the idea with 3 years of hard work.

To the other implementers: Bob, Brian, David, Don, and Jeff.

As always, some of the best ideas did not come from the implementers. To all of you who contributed, many thanks.

METALOGIC S.à R.L.
113, Rue de Luxembourg
L-7540 Rollingen
Luxembourg

<http://www.metalogic.eu.com>