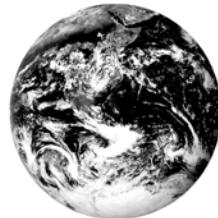


# FILEMANAGER

## FileManager Operations Guide for Unisys MCP Systems

Release 8.068E

September 2009



**dsi** dynamic  
solutions  
international.

Copyright

This document is protected by Federal Copyright Law. It may not be reproduced, transcribed, copied, or duplicated by any means to or from any media, magnetic or otherwise without the express written permission of **DYNAMIC SOLUTIONS INTERNATIONAL, INC.**

It is believed that the information contained in this manual is accurate and reliable, and much care has been taken in its preparation. However, no responsibility, financial or otherwise, can be accepted for any consequence arising out of the use of this material. **THERE ARE NO WARRANTIES WHICH EXTEND BEYOND THE PROGRAM SPECIFICATION.**

Correspondence regarding this document should be addressed to:  
Dynamic Solutions International, Inc.  
Product Development Group  
1 Inverness Drive East, Englewood, Colorado 80112  
(800)641-5215 or (303)754-2000  
Technical Support Hot-Line (800)332-9020  
E-Mail: support@dynamic solutions.com

# Contents

.....

|  |           |
|--|-----------|
| <b>OVERVIEW .....</b>                      | <b>1</b>  |
| <b>FILEMANAGER CONSIDERATIONS .....</b>    | <b>3</b>  |
| FileManager File Tracking .....            | 3         |
| File Recovery .....                        | 3         |
| FileManager File Selection .....           | 3         |
| <b>INSTALLING FILEMANAGER .....</b>        | <b>5</b>  |
| MARC Installation.....                     | 7         |
| Graphical User Interface Installation..... | 7         |
| Upgrading FileManager Software .....       | 8         |
| <b>ACCESSING FILEMANAGER .....</b>         | <b>9</b>  |
| TMREMOTESPO Interface .....                | 9         |
| MARC Command Interface .....               | 9         |
| Graphical User Interface.....              | 10        |
| <b>FILEMANAGER CONFIGURATION .....</b>     | <b>11</b> |
| Convention Configuration .....             | 13        |
| Database Configuration.....                | 15        |
| E-Mail Configuration .....                 | 17        |
| Report Configuration .....                 | 19        |
| Set Configuration .....                    | 21        |
| Backup Configuration.....                  | 23        |
| <b>FILEMANAGER SECURITY.....</b>           | <b>25</b> |
| <b>FILEMANAGER OPERATIONS.....</b>         | <b>30</b> |

|                                  |           |
|----------------------------------|-----------|
| DATABASE CLOSE Command .....     | 30        |
| DATABASE OPEN Command.....       | 31        |
| DATABASE BACKUP Command .....    | 32        |
| DATABASE REORGANIZE Command..... | 34        |
| DATABASE RESTORE Command .....   | 35        |
| DATABASE RECOVER Command.....    | 36        |
| DATABASE Command.....            | 37        |
| BACKUP Command.....              | 38        |
| RESTORE Command.....             | 41        |
| TL (Transfer Log) Command .....  | 43        |
| LOAD Command.....                | 44        |
| QUIT Command .....               | 45        |
| EMAIL Command.....               | 46        |
| WAIT Command.....                | 48        |
| <b>FILEMANAGER REPORTS .....</b> | <b>51</b> |
| OLDEST Command .....             | 51        |
| NEWEST Command.....              | 52        |
| ALL Command .....                | 53        |
| LAST Command.....                | 54        |
| REPORT and LIST Commands.....    | 55        |
| ALL Files Report.....            | 58        |
| File Activity Report.....        | 59        |
| File Exceptions Report.....      | 60        |
| Maintenance Report .....         | 61        |
| Log Report .....                 | 62        |

|  |            |
|--|------------|
| Reorganize Report.....                   | 65         |
| Summary Report.....                      | 66         |
| Audit Report.....                        | 67         |
| Ad Hoc Reports .....                     | 68         |
| STATUS Command.....                      | 71         |
| FIND Command.....                        | 72         |
| VERSION Command.....                     | 75         |
| <b>FILE DATABASE MAINTENANCE .....</b>   | <b>77</b>  |
| ADD Command .....                        | 77         |
| DELETE Command.....                      | 78         |
| MODIFY Command .....                     | 80         |
| <b>FILEMANAGER MACROS.....</b>           | <b>83</b>  |
| <b>BATCH MODE EXECUTION.....</b>         | <b>91</b>  |
| <b>DIAGNOSTIC COMMANDS .....</b>         | <b>95</b>  |
| LOG Command .....                        | 95         |
| DEBUG Command.....                       | 96         |
| TRACE Command.....                       | 100        |
| RESCUE Command.....                      | 101        |
| ABORT Command .....                      | 102        |
| <b>COMMON SYNTAX ELEMENTS .....</b>      | <b>103</b> |
| <b>FILEMANAGER DATABASE FIELDS .....</b> | <b>108</b> |
| <b>PROGRAMMATIC INTERFACES .....</b>     | <b>113</b> |
| <b>GLOSSARY.....</b>                     | <b>115</b> |
| <b>INDEX.....</b>                        | <b>119</b> |



# Chapter 1

## Overview

FileManager from *Dynamic Solutions International (DSI)*, is an add-on component to the DSI TapeManager product. While TapeManager tracks and manages tape media, FileManager tracks and manages files on the media.

FileManager uses the Unisys MCP Library/Maintenance utility and the LIBMAINTDIR feature to initiate backup or restore operations. Files can be backed up directly, or grouped by logical user-defined groups, called *sets*. The system administrator can create sets that span multiple usercodes across multiple families to create a logical backup strategy. FileManager can also be used to find multiple generations of files, and create reports on specific file information. Although FileManager requires TapeManager, FileManager is a free-standing library from TapeManager, and does not interfere with tape management operations. FileManager operations are controlled through a PC program GUI interface. FileManager commands also can be entered through a utility program that can either be run from a CANDE session, MARC session, or configured as a COMS program. There is no need to look up job or mix numbers. FileManager also includes a utility program that provides for batch-level processing via standard A-Series Work Flow Language (WFL).

### Features

- The FileManager database is updated in real-time as backup activity occurs.
- Commands may be input through **MARC**, batch, **TMREMOTESPO**, and **GUI**.
- MCP standard backup tapes are used, not a proprietary format.
- Reports are provided either in hardcopy and/or file format.
- Extensive searching capabilities locate files.
- Searches on strings can use "wildcards".
- Customization of each site's requirements is provided.
- Logging and reporting of exception activity is standard.
- The command interface provides for quick and easy entry of all functions.

### Requirements

- TapeManager Level 68E or higher.

## Overview

---

- 51.1 MCP release or later required.
- A Windows PC for the GUI interface program.
- MT or ET equivalent terminal for utility and TMREMOTESPO program.

# Chapter 2

## FileManager Considerations

FileManager tracks files by tape serial number and relies on TapeManager in order to restore files from backups. Before using FileManager, consider the following items in FileManager implementation.

### FileManager File Tracking

FileManager only tracks files backup by FileManager initiated processes. Files backed by other WFL or operator commands will not be tracked in the FileManager database and therefore not recoverable by FileManager.

### File Recovery

All FileManager backup tapes are created using MCP standard Library Maintenance (COPY) operations. These Library Maintenance tapes are all created using the LIBMAINTDIR option. The advantage of these features is that in a Disaster Recovery (DR) situation all files may be found and recovered without having FileManager installed.

If FileManager is installed at the DR site but the primary FileManager database is not available, the essential information about the files on the backup tapes may be recovered by doing the following steps.

1. For each FileManager backup tape use the WFL CREATE LIBMAINTDIR command to create the LIBMAINTDIR disk file for that tape.
2. Use the FileManager LOAD LMDIR command to read the above LIBMAINTDIR files and repopulate the FileManager database.

### FileManager File Selection

When specifying files to be backed up, avoid using direct file names without a usercode or pack family specification. This forces FileManager to search every pack on the system, looking for a matching file specification.



# Chapter 3

## Installing FileManager

If the file and run-time keys for FileManager are present when the TapeManager SYSTEM/TAPEMANAGER/INSTALL program is run, the FileManager system will be installed automatically at the same time as the TapeManager system is installed. FileManager will be installed in the same location (family and user directory) as the TapeManager software.

FileManager is able to exist on a different pack family and user directory than TapeManager. If a different location for FileManager is desired copy, the SYSTEM/FILEMANAGER/SUPPORT file to that location before the software is started for the first time. Then do the SL and MP commands as described below.

The following files are supplied with the FileManager system:

### **SYSTEM/FILEMANAGER/SUPPORT**

This file is a library that is the core module of the FileManager system. It must be available at all times. The library should be active whenever there is the possibility of backup activity. This library must be SLed as FILEMANAGER with the library attributes of ONEONLY, TRUSTED, and LINKCLASS=1.

Example: SL FILEMANAGER = SYSTEM/FILEMANAGER/SUPPORT ON <pack name>:TRUSTED, ONEONLY, LINKCLASS=1. The library code file is supplied with the PU and IDENTITY MP commands applied to it. (The library must be MP + PU at a minimum.)

FileManager is also setup to Auto Initiate when the MCP host is booted as follows.  
AI DSIFILEMANAGER = <usercode>SYSTEM/FILEMANAGER/SUPPORT ON  
<family name>

### **SYSTEM/TAPEMANAGER/UTILITY**

This program provides a batch interface to the FileManager system. Setting SW1 to true redirects a set of batch commands from FileManager to FileManager.

### **SYSTEM/TMREMOTESPO**

This program can be used to dynamically enter commands to the FileManager system by providing the FM command. The program may be run from a CANDE or MARC session, or it can be configured in COMS as a program and window for easy access through MARC.

## Installing FileManager

---

*Note:* All the above files are supplied as system files (non-usercoded) with a security of PUBLIC. The security must be changed if access to the system is to be restricted.

When FileManager is first activated, it will not find an existing database. Entering <mix #> AX OF will create an empty database to be populated by future FileManager actions. These files comprise the FileManager database:

SYSTEM/FILEMANAGER/CONTROL  
SYSTEM/FILEMANAGER/DB/DATA  
SYSTEM/FILEMANAGER/DB/DATA/FNINDEX  
SYSTEM/FILEMANAGER/DB/DATA/SNINDEX  
SYSTEM/FILEMANAGER/DB/DATA/TSINDEX  
SYSTEM/FILEMANAGER/DB/DATA/IDINDEX  
SYSTEM/FILEMANAGER/DB/DATA2  
SYSTEM/FILEMANAGER/SETS  
SYSTEM/FILEMANAGER/LOG

*Note:* Some of the files require the KEYEDIOII software to be installed before they may be accessed.

*Note:* Some of the above files may or may not exist depending on configuration options. Other files may also be created in the course of FileManager usage.

### MARC Installation

For the FileManager command interface to be available through MARC, the following command must be entered at a MARC screen:

```
DIRECTIVE + FM = <usercode>SYSTEM/DSISUPPORT ON <familyname>:SYSTEM
```

Example: DIRECTIVE + FM = (USER)SYSTEM/DSISUPPORT ON PACK:SYSTEM

The <usercode> and <familyname> specification must be the usercode and family where the FileManager software is installed. The :SYSTEM syntax restricts the use of FileManager commands to those users with SYSTEMUSER privileges. This option may be dropped or changed depending on the sites' security requirements. (See the A Series Menu-Assisted Resource Control (MARC) Operations Guide.)

*Note: The MARC Directive interface is not protected against program faults and operator actions as are some MCP interfaces. While FileManager attempts to protect the interface, it can not do so in all cases, such as an operator DS of the FileManager library. These kinds of faults/actions may lead to the DSing or hanging of the MARC processes which could lead to the loss of control of COMS until COMS is restarted.*

*Note: Implementing the MARC Directive of FM will override the existing FM MARC command. From MARC, the FM command is intended to modify an existing mix number to redirect a print job to the proper FORMID, or force a print job to print on a non-formed printer. The user should review their print operations that are modified via MARC before considering replacing the MARC FM directive.*

### Graphical User Interface Installation

The FileManager Graphical User Interface (GUI) is accessed through the TapeManager GUI, and can be installed at any user workstation. The GUI is a Microsoft Windows based application. The GUI can be installed by executing the GUI\_SETUP.EXE program which can be found in the TAPEMANAGER share of the MCP that was created when TapeManager was installed.

An MCP host component of the GUI, SYSTEM/TAPEMANAGER/GUIINTERFACE, is installed during the normal TapeManager/FileManager installation. A copy of the GUIInterface is started for each workstation that executes the TapeManager GUI.

### Upgrading FileManager Software

The FileManager software will periodically need to be upgraded as enhancements and corrections become available.

1. Backup all FileManager programs and data files to tape or another disk family. The **FM DB BACKUP** command may be used for this.
2. Mount the software release CD in a system CDROM drive unit, or unwrap the downloaded update.
3. Copy the new SYSTEM/FILEMANAGER/SUPPORT code file to the original installation location via Library/Maintenance, overwriting the existing SYSTEM/FILEMANAGER/SUPPORT code file.
4. All FileManager and TMRemoteSPO sessions must be ended to complete the install. Using TMREMOTESPO, enter 'FM QUIT' to stop FileManager. Ensure that FileManager leaves the mix before proceeding. If FM was configured under MARC, enter DIRECTIVE – FM from a MARC screen to continue.
5. Once SYSTEM/FILEMANAGER/SUPPORT completes, the library has to be reinstated. Do this by entering the following at an ODT:  
SL – FILEMANAGER  
(the system should respond that FILEMANAGER is no longer established)  
SL FILEMANAGER = <usercode>SYSTEM/FILEMANAGER/SUPPORT ON  
<family name>: TRUSTED, ONEONLY, LINKCLASS=1  
(the system should respond that FILEMANAGER has been established).
6. Restart FileManager by starting TMRemoteSpo, and entering 'FM START'.
7. If MARC Directives are desired, enter on a MARC screen,  
DIRECTIVE + FM = <usercode>SYSTEM/DSISUPPORT ON <family  
name>:SYSTEM. Note this will replace the standard MARC FM command. Refer to the MARC INSTALLATION notes above.
8. Finally, verify that the FileManager is active by entering a command through TMRemoteSpo such as **FM STATUS**. You should receive the status of the program, which indicates that FileManager is now back on-line.

The FileManager update has now been completed.

# Chapter 4

## Accessing FileManager

Access to the FileManager system is provided via multiple interfaces.

### TMREMOTESPO Interface

The TapeManager software CD includes a TMREMOTESPO program. This program may be installed as a COMS program, run from a CANDE session, or run from a ??MARC session of an ODT. When using FileManager, all commands must be preceded by FM via the TMREMOTESPO interface.

The TapeManager release media includes a source file for an example of a TMREMOTESPO program that also processes FM commands. You are welcome to use this source file, but it is not supported by DSI. The TMREMOTESPO program may be setup in MARC as a window by following the steps similar to the ones outlined in COMS Installation in the previous chapter.

*Note: The MCP limits the number of characters that can be input via the ODT or REMOTESPO to approximately 110 characters for the ODT and 100 characters for a REMOTESPO. The actual number of characters allowed will vary depending on unit or station number.*

### MARC Command Interface

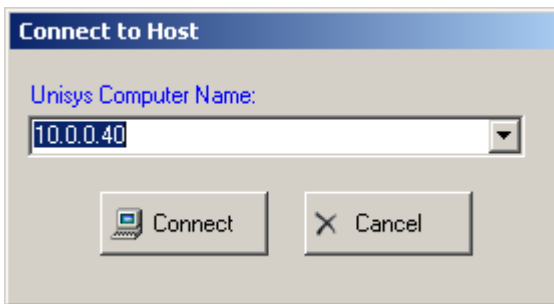
If MARC is configured as in the previous section, the FM commands may be entered on the action line of any MARC screen. Responses to these commands will be displayed by MARC in the same way as responses to other system commands. Note that setting an FM directive will override the existing FM (FORM) command through MARC.

# Graphical User Interface

A Graphical User Interface (GUI) is available for accessing FileManager functions. The GUI is installed on individual workstations. The GUI allows access to TapeManager and FileManager functions using menus and forms similar to other Windows based applications.



The initial form requests the name or IP address of an MCP host where TapeManager and FileManager are installed. Depending on the security setting of the host a user code and password may also be required.





## **FileManager Configuration**

---

### **CONFIGURE :FILE SAVED/FILEMANAGER/CONFIGURATION**

This option of the CONFIGURE :FILE command allows the configuration information to be placed in a file with a user specified name. The file must not already exist.

## Convention Configuration

The CONFIGURE CONVENTION command specifies the format to be used when specifying or reporting date and time values. Selecting a format will cause FileManager to use that format for all command input and output presentation.

### Syntax

```
FM — CONFIGure — CONVENTion —————
                               |
                               |— <date format spec> —|
                               |— <time format spec> —|
```

<date format spec>

```
— DATE ————— = ————
      |         |         |
      |         |         |— MM/DD/YY —|
      |         |         |— MM.DD.YY —|
      |         |         |— MM-DD-YY —|
      |         |         |— DD/MM/YY —|
      |         |         |— DD.MM.YY —|
      |         |         |— DD-MM-YY —|
      |         |         |— MM/DD/YYYY —|
      |         |         |— MM.DD.YYYY —|
      |         |         |— MM-DD-YYYY —|
      |         |         |— DD/MM/YYYY —|
      |         |         |— DD.MM.YYYY —|
      |         |         |— DD-MM-YYYY —|
      |         |         |— YYYY/MM/DD —|
      |         |         |— YYYY.MM.DD —|
      |         |         |— YYYY-MM-DD —|
```

<time format spec>

```
— TIME ————— = ————
      |         |         |
      |         |         |— HH:MM:SS —|
      |         |         |— HH.MM.SS —|
      |         |         |— HH-MM-SS —|
      |         |         |— HH:MM —|
      |         |         |— HH.MM —|
      |         |         |— HH-MM —|
```

### Explanation

#### CONFIGURE CONVENTION DATE FORMAT = DD.MM.YY

The DATE option specifies the format to be used when processing date values. For command input, this command specifies the format that is expected in any command where a date is required. For output, this command specifies the format that will be used in output displays and printed reports. The FORMAT syntax is provided for readability.

#### CONFIGURE CONVENTION TIME = HH:MM:SS

The TIME option specifies the format to be used when processing time values. For command input, this command specifies the format that is expected in any command where a time value is required. For output, this command specifies the format that will be used in output displays and printed reports. When a time format that includes the

## FileManager Configuration

---

seconds part is configured, the seconds values are only displayed where determined to be relevant. On command input, the seconds value may always be optionally specified regardless of the time format option. The FORMAT syntax is provided for readability.

### **Century Considerations**

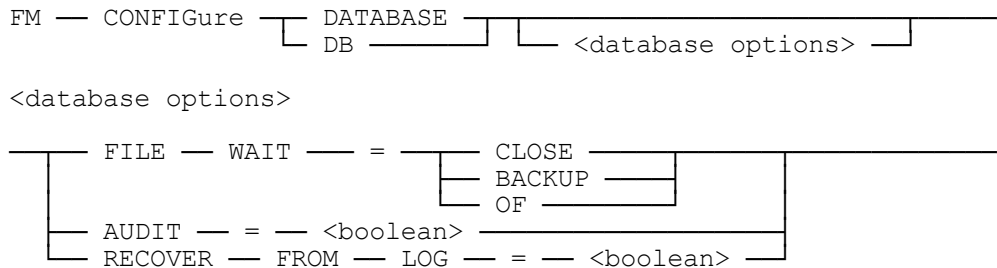
All date related fields within the FileManager database contain century information. The above convention specification allows a site to determine when (or if) the input and reporting features of FileManager require 4 digit values for year specification. While not required, it is recommended FileManager be configured for 4 digit years to eliminate any user confusion.

When 2 digit years are configured, FileManager must determine which century to which it refers. FileManager will use the century 1900 for any 2-digit years between 70 and 99 inclusive, otherwise the century 2000 is assumed.

## Database Configuration

The CONFIGURE DATABASE command specifies the option settings to be used with the FileManager database.

### Syntax



### Explanation

#### CONFIGURE DATABASE FILE WAIT = CLOSE

The CONFIGURE DATABASE FILE WAIT command specifies the action to be taken when FileManager detects that a Library Maintenance task is trying to copy the FileManager database. The FileManager database uses KEYEDIOII to manage the database, which cannot be copied while open. Library Maintenance will wait until a KEYEDIOII file set is closed or an operator OF before continuing.

The CLOSE option specifies that FileManager is to close its database so that Library Maintenance can copy it. While the database is closed, all commands or actions that require the database are rejected or suspended. Depending on the MCP level and copy options, Library Maintenance may hold the database files (prevent from re-opening) until the copy command has completed. The default value is false.

#### CONFIGURE DB FILE WAIT = BACKUP

The BACKUP option specifies that the FileManager should perform a DATABASE BACKUP TO DISK before allowing the copy to continue as with the CLOSE option.

#### CONFIGURE DB FILE WAIT = OF

The OF option specifies that FileManager will issue OF commands to the Library Maintenance task to prevent the copying of the database. This option also ensures that FileManager is kept available, but a DB BACKUP command should be performed regularly. This is the default if the database configuration option has never been specified.

## FileManager Configuration

---

### **CONFIGURE DB AUDIT = TRUE**

The AUDIT option specifies that the FileManager should create an audit trail of all accesses to the FileManager database. The audit trail is similar to those created by DMSII in that it contains both before and after images. The audit trail can be useful for diagnostics or to determine when a change was made. Unlike DMSII, this audit trail cannot be used for recovery purposes. The audit trail can be viewed by using the AUDIT report. The default value is false.

### **CONFIGURE DB RECOVER FROM LOG = TRUE**

The RECOVER FROM LOG option specifies that FileManager should attempt to find any system Sumlog records that it may have missed while the software was stopped. Only log records that cause database updates are processed. The default value is false.

### **CONFIGURE DB**

The CONFIGURE DATABASE command without any options will display the current settings of the DATABASE options.

## E-Mail Configuration

The CONFIGURE EMAIL command specifies how FileManager will use e-mail to send notices and perform support functions. To use the EMAIL function, the Unisys MCP E-Mail software (HMP 7.0/SSR 48.1 MCP or higher) must be installed. (Refer to the EMAIL Utility section of the *System Software Utilities Operations Reference Manual*.)

### Syntax

```
FM CONFIGure — EMAIL — [ <email options> | NONE ]
```

<email options>

```
[ <SUPPORT // <addresses> | SUPERvisor NONE | OPERator | SIGNATURE [ <quoted string> | IN <file file> | NONE ] ]
```

<addresses>

```
— <addr list> [ & <addr list> | & <addr list> ]
```

<addr list>

```
[ <email address> | SUPPORT | SUPERvisor | OPERator ]
```

### Explanation

#### CONFIG EMAIL SIGNATURE “Your DSI FileManager”

Establishes the string as the signature for any messages sent by FileManager. To use a file for a signature, refer to the EMAIL Utility section of the *System Software Utilities Operations Reference Manual*.

#### CONFIG EMAIL SUPERVISOR VAN.TU@MYCOMPANY.COM

Establishes the e-mail account for the supervisor. By default, the supervisor account is

## **FileManager Configuration**

---

**SUPERVISOR.** The supervisor receives all FileManager e-mail messages caused by alerts or faults.

### **CONFIG EMAIL SUPPORT SUPPORT@MYCOMPANY.COM && JOHNNY.ATLAS@MYCOMPANY.COM**

Establishes the e-mail account to receive blind courtesy copies for all e-mail messages sent to SUPPORT. The initial list of email addresses are the TO email list. Email addresses after the first & are the CC email list. Email addresses after the second & are the BCC email list. The // form of the command is used to separate parts of the command that might be confused as email addresses. By default, the support account is SUPPORT@DYNAMICSOLUTIONS.COM.

### **CONFIG EMAIL SUPPORT NONE**

Prevents all e-mails to SUPPORT.

### **CONFIG EMAIL NONE**

The NONE option clears all EMAIL configuration options.

### **CONFIGURE EMAIL**

The CONFIGURE EMAIL command without any options will display the current settings of the EMAIL options.



## **FileManager Configuration**

---

If the PAGESIZE attribute is specified, it will determine the number of lines to be printed on a page. The default value for PAGESIZE is 60 for the FileManager.

The ATTRIBUTES defined in this command may be added to or changed in the REPORT command.

### **CONFIGURE REPORT**

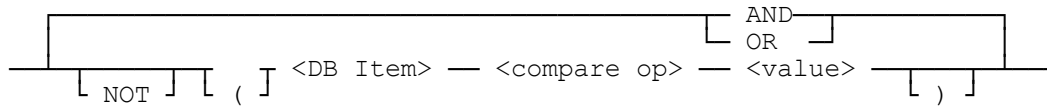
The CONFIGURE REPORT command without any options will display the current settings of the REPORT options.



## FileManager Configuration

---

<Select Spec>



<Name>

A sequence of letters, numbers, hyphens and underscores, plus wildcards including '?', '=' and '~'. A name can have additional special characters, such as a period '.', if the <Name> is enclosed in quotes.

The length of file names is controlled by the SYSOPS LONGFILENAMES system option. Refer to the MCP WFL manual for further details.

### Explanation

#### **FM CONFIG SET**

This command will list all of the existing sets that have been configured.

#### **FM CONFIG SET ALLFILES**

This command will display the set and the Set Specifications that the set name defines.

#### **FM CONFIG SET TEMPORARY -**

This command will delete the specified set from the FileManager database.

#### **FM CONFIG SET ALLFILES = FILE = ON DISK**

The command creates a new set and defines the set specifications.

#### **FM CONFIG SET STARFILES = FILE \*= ON PACK**

Configures a set called STARFILES which includes all non-usercoded files on the pack family titled PACK.

#### **FM CONFIG SET UCFILES = FILE (=) = ON PACK**

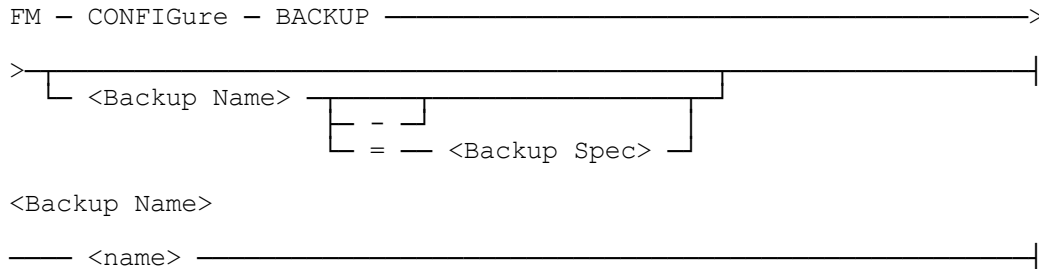
Configures a set called UCFILES which includes all usercoded files on the pack family titled PACK.

```
FM CONFIG SET ALLFILES = SET STARFILES SET UCFILES
```

## Backup Configuration

The CONFIGURE BACKUP command is used to define backup specifications for later use. A configured backup is used in a BACKUP command to initiate a backup.

### Syntax



### Explanation

#### FM CONFIGURE BACKUP

The CONFIGURE BACKUP command without any parameters will list all configured backups.

#### FM CONFIGURE BACKUP DAILYBACKUP

The CONFIGURE BACKUP with a backup name specified will list the backup specifications defined for that backup.

#### FM CONFIGURE BACKUP TEMPORARYBACKUP -

The CONFIGURE BACKUP with the - option deletes a stored backup definition from the FileManager database.

#### FM CONFIG BACKUP DAILY = INCR SET ALLFILES TO DAILYTAPE

The CONFIGURE BACKUP command defines a backup specification for later use in a BACKUP command.

*Note:* See the BACKUP command for a complete definition of the <Backup Spec>.







### **SECURE REPORT PUBLIC**

The PUBLIC option of the SECURE command removes all security restrictions from the specified command(s).

### **SECURE MODIFY SECURE**

The SECURE option of the SECURE command restricts a command from being used by any user, station, or unit unless that user, station, or unit is privileged.

### **SECURE PRIVILEGED + USER (OPERATOR2)**

The PRIVILEGED option of the SECURE command confers or removes a privileged status on a user, station, or unit. Privileged users, stations, and units are not affected by any restrictions that may have been placed on FileManager commands.

### **SECURE DEFAULT PUBLIC**

The DEFAULT option of the SECURE command with the PUBLIC attribute, removes all security restrictions from the FileManager system.

### **SECURE DEFAULT SECURE**

The DEFAULT option of the SECURE command with the SECURE option restricts all FileManager commands from being used by any user, station, or unit unless overridden by a command specific SECURE command or that user, station, or unit is privileged. This command will not be accepted unless a privileged user, station, or unit has been defined or a security definition for the SECURE command has been defined.

### **SECURE**

The SECURE command without any options displays a listing of the current security settings.

### **: FILE FILEMANAGER/SECURITY**

When security restrictions are placed on FileManager commands with the SECURE command, the security information is stored in the FileManager control file. This information is stored in such a way as to detect any attempt to change this information except through the SECURE command. Should this information become corrupt, FileManager will refuse to initialize until provided with a valid security file. The FILE option of the SECURE command provides a means of creating a backup of the FileManager security information.

The FILE option, when used with a security update creates a backup copy of all the current security information. This file should then be removed to a known, secure location. Whenever a SECURE command is issued for a system that has no current

## FileManager Security

---

security restrictions and the :FILE option is not present, FileManager will issue an RSVP to alert the user that a backup copy needs to be created. **It is strongly recommended that a backup copy of the security information be created and retained in a known, secure location.**

If for any reason it is required that a backed up version of the security restrictions be used in place of the current security values, an override option is available. Using the SECURE command with just the FILE option will replace the current security information with that in the designated security backup file. The file must be created by FileManager. Any other file will be rejected.

*Note: The SECURE command is not automatically secured. The SECURE command should be one of the first commands secured if FileManager security is to be used.*

### Caution

Use of the SECURE command implies that normal MCP file security is also used to ensure the safety of the FileManager database. The SECURE command only controls access to FileManager commands. It does not control access to the FileManager database. Access to the FileManager database should be controlled via file access security mechanisms.

### Implementation Hints

When deciding to implement security restrictions for FileManager, a review of the impact of the various FileManager commands is necessary.

#### Commands that control configuration values

The CONFIGURE and SECURE commands control the operational characteristics of the FileManager system. These commands should be restricted to those designated as system backup administrators.

#### Commands that modify the FileManager database

The ADD, DELETE, MODIFY, and REPLACE commands can modify one or more file database records. These commands should be restricted to those designated as system backup administrators.

#### Other commands

Other commands that should be considered for restriction are; TL, DATABASE, and SET. These commands could cause operational problems and inaccurate databases if misused.

# Chapter 7

## FileManager Operations

### DATABASE CLOSE Command

The DATABASE CLOSE command temporarily closes the FileManager database. DB is an abbreviation for DATABASE. This command allows the database to be closed for backup purposes without shutting the FileManager system down. While the FileManager system is active, tape activity will be queued when the database is closed. This ensures no tape activity will be missed. The user count of FileManager must be zero before the action is allowed.

#### Syntax

```
FM  _____ DATABASE _____ CLOSE _____  
   |_____ DB _____|
```

#### Explanation

##### DB CLOSE

Temporarily closes the FileManager database so that it may be backed up. While the database is closed, commands that require database access will receive an error message. The utility program will not execute.

*Note:* FileManager will attempt to automatically close and reopen its database if it detects a COPY of its database when the FM CONFIG DB setting is CLOSE. (Only if the user count is zero.)

## DATABASE OPEN Command

The DATABASE OPEN command reopens the FileManager database after it has been closed with the DATABASE CLOSE command. DB is an abbreviation for DATABASE. This command allows reopening of the database after it has been closed for backup purposes.

### Syntax

```
FM [DATABASE] OPEN _____  
   [DB] _____
```

### Explanation

#### DB OPEN

Reopens the FileManager database closed by a previous DB CLOSE command.

While the database is closed, commands that require database access will receive an error message. The utility program will not execute.

# DATABASE BACKUP Command

The DATABASE BACKUP command temporarily closes the FileManager database and copies the database to tape or disk. Once the database is copied, it is reopened. The DB BACKUP process attempts to copy all parts of the FileManager system; database, configuration, control, and object files. A BACKUP file set will then contain a complete file set if recovery is needed. DB is an abbreviation for DATABASE.

### Syntax

```
FM  DATABASE  BACKUP  <dest>
   DB
<dest>

TO  TAPE  ( <name> ( <destination attributes> ) )
    DISK  ON  <family name>
    PACK
```

### Explanation

#### DB BACKUP

#### DB BACKUP TO TAPE

Copies the FileManager database to a library maintenance tape named FILEMANAGERBACKUP. If the FileManager object files have not been renamed and reside on the same family as the database, they will be backed up also.

#### DB BACKUP TO MYFMTAPE

Copies the FileManager backup set to a tape labeled MYFMTAPE. This option allows a site to name the backup tape something other than the default of FILEMANAGERBACKUP.

#### DB BACKUP TO TAPE(SCRATCHPOOL=LIBRARY)

When the TO TAPE or TO <name> specification is used, optional destination volume attributes may be supplied. These attributes include (but are not limited to) SCRATCHPOOL, DENSITY, and SAVEFACTOR. The list of available attributes and their syntax is described in the MCP/AS Work Flow Language (WFL) Programming Reference Manual in the COPY or ADD Statement section under <destination volume attribute list>.

*Note:* Not all destination volume attributes may be allowed. The BACKUP command generates a COPY & VERIFY statement in the WFL job it

*creates. Some destination volume attributes may not be available with the VERIFY option.*

### **DB BACKUP TO DISK**

Copies the FileManager database as BACKUP/= to the pack family where the FileManager database resides. The DB RESTORE command can be used to recover a database backed using the DB BACKUP command.

### **DB BACKUP TO PACK ON BACKUPPACK**

This command copies the FileManager database as BACKUP/= to the pack family named BACKUPPACK.

*Note: While the database is closed for the backup, commands that require database access will receive an error message.*

# DATABASE REORGANIZE Command

The DATABASE REORGANIZE command temporarily closes the FileManager database and reorganizes the database files. Once the reorganization has completed the database is reopened. DB is an abbreviation for DATABASE.

### Syntax

```
FM [DATABASE] REORGanize [DB]
```

### Explanation

#### DB REORG

This command starts a reorganize process of the FileManager database. During the reorganization, the database is checked for validity, any deleted record space is returned to the system, and the index tables are rebuilt for efficiency. The pack family must have enough space to temporarily hold two copies of the database. While the database is closed for the reorganization, commands that require database access will receive an error message.

The database reorganization will create a file that contains a list of the changes and corrections made during the reorganization. The REPORT/LIST REORGANIZATION command is used to view this data. Only one reorganization results file is maintained. Executing the reorganization command will delete a previous results file.

## DATABASE RESTORE Command

The DATABASE RESTORE command restores the FileManager database from a backup copy of the database files. Once the restore has completed the database is reopened. DB is an abbreviation for DATABASE.

### Syntax

```
FM [DATABASE | DB] RESTORE [ <directory> ]
```

### Explanation

#### DB RESTORE

Restores the FileManager database from the BACKUP/= directory created with the DB BACKUP TO DISK command. The current database is overwritten and any data not in the backup database is lost.

#### DB RESTORE BACKUP/= ON BACKUPPACK

Restores the FileManager database from the BACKUP/= directory on the family BACKUPPACK.

*Note:* While the database is closed for the recovery, commands that require database access will receive an error message.

### DATABASE RECOVER Command

The DATABASE RECOVER command causes FileManager to recover records from the MCP Sumlog and apply them to the FileManager database. DB is an abbreviation for DATABASE.

#### Syntax

```
FM [DATABASE] RECOVER _____  
   [DB]
```

#### Explanation

##### DB RECOVER

FileManager calls the Sumlog and requests records from the time of the last FileManager database update. This command is a manual version of the CONFIGURE DB RECOVER FROM LOG = TRUE. Use this command if the RECOVER FROM LOG is not set and information from the system Sumlog is required after a FileManager restart. This command must be entered before any database updates have occurred to be of any use.

## DATABASE Command

The DATABASE command displays the current status of the FileManager database. The database record count and license information is also displayed. DB is an abbreviation for DATABASE.

### Syntax

```
FM [DATABASE | DB]
```

### Explanation

#### DATABASE

Displays the open or closed state of the FileManager database. The number of records in the database is displayed. FileManager license information is also displayed.

## BACKUP Command

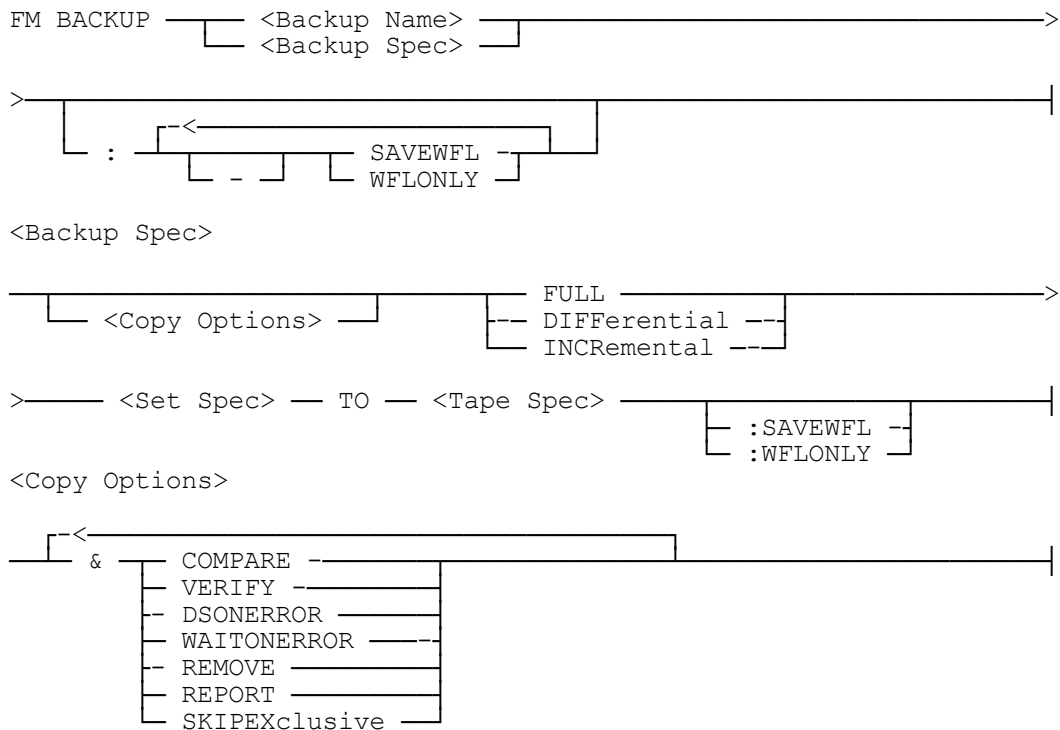
The BACKUP command copies select files from disk to tape. The files are selected based on the type of backup (Full, Differential or Incremental) and the <Set Specification>.

A full backup copies all files that match the <Set Specification>. A differential backup copies all files with a **TIMESTAMP** newer than the last time they were backed up by a full backup. An incremental backup copies all files with a **TIMESTAMP** newer than the last time they were backed up by any backup.

A full backup copies all files that match the <Set Spec>. A differential backup copies all files with a **TIMESTAMP** newer than the last time they were backed up by a full backup. An incremental backup copies all files with a **TIMESTAMP** newer than the last time they were backed up by any backup.

The FM command generates a WFL (Work Flow Language) to initiate the copy. At the completion of the operation, the files that were backed up are entered into the FileManager database.

### Syntax



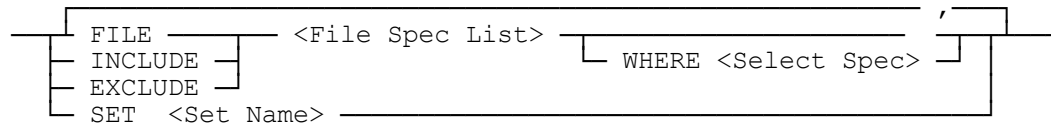
Some copy options are mutually exclusive, that is, if one is specified, the other cannot be included;

example: COMPARE cannot be combined with VERIFY.

DSONERROR cannot be combined with WAITONERROR.

Refer to the MCP/AS Work Flow Language (WFL) Programming Reference Manual for semantics of the Library/Maintenance copy options.

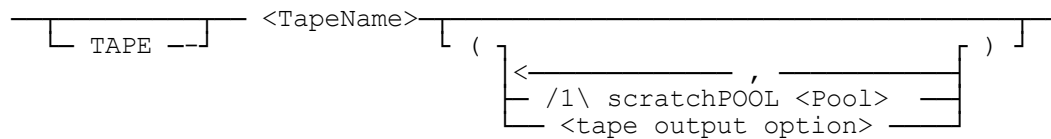
<Set Spec>



<File Spec List>



<Tape Spec>



<tape output option>

A <tape output option> can be any tape attribute allowed by WFL for an output tape. FileManager will add to this attribute as required. Refer to the MCP/AS Work Flow Language (WFL) Programming Reference Manual for <destination volume attribute list>.

### Explanation

#### **FM BACKUP & SKIPEX FULL SET UCFILES TO TAPEBACKUP (POOL=BACKUP)**

Do a full backup of all the files in the UCFILES set. Files that are opened exclusively will be skipped. The resulting tape name will be TAPEBACKUP, using media from the scratch pool defined by BACKUP.

#### **FM BACKUP & SKIPEX DIFF SET UCFILES TO TAPEBACKUP (POOL=BACKUP)**

Do a differential backup of all the files in the UCFILES set. Files that are opened exclusively will be skipped. The resulting tape name will be TAPEBACKUP, using media from the scratch pool defined by BACKUP.

### **FM BACKUP FULL FILES \*= ON PACK TO PACKBACKUP (POOL=BACKUP)**

Do a full backup of all the non-usercoded files on the pack family titled PACK. The resulting tape name will be PACKBACKUP, using media from the scratch pool defined by BACKUP.

### **FM BACKUP FULL SET UCFILES TO TAPEBACKUP:SAVEWFL**

Do a full backup of all the files in the UCFILES set. The resulting tape name will be TAPEBACKUP, using the first available scratch media (following system option 27). At completion of the backup, the generated Work Flow will be saved on disk instead of removed. Work flows are saved under the SYSTEM/FILEMANAGER/TEMP directory under the usercode and on the pack family where the software is installed.

### **FM BACKUP FULL SET UCFILES TO TAPEBACKUP:WFLONLY**

Do a full backup of all the files in the UCFILES set. The Work Flow will be generated, but NOT started. Work flows are saved under the SYSTEM/FILEMANAGER/TEMP directory under the usercode and on the pack family where the software is installed.

## RESTORE Command

The RESTORE command copies selected files from tape to disk. The files are selected based on the <Set Specification>.

### Syntax

```
FM RESTORE _____ < Set Spec > _____
      |_____ <Copy Options> _____|
_____ TO _____ <Pack Family Name> _____
                                     |_____ :SAVEWFL _____|
                                     |_____ :WFLONLY _____|
```

<Copy Options>

```
|_____ & _____|
|_____ COMPARE _____|
|_____ VERIFY _____|
|_____ DSONERROR _____|
|_____ WAITONERROR _____|
|_____ RELEASE _____|
|_____ REPORT _____|
|_____ SKIPEXclusive _____|
```

Some options are mutually exclusive, that is, if one is specified, the other cannot be included;

COMPARE cannot be combined with VERIFY.

DSONERROR cannot be combined with WAITONERROR.

Refer to the MCP/AS Work Flow Language (WFL) Programming Reference Manual for semantics of the Library/Maintenance copy options.

<Set Spec>

```
|_____ FILE _____ <File Spec List> _____ |_____ ' _____|
|_____ INCLUDE _____|
|_____ EXCLUDE _____|
|_____ SET <Set Name> _____|
|_____ WHERE <Select Spec> _____|
```

<File Spec List>

```
|_____ <file name> _____|
|_____ * = _____|
|_____ (usercode) = _____|
|_____ FROM <Pack Family Name> _____|
```

### Explanation

#### FM RESTORE & VERIFY SET UCFILES

Restore the newest version of all the files defined in the UCFILES set to the original pack family. Verify the files upon restoration.

### **FM RESTORE (UC)MYFILE ON MYPACK**

Restore the newest version of the single file (UC)MYFILE ON MYPACK.

## TL (Transfer Log) Command

The TL command changes the name of the current FileManager log file by adding the current date and time to the title and creates a new empty log file. This command can be executed whenever the site feels the log file is too big or the activity or exception reports take too long. The log is automatically transferred at the end of the month or when it has acquired 40000 records.

### Syntax

FM - TL \_\_\_\_\_|

### Explanation

#### TL

Closes the current log file and adds the current date and time to the files title. A new empty log file is then created.

### LOAD Command

The LOAD command is used to read a previous LIBMAINDIR file and re-populate the FileManager database with LIBMAINDIR information. The intent of the command is to rebuild a lost FileManager database, or to include a FileManager-generated Work Flow backup that ran while FileManager was off-line.

#### Syntax

```
FM — LOAD — LMDIR ————— <File Title> —————|
                        |
                        | — FULL —————|
                        | — INCRemental ———|
                        | — DIFFerential ———|
                        |
                        |
<File Title>
————— <Proper Library/Maintenance name of LIBMAINDIR File > —————|
```

#### Explanation

##### **LOAD LMDIR FULL (UC)LIB=FULL= ON DISK**

Find all of the LIBMAINDIR files under the UC usercode that include the string =FULL=, and load the records into the FileManager database.

##### **LOAD LMDIR INCR (UC)LIB=INCR= ON DISK**

Find all of the LIBMAINDIR files under the UC usercode that include the string =INCR=, and load the records into the FileManager database.

## QUIT Command

The QUIT command shuts down the FileManager system.

### Syntax

FM - QUIT \_\_\_\_\_|

### Explanation

#### QUIT

Shuts down the FileManager system. No new FM commands will be allowed after the QUIT has been issued.



message or the portion until the first period (.) whichever comes first. Extra spaces in the message text will be compressed unless the text is contained within quotes. The // option is available to be consistent with the Unisys email software.

### **FM EMAIL SUPPORT FILE TEST/FILE ON WORK, “File as requested”**

The FILE option of the EMAIL command causes the specified file to be placed in the email as an attachment. The email is sent to the SUPPORT account as defined by the CONFIGURE EMAIL SUPPORT command. A maximum of 10 total traces and files may be included in an email.

### **FM EMAIL SUPPORT TRACE 1204, “Trace as requested”**

The TRACE option of the EMAIL command causes the FileManager trace with the specified trace number is placed in the email as an attachment. The email is sent to the SUPPORT account as defined by the CONFIGURE EMAIL SUPPORT command. A maximum of 10 total traces and files may be included in an email.

### **FM EMAIL OPERATOR FM STATUS; END, “Check the status”**

The FM option of the EMAIL command causes the output of the specified FileManager command(s) to be placed in the email as an attachment.

### **FM EMAIL SUPPORT TRACE FM LAST; END, “LAST command has a bug”**

The TRACE option followed by the FM option is a special case of the email TRACE option. This sequence will activate the FileManager DEBUG tracing feature, execute the specified command(s), and deactivate the tracing feature. The output from the command(s) and the resulting trace files are placed in the email following the message portion of the email.

### WAIT Command

The WAIT command causes the process executing the command to pause until the wait condition has cleared. The WAIT command is designed to allow batch execution of FileManager commands to pause until asynchronous tasks have completed.

#### Syntax

```
FM - WAIT ( 

|           |
|-----------|
| <integer> |
| OK        |
| <task id> |
| DBOPEN    |
| DBCLOSE   |

 ) _____|
```

#### Explanation

##### WAIT(5)

The WAIT command with the integer parameter causes the process executing the command to pause for the integer number of seconds. If the integer is greater than 164925 then a value of 164925 will be used.

##### WAIT(OK)

The WAIT command with the OK parameter causes the process executing the command to pause until the operator allows the process to continue by entering a <mix #>OK.

##### WAIT(STACKTASK)

The WAIT command with the task id parameter causes the process executing the command to pause until the task associated with the task id has completed. If the task id has never been associated with a command, the associated command does not execute asynchronously, or the associated task has already completed then the wait command completes immediately.

##### WAIT(DBOPEN)

The WAIT command with the DBOPEN parameter causes the process executing the command to pause until the FileManager database is in an open and ready condition. If the database is already open, the wait command completes immediately.

##### WAIT(DBCLOSE)

The WAIT command with the DBCLOSE parameter causes the process executing the command to pause until the FileManager database is in a closed condition. If the database is already closed, the wait command completes immediately.

### *Special Considerations:*

- 1. The FM WAIT(<task name>) command will return an error if the task that is being waited on has been DSed. The impact is that a FM Utility batch run could abort on this WAIT. To override this functionality see the RESUME/IGNORE Dollar option card in ChapterXX.*
- 2. The WAIT(DBOPEN) command must be used if the requirement is to pause until a REORGANIZE, BACKUP, or RESTORE has completed. The WAIT(<task id>) form of the command will complete immediately if used with a DATABASE command.*
- 3. The FileManager will not allow the WAIT command to be executed on the FileManager or LOGPROCESSOR stacks as this could cause a slowdown or temporary hang of the FileManager system. In practical terms, this means that a WAIT command entered at the operators console (ODT/SPO) will cause a warning and the WAIT will complete immediately.*

See Chapter XX, Batch Mode Execution later in this manual for additional information on assigning a task id to a command.



# Chapter 8

## FileManager Reports

### OLDEST Command

The OLDEST command displays the tape with the oldest creation date that matches the label selection string.

#### Syntax

```
FM — OLDEST — <Set Selection> _____|
```

#### Explanation

##### **OLD (UC)LIST ON MYPACK**

Searches for all files matching the file title (UC)LIST ON MYPACK and displays the version with the oldest timestamp. This command is useful for quickly finding the oldest file of multiple versions of a file in the backup database.

##### **OLD (UC)LIST/= ON MYPACK**

Searches for all files matching the file title (UC)LIST/= ON MYPACK and displays the version with the oldest timestamp.

### NEWEST Command

The NEWEST command displays the tape with the latest creation date that matches the label selection string.

#### Syntax

```
FM — NEWest — <Set Selection> _____|
```

#### Explanation

##### NEW (UC)LIST ON MYPACK

Searches for all files matching the file title (UC)LIST ON MYPACK and displays the version with the most recent timestamp. This command is useful for quickly finding the newest file of multiple versions of a file in the backup database.

##### NEW (UC)LIST/= ON MYPACK

Searches for all files matching the file title (UC)LIST/= ON MYPACK and displays the version with the newest timestamp.

## ALL Command

The ALL command displays all files matching the selection string.

### Syntax

```
FM - ALL — <Set Selection> —————|
                                     |
                                     | BY ————|
                                     |         |
                                     |         | TITLE ————|
                                     |         |         |
                                     |         |         | TIME ————|
```

### Explanation

#### ALL (UC)LIST ON MYPACK

Searches for all files matching the file title (UC)LIST ON MYPACK and displays all versions of that file.

#### ALL (UC)LIST= ON MYPACK BY TITLE

Searches for all files matching the file title (UC)LIST/= ON MYPACK and displays all versions of the matching files, sorted by file title.

The default sort is BY TIME.

*Note:* The capabilities of the ALL command have been merged with the LIST/REPORT ALL command. Use the REPORT ALL command if a hard copy of this output is required.

### LAST Command

The LAST command displays the last 20 files created in the backup database.

#### Syntax

```
FM — LAST _____|
```

#### Explanation

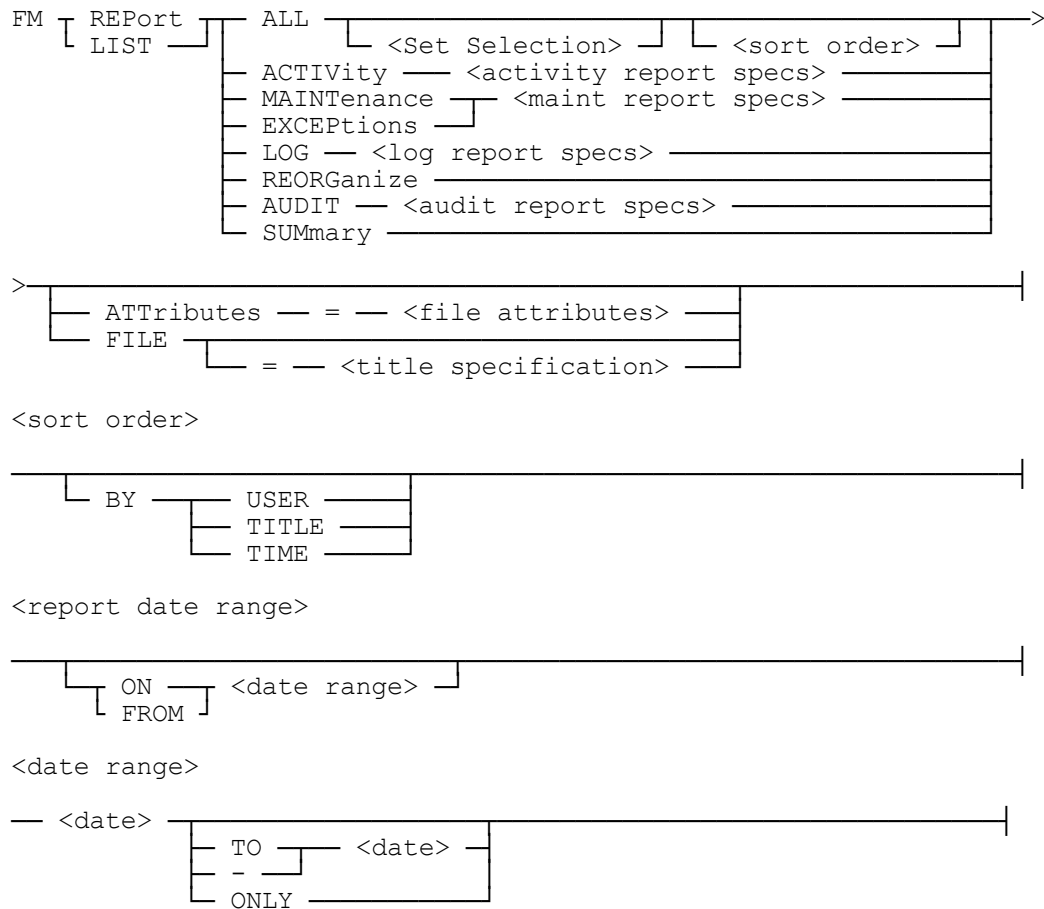
##### LAST

Displays the last 20 files created in the backup database. They are displayed in order of newest to oldest.

## REPORT and LIST Commands

The REPORT command prepares one of the FileManager systems printed reports. The LIST command is the same as the REPORT command except that the report is returned to the requesting ODT or station. The format of LIST reports is slightly different due to the smaller width of a terminal compared to a printer.

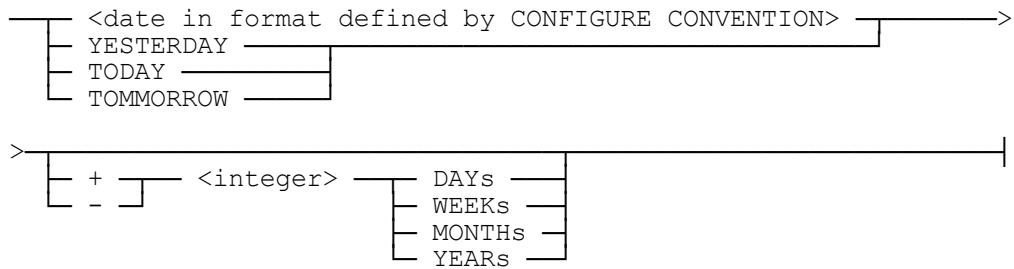
### Syntax



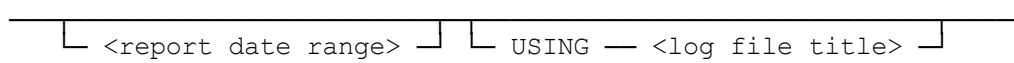
## FileManager Reports

---

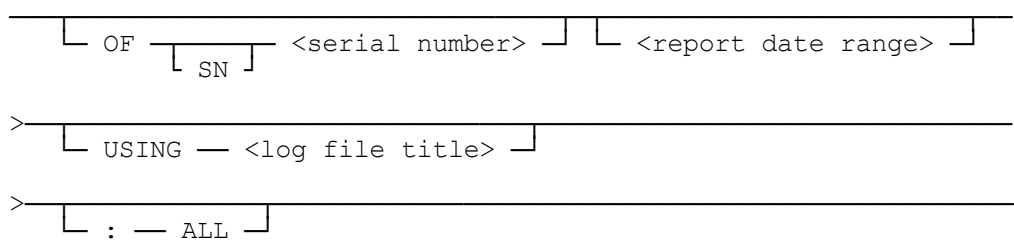
<date>



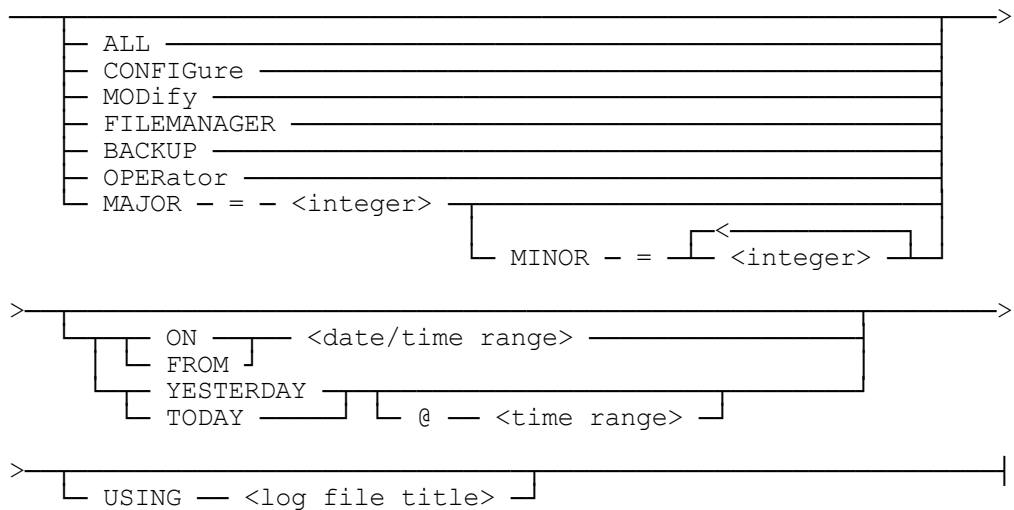
<maint report specs>



<activity report specs>



<log report specs>



```
<audit report specs>
```

```

┌──────────────────────────────────────────────────────────────────────────────────┐
│ OF ┌── SN ── <serial number> ──┐ ┌── <report date range> ──┐                    │
│   └── MIX ── <integer> ──┐    └──────────────────────────┘                    │
│   └── LABel ── <label ID> ──┘    ┌──────────────────────────────────────────┐    │
│   └──────────────────────────┘    └──────────────────────────────────────────┘    │
└──────────────────────────────────────────────────────────────────────────────────┘
```

```
> ┌── USING ── <audit file title> ──┘ ┌──────────────────────────────────────────┘
```

### Explanation

#### REPORT ALL

Creates a printed report of all the tapes in the FileManager database.

#### LIST ALL

Displays a report of all the tapes in the FileManager database at the originating station or ODT.

#### ATTRIBUTES = FORMID = "FILEMANAGER", PAGESIZE = 80

The ATTRIBUTES option may only be used with the REPORT command. This option provides file attributes to be applied to the printer file of the report. These attributes are applied after the attributes are defined with the CONFIGURE REPORT command and so may override those default attributes.

#### FILE = "MYREPORT.TXT" ON MYPACK

The FILE option may only be used with the REPORT command. This option causes all report output to be placed in a disk file. The specified file must not already exist. If a file title is not specified, the file will be created with a title of the form SYSTEM/FILEMANAGER/REPORT/<date>/<time> with the user code and family under which FileManager is running. Open system file titles are supported as defined by the LTITLE file attribute if allowed by the LONGFILENAMES system option.

### ALL Files Report

The ALL report lists all files in the backup database. Files that have been added with the ADD command but not yet seen are not listed. The report lists a file's serial number, label and file ID, creation date and time, reel number, cycle and version, and density. The printed report will also list the name of the program that created the tape.

#### Syntax

```
FM | REPort | ALL |-----| . .
  | LIST |      | <Set Selection> | <sort order> |
<sort order>
  | BY | TITLE |
  |   | TIME  |
```

#### Explanation

##### ALL

Creates a report of all the files in the FileManager database sorted by TITLE.

##### ALL BY TITLE

Creates a report of all the files in the FileManager database in serial number order.

##### ALL BY TIME

Creates a report of all the tapes in the FileManager database in alphabetical order by label name. Scratch tapes will be listed by their previous label name.

##### ALL ABCBACKUP=

Creates a report of all the files in the FileManager database that have a filename starting with the characters ABCBACKUP.

## File Activity Report

The ACTIVITY report lists all tapes that have been used on the current or specified date range. All tapes are listed, including those not tracked by the FileManager system. If a tape was used more than once, only its last activity is listed. This report is useful for verifying that tapes used during the day are replaced in the library. The activity report may also be used to report all activity for a particular serial number.

### Syntax

```
FM [ REPort ] [ LIST ] ACTIVITY [ _____ ] . . .
   [ BACKUP ]
```

### Explanation

#### **ACTIVITY**

Creates a report of all activity happening today.

#### **ACTIVITY BACKUP**

Creates a report of all backup job activity happening today.

#### **ACTIVITY ON 10/2/93**

Creates a report of all the files used on 10/2/93.

#### **ACTIVITY FROM 10/2/93 TO 10/4/93**

Creates a report of all the files used on the days 10/2/93 through 10/4/94.

#### **ACTIVITY FROM YESTERDAY TO TODAY: ALL**

The default Activity report only shows the last action that was done to that file. Use the ALL option to show all activity for all files in the specified date range.

#### **ACTIVITY BACKUP USING FILEMANAGER/LOG/OTHER/HOST**

The USING feature of the Activity report allows the generation of an Activity report using a FileManager log file other than those in the current log chain. This feature can be useful for generating reports from old archived logs or logs from a different FileManager host.

# File Exceptions Report

The EXCEPTIONS report is similar to the activity report but list only tapes with exception conditions. Exception conditions include tapes that had media errors, tapes used but not tracked, and tapes whose labels do not match those in the database. Review this report daily to see if any operations actions are required.

## Syntax

```
FM [ REPort ] EXCEPTIONS _____ . . .  
   [ LIST ]
```

## Explanation

### **EXCEPTIONS**

Creates a report of all the file exceptions that were identified today.

### **EXCEPTIONS ON 5/7/93**

Creates a report of all the file exceptions that were identified on 5/7/93.

### **EXCEPTIONS ON 5/7/93 - 5/10/93**

Creates a report of all the file exceptions identified on the days 5/7/93 through 5/10/93.

### **EXCEPTIONS ON 5/7/93 USING FILEMANAGER/LOG/OTHER/HOST**

The USING feature of the Exception report allows the generation of an Exception report using a FileManager log file other than those in the current log chain. This feature can be useful for generating reports from old archived logs or logs from a different FileManager host.

## Maintenance Report

The MAINTENANCE report is a report of exceptions and error conditions seen by the FileManager software. The report lists log entries that are placed in the FileManager log whenever a program error or hardware exception condition is observed. The operator may also create a log entry through the use of the LOG command. The report does not include the same entries as described in the EXCEPTION report above. This report is an additional diagnostic tool to help resolve hardware or software problems.

### Syntax

```
FM [ REPORT ] MAINTENANCE _____ . . .  
   [ LIST ]
```

### Explanation

#### MAINTENANCE

Creates a report of all the FileManager log entries that were created today.

#### MAINTENANCE ON 5/7/93

Creates a report of all the FileManager log entries that were created on 5/7/93.

#### MAINTENANCE ON 5/7/93 - 5/10/93

Creates a report of all the FileManager log entries for the days 5/7/93 through 5/10/93.

#### MAINTENANCE ON 5/7/93 USING FILEMANAGER/LOG/OTHER/HOST

The USING feature of the Maintenance report allows the generation of a Maintenance report using a FileManager log file other than those in the current log chain. This feature can be useful for generating reports from old archived logs or logs from a different FileManager host.

*Note:* The MAINTENANCE report is being replaced by the LOG report which has additional features and capabilities.

### Log Report

The LOG report is a report of exceptions and error conditions seen by the FileManager software. The report lists log entries that are placed in the FileManager log whenever a program error or hardware exception condition is observed. The operator may also create a log entry through the use of the LOG command. The report does not include the same entries as described in the EXCEPTION report above. This report is an additional diagnostic tool to help resolve hardware or software problems.

#### Syntax

```
FM [ REPort ] LOG _____ . . .  
   [ LIST ]
```

#### Explanation

##### **REPORT LOG ALL**

Creates a report of all the FileManager maintenance log entries that were created today. ALL is the default if no selection condition is specified. This report is essentially the same as that produced by the REPORT MAINT command.

##### **REPORT LOG CONFIGURE ON 2/1/2000**

Creates a report of all the FileManager configuration changes that were made on 2/1/2000.

##### **REPORT LOG MODIFY TODAY @ 10:00 – 13:00**

Creates a report of all the FileManager database modification commands (ADD, DELETE, MODIFY) that were processed between 10:00 AM and 1:00 PM on the current day.

##### **REPORT LOG FILEMANAGER FROM 1/1/2000 @ 10:00 TO 2/2/2000 @ 15:00**

Creates a report of all the log entries made by the FileManager software processes between January 1, 2000 at 10:00 AM and February 2, 2000 at 3:00 PM. If the time portion of the start date is not specified, midnight of the start date is assumed. If the time portion of the ending date is not specified, midnight of the ending date is assumed.

##### **LIST LOG MAJOR = 6**

Creates a report of all the FileManager configuration changes that were made today. When a Minor type or types are not specified, all Minor types are reported. See the table below for a list of the major and minor log types. The LIST command causes the output to go to the terminal.

**REPORT LOG MODIFY ON 5/7/93  
USING FILEMANAGER/LOG/OTHER/HOST**

The USING feature of the Log report allows the generation of a Log report using a FileManager log file other than those in the current log chain. This feature can be useful for generating reports from old archived logs or logs from a different FileManager host.

**FileManager Log Major and Minor Types**

| Major Type | Minor Type | Description   |
|------------|------------|---|
| 1          |            | FileManager messages                                |
|            | 0          | General FileManager entries                         |
|            | 1          | FileManager fault entries                           |
|            | 2          | Purge allowed entries                               |
|            | 3          | Purge denied entries                                |
| 2          |            | LibraryManager messages                             |
|            | 1          | LibraryManager events and errors                    |
|            | 2          | Import or Export process start or stop              |
| 3          |            | Operator commands                                   |
| 4          |            | DSICONTROLLER Support messages (currently not used) |
| 5          |            | Library Controller messages                         |
| 6          |            | FileManager configuration changes                   |
|            | 1          | Configure OPERATIONS command                        |
|            | 2          | Configure DATABASE command                          |
|            | 3          | Configure REPORT command                            |
|            | 4          | Configure TRACKING command                          |
|            | 5          | Configure HOSTS command                             |
|            | 6          | Configure RETENTION command                         |
|            | 7          | Configure RETIRING command                          |
|            | 8          | Configure CLEANING command                          |
|            | 9          | Configure CONVENTION command                        |
|            | 10         | Configure LIBRARY command                           |
|            | 11         | Configure LOCATION command                          |
|            | 14         | Configure EMAIL command                             |
|            | 15         | Configure MT command                                |
|            | 16         | Configure STACKING command                          |
| 7          |            | FileManager database changes                        |
|            | 1          | FileManager database ADD command                    |
|            | 2          | FileManager database DELETE command                 |
|            | 3          | FileManager database MODIFY command                 |
| 8          |            | CSCLIB Inventory Notifications                      |
|            | 1          | CSCLIB Eject Notification                           |
|            | 2          | CSCLIB Enter Notification                           |
|            | 3          | CSCLIB Error/Exception Notification                 |
|            | 4          | CSCLIB Mount request                                |

## FileManager Reports

---

|    |   |   |
|----|---|---|
|    | 5 | CSCLIB Eject request                        |
| 9  |   | CSC-A interface entries                     |
|    | 1 | CSC-A eject notification                    |
|    | 2 | CSC-A import notification                   |
|    | 3 | CSC-A error/exception notification          |
|    | 4 | CSC-A inventory build complete notification |
|    | 5 | CSC-A mount statistics report               |
| 10 |   | MT DriveManager entries                     |
|    | 1 | MT Drive Added                              |
|    | 2 | MT Drive Deleted                            |
|    | 3 | MT Drive Replaced                           |
| 11 |   | Remote logging entries                      |
|    | 1 | Remote identity added                       |
|    | 2 | Remote identity deleted                     |
| 12 |   | TapeStack entries                           |
|    | 0 | Messages about TapeStack processes          |
|    | 1 | VTV stack messages                          |
|    | 2 | VTV unstack messages                        |
|    | 3 | VTV append messages                         |
|    | 4 | Stacked tape merge/duplicate messages       |
|    | 5 | VTV invalidate messages                     |
|    | 6 | Tape size messages                          |

## Reorganize Report

The REORGANIZE report is a listing of all the changes and corrections made when the FileManager database reorganization was last run. A file is created whenever the reorganization process is executed. This report lists the entries in that file.

### Syntax

```
FM [ REPort ] REORGanize _____ . . .  
   [ LIST ]
```

### Explanation

#### **LIST REORGANIZE**

Creates a report of all the changes and corrections made by the last run of the FileManager database reorganization.

# Summary Report

The SUMMARY report produces a summary of the information contained in the FileManager database.

### Syntax

```
FM [ REPort ] SUMmary _____ . . .  
   [ LIST ]
```

### Explanation

#### **REPORT SUMMARY**

Creates a report that summarizes the information contained in the FileManager database.

## Audit Report

The AUDIT report is a report of accesses made to the FileManager database. Only the main file data structure is audited. The control, log, and other files are not audited. The audit file has before and after images of records that have been updated. The audit file also contains information on database opens, closes, backups and reorganizations.

### Syntax

```
FM [ REPort ] AUDIT _____ . . .  
   [ LIST ]
```

### Explanation

#### **REPORT AUDIT**

Creates a report of all the FileManager database audit entries that were created today.

#### **REPORT AUDIT OF SN 123456**

Creates a report of all the FileManager database audit entries that were created today, plus accessed records for the specified serial number. If audit report specifies a specific serial number, mix number, or label name, the report filters out all records that do not match the requested item.

#### **REPORT AUDIT ON 5/7/93**

Creates a report of all the FileManager database audit entries that were created on 5/7/93.

#### **REPORT AUDIT ON 5/7/93 - 5/10/93**

Creates a report of all the FileManager database audit entries for the days 5/7/93 through 5/10/93.

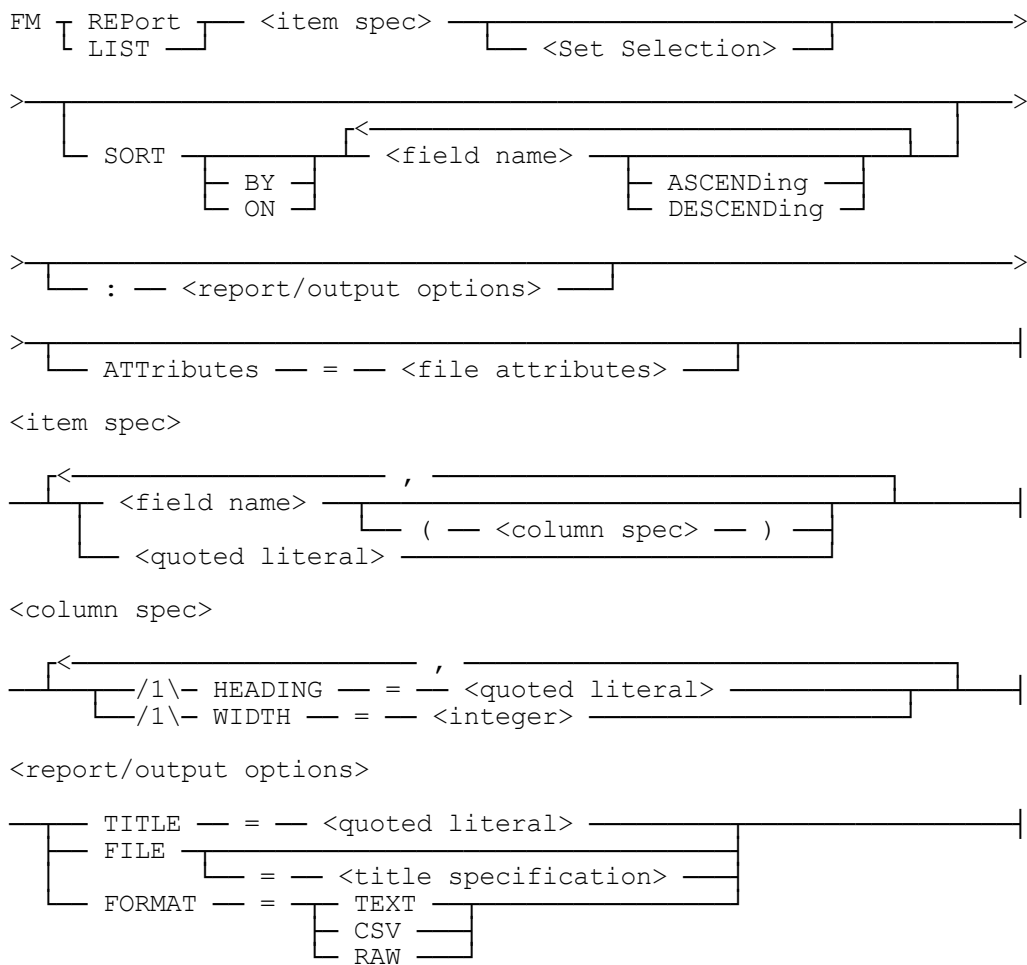
#### **REPORT AUDIT ON 5/7/93 USING SYSTEM/FILEMANAGER/DB/DBAUDIT12**

The USING feature of the Audit report allows the generation of an Audit report using a FileManager audit file other than those in the current audit chain. This feature can be useful for generating reports from old archived audits or audits from a different FileManager host.

## Ad Hoc Reports

The ad hoc reporting capability allows a custom report to be created when one of the standard reports does not meet a sites requirements. Ad hoc reports allow for the selection of the reported database fields, the specifying of database record selection criteria, and the specifying of the sort order of the output. Ad hoc report output can also be stored on disk and formatted for different uses.

### Syntax



### Explanation

**FM LIST SERIALNO," , LABEL," , TITLE**  
**FM REPORT SERIALNO," ,LABEL," , TITLE**

Creates a report that shows files, listing backup tape serial number, label and file title. Since a Set Selection was not specified, all files are reported. The use of LIST or

REPORT determines the width of the report and its output destination. Using LIST specifies that the report will be 80 characters in width and the output is to be returned to the originating station. (Note: if the LIST command is received from a source to which the output can not be sent, such as an AX, the report will be shown via the MCP display (MSG) mechanism.) Using the REPORT option specifies a report that is 132 characters wide and the output is directed to the system printer.

### **FM REPORT SERIALNO (HEADING="SERIAL #"), " ", LABEL(WIDTH=40)**

The <column spec> options are used to control attributes of the report columns. The HEADING option sets the title at the top of the report column. The WIDTH option is used to control how many characters are used to display the data of the field. If WIDTH is not specified, then the column width will be the greater of the default column width for the data item and the column heading string. If the specified width is less than the column heading and/or the data to be show, the heading and/or data will be truncated.

### **FM REPORT TIMESTAMP, " ", TITLE WHERE TAPECREATED GE YESTERDAY**

The WHERE option allows for selected records to be reported. In this example, only those tapes that are on tapes that were created yesterday or later will be listed in the report.

### **FM REPORT SERIALNO," ", LABEL SORT BY SERIALNO**

The SORT option allows the output listing to be organized in a user specified order. If the SORT option is not specified the output will be in serial number order whether or not the SERIALNO field has been selected for reporting. The ASCENDING and DESCENDING sort modifiers determine if the sort item is to be ordered lower to higher or higher to lower. The default is ASCENDING.

### **FM REPORT SERIALNO, " ", TITLE :TITLE="MY CUSTOM REPORT"**

The TITLE report option allows the ad hoc report to be given a custom title. If the TITLE report option is not specified, the default is TAPEMANAGER ADHOC REPORT.

### **FM REPORT SERIALNO, " ", TITLE :FORMAT = CSV**

The FORMAT report option specifies what kind of output will be generated by the report. The FORMAT option is only available with the REPORT command. The TEXT format specifies that the output is to be in normal textual strings. If FORMAT is not specified, TEXT is the default.

The CSV format specifies that the output is to be in Comma Separated Values (CSV). CSV is a file format understood by many PC based spreadsheet and word processor programs. If the CSV format is specified, the FILE option is assumed. If the FILE

## FileManager Reports

---

option does not specify a file name then ends in a quote (i.e. "XFERFILE.CSV") then the last node of the file name will have .CSV appended to it and quoted.

The RAW format specified that the output is to be in raw database records. These records are a copy of the FileManager database record. The FIELD\_VALUE\_FM procedure (see Appendix D) should be used to interpret the information in these records. If the RAW format is specified, the FILE option is assumed.

### **FM REPORT SERIALNO, " ", TITLE : FILE = MY/REPORT**

The FILE option may only be used with the REPORT command. This option causes all report output to be placed in a disk file. The specified file must not already exist. If a file title is not specified, the file will be created with a default title of the form SYSTEM/TAPEMANAGER/REPORT/<date>/<time> with the user code and family that FileManager is running with. Open system file titles are supported as defined by the LTITLE file attribute if the LONGFILENAMES system option is set.

## STATUS Command

The STATUS command displays information about the current state of various components of the FileManager system.

### Syntax

```
FM — STATUS — [ OF ] — [ FILEMANAGER — ]
                  [ DATABASE — ]
                  [ MACROs — ]
```

### Explanation

#### STATUS or STATUS FILEMANAGER

Displays the current status of the FileManager system. The display shows if the FileManager system is operational (or not) and what mode FileManager is running in (active or passive). Also displayed is the open/closed state of the FileManager database. The FileManager version is also displayed. FileManager will then display a list of the tasks associated with FileManager so they can be easily found in a heavy mix environment.

#### STATUS DB

Displays the current status of the FileManager database. The open or closed state of the database is displayed. The number of records in the database is also displayed. FileManager license information is also displayed.

#### STATUS MACROS

This status option displays the names of the currently defined macros.



file. If the selection specification results in more than one tape being reported, the single line per tape format is used and the ALL option is ignored.

### **INQUIRE WHERE LABEL = =Z=**

Selects any files that have been copied to a tape with a Z in the label. If only one file is selected, a full page on that file is displayed. If more than one file is selected, a single line per file is displayed.

### **FIND WHERE TAPECREATED = 1/19/2000 AND LABEL = FMFULL=**

Selects any files that were copied to tapes on January 19, 200, with a tape label that starts with FMULL. The format of the <date> and <time> values must conform with the Convention Configuration except a 4-digit year may be specified even though only a 2-digit year was defined in the Convention Configuration.

### **FIND WHERE LABEL =DBBACKUP= SORT ON TAPECREATED DESCENDING**

The SORT option of the find command allows the output of the FIND to be ordered based on one or fields within the file record. The fields selected for ordering do not have to be items that are displayed. Each field selected for sorting may specify an ASCENDING or DESCENDING order. Ascending sort order is assumed if not specified. A FIND command with the SORT option will always use the single line per tape output even if only 1 tape is selected.

### **FIND WHERE TAPECREATED = 6/30/09 :FILE EXPORT/063097**

Selects all the tapes created with the specified processing date and places those records in a disk file. If a file name is not specified, the name of the file will depend on the FORMAT of the output file. For files with FORMAT = RAW (default), the default file title is SYSTEM/TAPEMANAGER/EXTRACT in the same user code and family as the FileManager files. The file must not already exist. For disk files with a FORMAT of TEXT or CSV, the file will be created with a default title of the form SYSTEM/TAPEMANAGER/REPORT/<date>/<time> with the user code and family that FileManager is running with. Open system file titles are supported as defined by the LTITLE file attribute if the LONGFILENAMES system option is set.

### **FIND WHERE CREATIONDATE = YESTERDAY :FILE FORMAT = TEXT**

Selects all the files that were created on the specified creation date and places those records in a disk file. The format of the disk file depends on the FORMAT option. The RAW format specifies that the output is to be in raw database records. These records are a copy of the FileManager database record. The FIELD\_VALUE\_FM procedure (see Appendix D) should be used to interpret the information in these records. If FORMAT is not specified RAW is the default.

## FileManager Reports

---

The TEXT format specifies that the output is to be placed in a disk file in a format similar to that if the output had been printed.

The CSV format specifies that the output is to be in Comma Separated Values (CSV). CSV is a file format understood by many PC based spreadsheet and word processor programs. The output to the CSV file is fixed and contains the fields that are seen in the printed report available from the FIND command.

### **FIND WHERE LABEL = BACKUP= :PRINTER**

Selects and prints the current information for any tapes whose label name starts with BACKUP. The printer options defined with the FM CONFIG REPORT command are applied to the output.

### **FIND WHERE LABEL = #LABELID SORT BY TITLE DESCENDING TAPECREATED DESCENDING: QUERY**

The QUERY option causes the records selected by the FIND command to be retained in an internal queue rather than be displayed, printed, or sent to a file. The purpose of the QUERY option is to allow the results of a FIND to be used within a macro definition. See the QUERY description in the FileManager Macros section.

## VERSION Command

The VERSION command displays the version information of the FileManager software. If the LibraryManager system is configured with FileManager additional information is displayed.

### Syntax

```
FM — VERsion _____|
```

### Explanation

#### VERSION

Displays the version of the FileManager software that is running. If LibraryManager is also running then the version information for Tapelibrary Support and DSIcontroller Support are also displayed. For each library attached, the microcode version of the DSIcontroller and the microcode version of the tape library is displayed.



# Chapter 5

## File Database Maintenance

### ADD Command

The ADD command is used to add file records to the database so that those files will be tracked. Its primary purpose is to initially add a range of SNs to the database so that the FM CONFIG OPERATIONS UNKNOWN option can be set to IGNORE. This command can also add tapes to the database that do not match the current tracking specification so that they will be tracked.

#### Syntax

```
FM — ADD — SN — <SN list> —————>
> —————|
  | WITH — <update spec> |
  | <update spec>         |
  | <field name> = ' <value> |
```

#### Explanation

##### FM ADD SN 200000-201999

Creates records in the tape database for tapes with serial numbers from 200000 through 201999.

##### WITH REELLENGTH = 2400

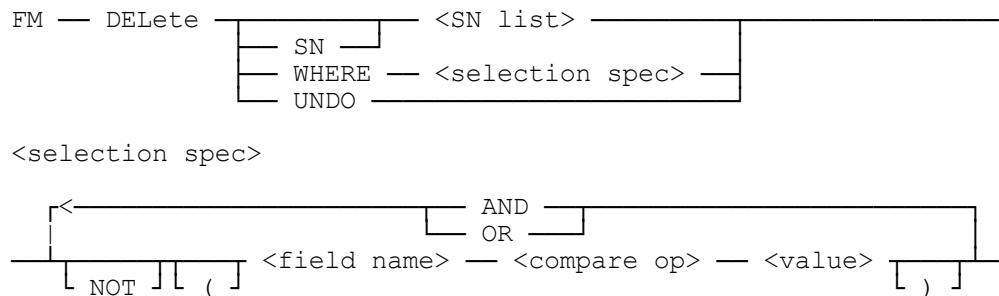
The WITH part is used to specify initial tape field values for the added tapes such as the REELLENGTH.

*Note:* If many tapes are being added (i.e., FM ADD SN 100000-199999) it may take some time before a response is returned. Tapes added with the ADD command do not show up in any reports or displays until they have been used on the system.

# DELETE Command

The DELETE command is used to delete records from the database. It is used to remove files from the system that should no longer be tracked or reported. Deleted records are not actually removed from the database but rather marked as deleted.

### Syntax



### Explanation

#### FM DELETE MYSN

Removes the tape record from the database for the tape with the serial number of MYSN.

*Note:* If the serial number is within the range specified by the FM CONFIG TRACKING command and the system is in AUTO mode, the tape will be placed back in the system the next time it is seen.

#### FM DELETE WHERE DATENEW < TODAY – 5 YEARS

Removes all tape records that have a new date greater than 5 years old. The WHERE clause can be used with any database field (see Appendix B).

#### FM UNDO

The UNDO option of the DELETE command removes the effects of the last DELETE command (i.e. un-deletes). The UNDO option can be used multiple times to undo successive DELETE commands in the reverse order in which they were done. UNDO will not undelete any records that may have been added back in that were part of the delete process. The UNDO will not be able to find any records to un-delete after a DATABASE REORGANIZE has been done. A database reorganize physically removes the deleted records from the database.

*Note:* A DATABASE REORGANIZE must be done to recover the database space used by deleted records. This is particularly important if the database record count is near the limit set by the license key. Use the STATUS DB command to display license and record count information.



## MODIFY Command

The MODIFY command is used to change fields in tape records in the database. Tapes may be selected by serial number, by label, or by the values in the records fields. Tapes that are selected may have their fields changed to new values.

### Syntax

```

FM — MODIfy ———— <SN list> —————>
      |—— SN ——|
      |—— LABEL — <label ID> ————|
      |—— WHERE — <selection spec> —|
>— WITH — <update spec> —————|
<selection spec>
  <————— AND —————>
  <————— OR —————>
  [ NOT ] [ ( ] <field name> — <compare op> — <value> [ ) ]
<update spec>
  <————— ' —————>
  <field name> — = — <value>—————|
  
```

### Explanation

#### FM MODIFY SN 100101 WITH PERMANENT = TRUE

Selects the tape with serial number 100101 and changes the PERMANENT field to TRUE so that the tape cannot be purged.

#### FM MODIFY LABEL MYDB= WITH RETAINGENS = 4

Selects all tapes whose labels start with MYDB and changes the RETAINGENS field to 4 so that the tape cannot be purged unless there are four newer generations.

#### FM MODIFY SN 1234 WITH RETENTION = 0

Selects the tape with serial number 1234 and clears the retention values so that the tape can be purged.

#### FM MODIFY WHERE REELLENGTH = 0 WITH REELLENGTH = 2400

Selects all tapes whose REELLENGTH value is 0 and changes the REELLENGTH value to 2400.

*Note:* Some fields in the record, such as SERIALNO, are read only and may not be modified.

***Note:** For multi-host FileManager environments, the MODIFY command may not be able to update all records that match the <selection spec>. In a multi-host environment, a tape record is “owned” by one host at a time. Only the “owning” host can modify the tape record. Therefore on multi-host systems the MODIFY command may need to be executed on each host in the multi-host environment in order to ensure that all records are updated.*

## **Backup Database Maintenance**

---

# Chapter 6

## FileManager Macros

### Overview of FileManager Macros

The FileManager macro facility provides a means for FileManager commands and other strings to be processed based on a defined name. Invoking a macro name is the same as entering the command text manually.

FileManager macros may be stored as text in an ordinary text file on MCP host machines. FileManager macros are manually loaded from this text file using the LOAD MACROS command or the execution of a DO command referencing a file containing macro DEFINE commands. When macro definitions are loaded via the FM LOAD MACROS command, all the currently defined macros are deleted first. All the currently defined macros may be deleted with the FM UNLOAD MACROS command. The set of loaded macros can be modified on-line.

A FileManager macro can override the definition of a standard FileManager command. For instance, the macro with the name "PURGE", could replace the standard FileManager PURGE command. Other keywords (e.g. "SLOT") are always recognized in context. (See OVERRIDE below)

The definition of a macro includes principally rules for expanding into text, which represents valid FileManager command input. The definition may also include rules for invoking the macro automatically based on the text of a program's system display or RSVP message, or a FileManager event message.

The set of loaded macro definitions is recognized throughout the FileManager software installation, so utility sessions, ODT input, and TMREMOTESPO input all are treated the same.

### Defining a Macro


Macro definition uses the same syntax whether done on-line or in a loaded text file. The definition begins with the keyword DEFINE and ends with the keyword END. A name must be placed after the END, matching the name of the macro being defined. Semicolons in a text file separate successive macro definitions.


Macro definitions are not stored over FileManager sessions. It is recommended that a site's regularly used macro definitions be stored in a STARTUP file so that the macro definitions are loaded each time FileManager is initiated. (see the DO command)

## FileManager Macros

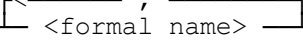
---

### Syntax

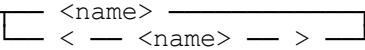
FM — DEFINE — <name> —  —>

>  = — <body> —————|

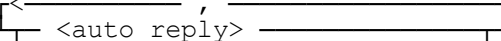
<formal parameters>

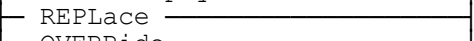
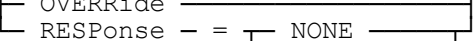
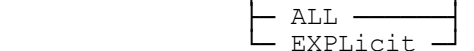
—  —————|

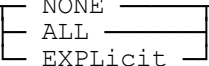
<formal name>

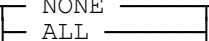
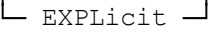

—  —————|

<options>

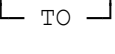
— : —  —————|




-  —————|
-  —————|
-  —————|

RESPonse — = —  —————|


-  —————|
-  —————|
-  —————|

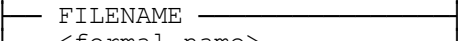
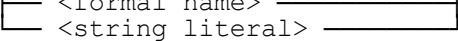

<auto reply>

REPLY —  —————|

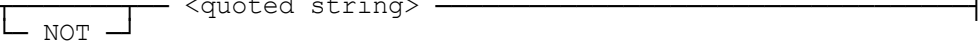
-  —————|
-  —————|
-  —————|


<message template>

—  —————|

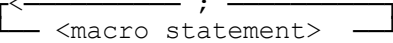
-  —————|
-  —————|
-  —————|

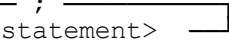
<string literal>

—  —————|

— NOT —  —————|

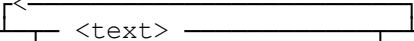
<body>

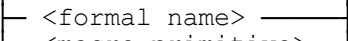
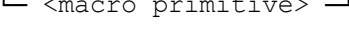
BEGIN —  —————|

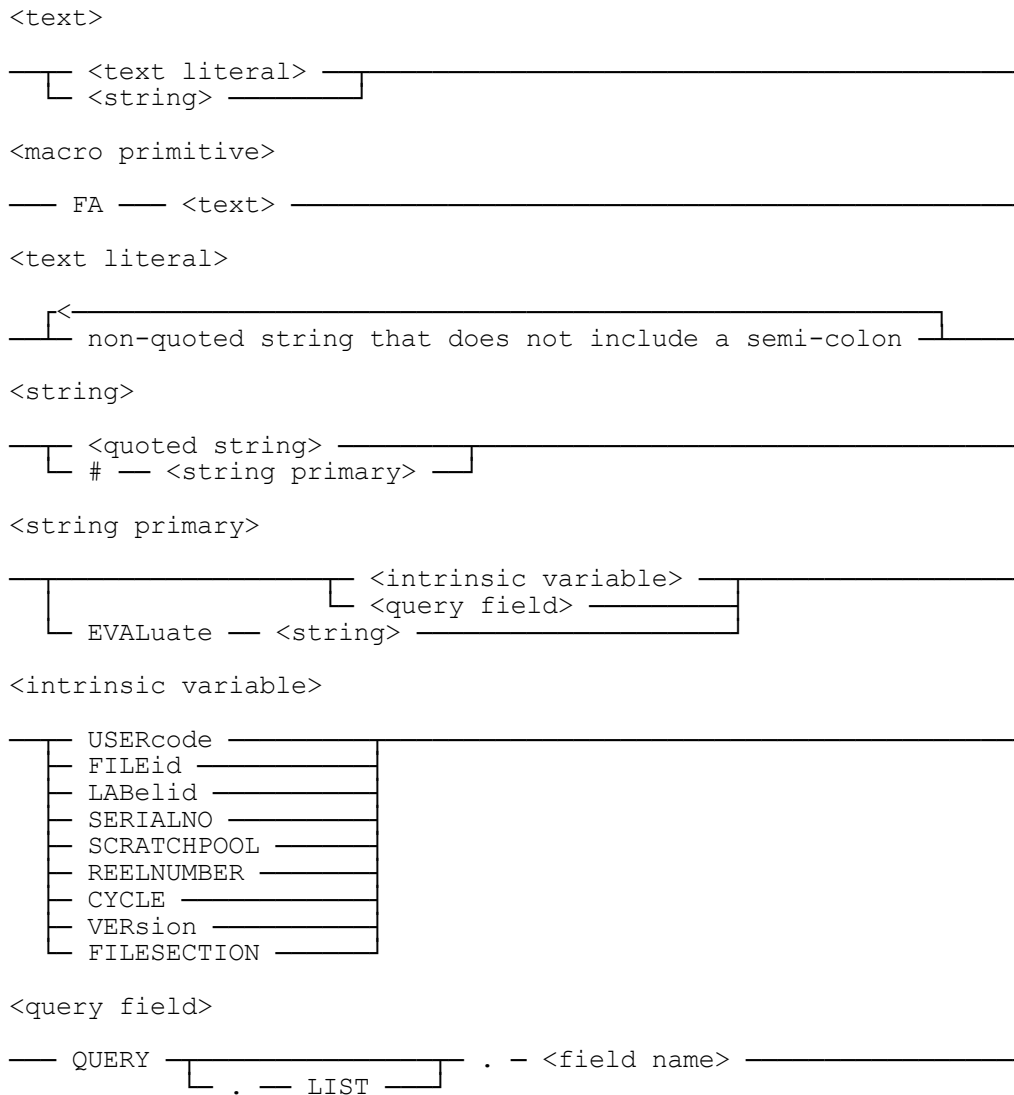
—  —————|

END — <name> —————|

<macro statement>

—  —————|

-  —————|
-  —————|



### Macro Parameters

Macro parameters allow variable information to be passed to the macro deinfo. Parameters may also be used in auto macros (described below) to acquire the parameter information from the message that triggers the auto macro. When a macro is invoked, the supplied parameters are inserted in the body of the macro at the designated places as if the text had been originally written in the macro definition. All defined parameters must be supplied when the macro is invoked. Any extra parameters supplied when the macro is invoked are ignored.

### Macro Options

Macro options give the macro author control over how macro functions are used and how they operate.

## FileManager Macros

---

The **REPLY** option defines a macro that is executed automatically (auto macro). This allows FileManager to invoke a macro in response to a particular display or RSVP from a program. A program may any program producing display or RSVP text that fits the <message template>.

A message template is merely an abstract description of an acceptable message. It contains strings that are required to appear or portions that may contain anything as indicated by the **SKIP** keyword. The template can also specify that certain strings not appear. Only if the program's text fits the template will the macro be invoked. Except as part of a string literal, spaces are ignored in matching a string to a template. A <string literal> that appears in a message template may contain wildcard characters so the template can match more than one specific message. If the **FILENAME** keyword appears following a string literal or <formal name>, then the matching string must conform to the system definition for a valid display form file name. If the **SKIP** keyword appears following a string literal, then anything that follows the given string literal will match.

The **REPLACE** option provides for replacing a current macro definition with a new macro definition. The **REPLACE** option ensures that the macro writer understands that a macro definition is being replaced for sites where there are multiple macro authors. If the name provided in the macro's definition matches a current macro name, an error is returned if **REPLACE** is not specified.

The **OVERRIDE** option provides for replacing a FileManager command with a macro definition. Only FileManager commands can be replaced. Individual syntax items of a command cannot be replaced. The **OVERRIDE** option ensures that the macro author understands that the macro will replace a FileManager command. If the name provided in the macro's definition matches a FileManager command name, an error is returned if **OVERRIDE** is not specified.

The **RESPONSE** option provides control over the display of text from an auto macro. If **NONE** is specified, this text is not displayed. If **ALL** is specified, then all response text from an auto macro is displayed with the system messages. If **EXPLICIT** is specified, then only responses coded directly within the macro are displayed. A response may be coded in the macro body by specifying a quoted string as a macro statement. The default is **NONE**.

### Macro Body

The macro body is a list of macro statements that are evaluated when the macro is invoked. Each macro statement must evaluate to a valid FileManager command or macro primitive.

The text portion of the body of the macro is the non-variable part of the FileManager command(s) to be executed. The text is entered, as it would be at any of the FileManager command interfaces, with the need for the preceding FM. Depending on the syntax of the FileManager command, some parts of the text may need to be enclosed in quotes.

Use of parameters (<formal name>) within the body allows variable information to be included in the evaluated command. A parameter name can be any alpha-numeric identifier. When the command is evaluated, any instance of the string that was identified as a parameter, is replaced by the passed information for that parameter. It is important to choose parameter names carefully as names that also match command syntax items may produce unexpected results.

Macro primitives are MCP commands available within the macro. These macro primitives allow the macro to control aspects of a waiting entry. Macro primitives are designed specifically for use with auto macros. These commands are not available from normal FileManager command interfaces.

Intrinsic variables give the macro writer access to attributes of an outstanding tape assignment request. The various attributes of the request such as usercode of the program or the scratch pool requested, can be referred to symbolically. Intrinsic variables may be used wherever a value or string would be accepted, such as in a <selection spec> or <update spec>.

QUERY fields give the macro writer access to the results of a FM FIND command where the QUERY clause appears as the output destination of the FIND command. Any field of the current query database record can be accessed by giving the database field name prefixed by QUERY and a period (i.e. QUERY.FILEID). The requested item evaluates as a quoted string. As with intrinsic variables, query fields may be used wherever a value or string would be accepted. When a FM FIND:QUERY returns more than one database record, the information returned for the QUERY.<field name> refers to the first record found. If the .LIST option is used with QUERY, it evaluates to an ordered, comma separated list of quoted strings that contain the values of the specified database field for all records of the previous query. If the FIND:QUERY is not part of the macro definition, the results are undefined (usually a null string).

EVALUATE renders the <string> following it without the quotes. This is intended for use when some commands, such as the FA macro primitive, do not accept quoted parameters. (Example: FA SCRATCHPOOL = #EVAL #QUERY.POOLID)

## Using Macros

Invoking a macro is as simple as giving its name as a command (i.e. FM <macro name>). Macros may be invoked automatically as well as by text input. The automatic method is controlled by the REPLY macro option as described above.

## Examples

```
DEFINE FINDSPECIAL =
  BEGIN
  FIND WHERE POOLID = "SPECIAL" OR
  OWNER = "SPECIAL" OR
  CREATINGUSER = "SPECIAL" :PRINTER
  END FINDSPECIAL
```

## FileManager Macros

---

Macros can be used to make a long, often used command into a much shorter command. The FINDSPECIAL command will create a printed report of all cartridges that have the string SPECIAL as a Scratch Pool, Owner, or Creating Usercode.

```
DEFINE MYFIND <KEY> =  
  BEGIN  
  FIND LABEL <KEY>= SORT ON CREATINGUSER  
  :FILE MYFIND/RESULTS  
  END MYFIND
```

Parameters allow a single define to be used for multiple requests. The MYFIND command will create a file of all cartridges that have a label name that starts with the passed parameter.

```
DEFINE REPORTYESTERDAY =  
  BEGIN  
  REPORT ACTIVITY YESTERDAY;  
  REPORT EXCEPTIONS YESTERDAY  
  END REPORTYESTERDAY
```

Use macros to combine multiple commands into a single command. Issuing the command REPORTYESTERDAY will cause both an ACTIVITY and an EXCEPTION report for the previous day to be run.

```
DEFINE DBHOSTPOOL  
  : REPLY TO REQUIRES MT "DBBACKUP=" FILENAME SKIP =  
  BEGIN  
  FA SCRATCHPOOL=DBPOOL  
  END DBHOSTPOOL
```

Macros can be used to assign the SCRATCHPOOL attribute to a waiting entry. DBHOSTPOOL assigns the SCRATCHPOOL attribute using the FA macro primitive. It causes the FA (File Attribute) MCP command to be issued whenever a waiting entry is seen that is waiting for an output tape and the label starts with the characters DBBACKUP.

```
DEFINE BLPACKREL <SN>  
  : REPLY TO DISPLAY "BL/PACK(" <SN> "): RELEASED" SKIP =  
  BEGIN  
  MODIFY <SN> WITH EXPIRED = TRUE;  
  LOG "BLPACKREL RELEASED " <SN>;  
  END BLPACKREL
```

BLPACKREL allows FileManager to recognize the display produced by BL/PACK that releases a tape for reuse. The message template describes how to find the released serial number in the display message and to supply the found value as the <SN> parameter.

With this value, the macro uses the MODIFY command to force the tape to be expired in the database, permitting the tape to be purged and reused. The LOG step enters the release into the FileManager log file.

### Other Commands Used with Macros

#### **LOAD MACROS**

The LOAD MACROS command is used to load a file that contains macro definitions. See the LOAD Command in the FileManager Operations section.

#### **UNLOAD MACROS**

The UNLOAD MACROS command is used to remove all macros defines from FileManager. See the UNLOAD Command in the FileManager Operations section.

#### **DO**

The DO command is used to process FileManager commands contained in a disk file. The DO command is useful for loading macro defines particularly as part of a STARTUP file. See the DO Command in the FileManager Operations section.

#### **STATUS MACROS**

The STATUS MACROS command can be used to list the currently defined macros. See the STATUS Command in the FileManager Reports section.



# Chapter 7

## Batch Mode Execution

### Starting the Utility in Batch Mode

The FileManager utility may be executed in batch mode either by executing the utility from a WFL job, by file equating the utility's CARD file, or by setting the utilities TASKSTRING task attribute. If the utility any either of these conditions it will expect to receive its input through the CARD file or TASKSTRING task attribute. The utility is used for either FileManager tasks, or FileManager tasks. Starting the SYSTEM/TAPEMANAGER/UTILITY program with SW1 = TRUE will direct card file command to FileManager.

### Batch Mode Parameters for CARD file

In batch mode the utility reads commands from the CARD file. The CARD file is declared as kind READER but may be equated to a disk file. This file will only be opened if the TASKSTRING task attribute has not been set and the CARD file has been file equated to another file. The format of the commands is the same as those described in the previous sections with the following enhancements and restrictions:

- Each command must end with a semicolon (;).
- A command may be continued over many cards. This includes strings but intervening spaces may be included in the string if the first part of the string does not end in column 72 and the next part of the string does not begin in column 1.
- The percent sign (%) may be used to place comments in the batch command file. The utility will stop scanning the record when a "%" is found that is not in a quoted string.
- The CARD file may be equated to almost any file containing FileManager commands in EBCDIC. However, the utility will only scan the first 72 characters of a record. The remainder of the record will be ignored. Generally, a simple CANDE text file will be used as input.

### Dollar (\$) Option Card

A dollar (\$) option card may be used to control the processing of a batch Utility run. The dollar option card is implemented as follows.

- The dollar option card must be the first card in the deck.
- There can only be a single dollar option card.
- There must be a "\$" in column 1 or column 2 to identify a dollar option card.
- A list of options may be set or reset by specifying the SET or RESET keyword. If SET or RESET is not specified SET is assumed.
- The following options are available on the dollar option card.

## Batch Mode Execution

---

### **LIST dollar Option**

The LIST option controls the creation of the printed listing of the batch run of the Utility. If the LIST dollar option is set, all input and output is sent to the printer. Each command read by the utility will be printed along with the results of the command. The default for the LIST option is SET.

### **IGNORE dollar Option**

The IGNORE option controls the action taken by the Utility on either a syntax or processing error. If the IGNORE option is SET then the batch process will continue with the next command if a syntax error is found. Also if a process error (DSed) is detected with a FM WAIT(<task id>) the batch process will continue with the next command. If IGNORE is set the TASKVALUE of the FM Utility run is NOT set if there were any errors. If the IGNORE option is RESET then either a syntax or processing error will cause the batch job to abort and set the TASKVALUE to 1. The default for IGNORE option is RESET. RESUME is a synonym for IGNORE.

## **Batch Mode Parameters for TASKSTRING task attribute**

If the TASKSTRING task attribute has been set the utility will use that string as the command input. The format of the commands is the same as those described in the previous sections with the following enhancements and restrictions:

- Each command must end with a semicolon (;).
- The total length of the string must not exceed 254 characters. This is an MCP/WFL restriction.
- Comments (%) must not be used in the string.
- Dollar (\$) option cards can not be used.

## **Waiting for Asynchronous Commands to Complete**

Various FileManager commands cause asynchronous processes to be started. The command that starts the process(es) returns after the process has started but before the asynchronous process has completed. The following FileManager commands cause asynchronous processes to be processed; DATABASE (BACKUP, REORGANIZE, RESTORE).

By using the WAIT command and a <task id> associated with the asynchronous task, it is possible to pause execution until the asynchronous task has completed. A <task id> may be associated with a command by placing [<task id>] at the end of the command but before the semicolon (;). A <task id> can be any user defined alpha-numeric identifier up to 17 characters in length. The <task id> can then be used in a later WAIT command to pause the batch processing until the asynchronous task associated with the <task id> has completed.

See the WAIT command in Chapter 6 of this manual for additional information.

***Note:** The [`<task id>`] syntax is accepted for all FileManager commands but is ignored if the command does not process an asynchronous task.*

## Batch Mode Execution

---

### Examples

#### Executing the TapeManager utility from WFL

```
100 BEGIN JOB TEST/BATCH;
200 TASK TSK;
300 ?DATA CARD/TAPEMANAGER
400 % TEST RUN
500 FM REPORT ALL BY SN; % PRINT ALL BY SN
700 FM FIND SN 100100; % DISPLAY A TAPES ATTRIBUTES
800 FM OLDEST DAILYDUMP;
900 FM PURGE WHERE INLIBRARY AND EXPIRED [PGTASK];
1000 FM WAIT(PGTASK); % WAIT FOR THE PURGING TO FINISH
1100 FM DB      % MULTI CARD COMMAND
1200 BACKUP;
1300 ?
1400 RUN SYSTEM/TAPEMANAGER/UTILITY [TSK];
1500     FILE CARD(TITLE=CARD/TAPEMANAGER);
1600 IF TSK(VALUE) = 1 THEN
1700     ABORT "ERROR IN TAPEMANAGER BATCH RUN";
1800 END JOB.
```

#### Executing TapeManager Utility In Batch Mode From CANDE

```
RUN $SYSTEM/TAPEMANAGER/UTILITY;SW1; FILE CARD(KIND=DISK,
TITLE=TAPEMANAGER/BATCH/INPUT)
```

#### Executing TapeManager Utility In Batch Mode with TASKSTRING

```
100 BEGIN JOB TEST/TASKSTRING;
200 TASK TSK;
300 RUN SYSTEM/TAPEMANAGER/UTILITY [TSK];SW1
400     TASKSTRING="FM STATUS;" &
500         "FM FIND WHERE TAPECREATED = TODAY [STKTSK];"&
600         "FM WAIT (STKTSK);";
600 IF TSK(VALUE) = 1 THEN
700     ABORT "ERROR IN FILEMANAGER BATCH RUN";
800 END JOB.
```

**Note:** *The DO command may also be used to input a series of FileManager commands from a file.*

# Chapter 8

## Diagnostic Commands

### Caution

Dynamic Solutions International reserves the right to change the syntax and output of the diagnostics commands at any time without the normal advance notification.

## LOG Command

The LOG command allows the operator to place a maintenance log entry into the FileManager log file. A maintenance log entry allows for a time marker, event message, or information that would be useful in diagnosing a hardware or software problem.

### Syntax

```
FM — LOG — <text> _____|
```

### Explanation

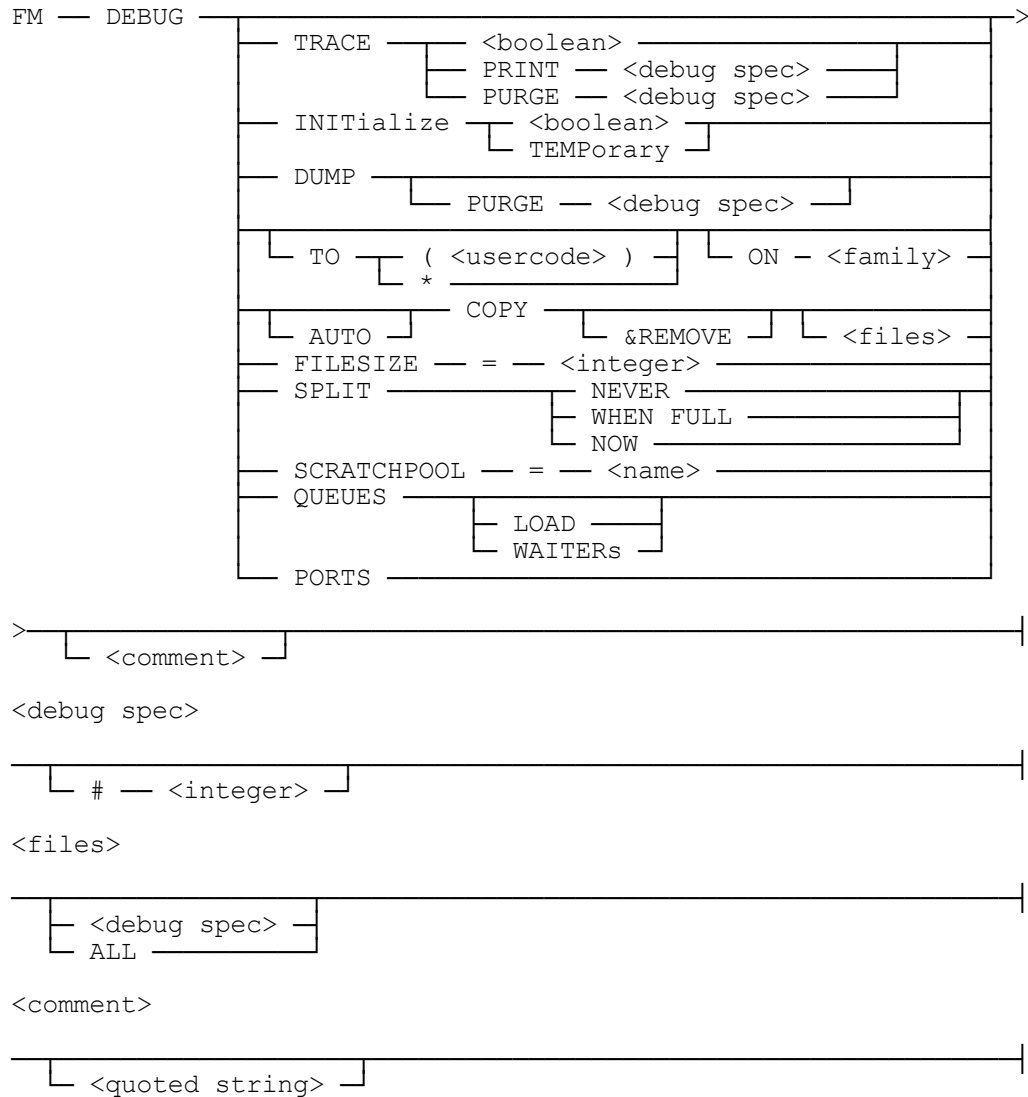
```
FM LOG *** START OF TEST ***
```

The LOG command allows the operator to place a text entry into the FileManager log file. These entries are shown in the reports generated by the LIST/REPORT MAINTENACE command and the LIST/REPORT LOG OPERATOR/ALL command.

## DEBUG Command

The DEBUG command controls the various capabilities of the FileManager and LibraryManager software modules diagnostic capabilities.

### Syntax



### Explanation

#### FM DEBUG TRACE ON “Test of MOVE command”

The DEBUG TRACE command causes the diagnostic tracing function to be activated (ON), deactivated (OFF), or its current status to be displayed. When tracing is active, information about the internal operations of the FileManager are written to a disk file.

The size of the disk file(s) is determined by the `DEBUG FILESIZE` command. If this record count is exceeded the file wraps so that the oldest information is overwritten. Each time tracing is activated a new set of trace files is created. If the LibraryManager software is also executing, separate trace files of the LibraryManager components are also produced. By default, the files are placed on the same family as the FileManager files. The files will be named as follows:  
<user name>/DEBUG/TRACE/<debug sequence #>/<module>/<date>/<time>. See the `DEBUG TO` and `DEBUG ON` commands below for additional information.

If a comment is included (highly recommended), that comment will be included in the trace output.

### **FM DEBUG INITIALIZE ON**

In some situations it may be necessary to have tracing be activated when the software initializes. Setting `INITIALIZE` to `ON` will cause the tracing to be activated the next time the software is started. When the `INITIALIZE` option is set to `OFF`, trace files will not be created unless a `TRACE ON` command is given.

### **FM DEBUG INITIALIZE TEMPORARY**

Setting the `INITIALIZE` option to `TEMPORARY` causes the FileManager to start tracing when the software is started. These files are not saved unless a subsequent `TRACE ON` command is issued. This option is useful for diagnosing intermittent problems that do not occur on a predictable basis so that trace files do not accumulate unnecessarily.

### **FM DEBUG TRACE PRINT #143**

The `PRINT` option of the `DEBUG TRACE` command will cause the specified trace file(s) to be printed. If a trace file set number is not given, the current trace file(s) are printed. The print attributes defined with the `CONFIGURE REPORT` command are used when printing trace files.

### **FM DEBUG TRACE PURGE #157**

The `PURGE` option of the `DEBUG TRACE` command is used to remove trace files from disk. All trace files in the specified trace set are removed. If a trace file set number is not given the current trace files are removed. If the specified trace files are active (`TRACE ON`), tracing is turned off before the remove.

### **FM DEBUG DUMP**

The `DEBUG DUMP` command causes a program dump of all FileManager stacks and libraries to be placed on disk. If the LibraryManager software is also executing, program dumps of the LibraryManager components are also produced. By default, the dump files are placed on the same family as the FileManager files. The files will be named as follows:

## Diagnostic Commands

---

<user name>/DEBUG/PDUMP/<debug sequence #>/<module>/<date>/<time>/<mix #>/<sequence #>. See the DEBUG TO and DEBUG ON commands below for additional information.

### **FM DEBUG DUMP PURGE**

The PURGE option of the DEBUG DUMP command is used to remove dump files from disk. All trace files in the specified dump set are removed. If a dump file set number is not given the current dump files are removed.

### **FM DEBUG TO (MYUSERCODE) ON MYPACK**

The TO and ON options control the placement of the trace and dump files. By default, trace and dump files are placed on the same family as the FileManager database and program files. The ON option can be used to change the pack family used to store these files. The TO option can be used to change the directory name under which the files are stored. By default, the directory name is the usercode under which the FileManager software is installed without the parenthesis (i.e. an installation usercode of (DSI) will cause the trace and dump files to be placed in the directory \*DSI/DEBUG/=). If FileManager was installed without a usercode or the TO option is used to set the debug usercode to \*, then the debug directory name is DEBUG/=.

### **FM DEBUG COPY & REMOVE # 152 “Load fail problem”**

The COPY option of the DEBUG command initiates a copy of trace and dump files to tape. The files collected will include some or all debug traces and dump files, the running set of code files, the FileManager database files, the tape library configuration file, and the current MCP code file. If a debug serial number is not specified, the current debug serial number is assumed. &REMOVE option causes the dump and trace files to be removed after they have been copied to tape. The tape created will be called TMDUMP. The copy job will request a scratch tape according to the settings of the DEBUG SCRATCHPOOL option.

If a comment is included (highly recommended), it is placed in a file that is included on the tape.

### **FM DEBUG FILESIZE = 28800**

The FILESIZE option allows the controlling of the size of trace files. This value represents the number of records to be stored in each trace file created. The default value is 14400 records. The value must be between 900 and 1,000,000. A change to this value will not affect a trace that is currently being taken, but will take effect the next time a trace is started.

### **FM DEBUG SPLIT WHEN FULL**

Setting this tracing option prevents the wraparound overwrite when a trace file fills up. Instead, a new set of trace files under the next debug serial number is created. Tracing picks up directly in the new files with no entries lost. Use the FILESIZE option to control the point at which the files will be split.

### **FM DEBUG SPLIT NEVER**

Setting this tracing option causes trace files to wraparound and overwrite older entries when full. This is the default behavior.

### **FM DEBUG SPLIT NOW**

Use this command to cause the trace files to be split immediately. Tracing must already be on.

### **FM DEBUG QUEUES**

The QUEUES option displays the queue depths of various internal queues used by the FileManager.

### **FM DEBUG QUEUES LOAD**

The QUEUES LOAD option displays the details of the load queue. This information may be useful to understand why a tape did not load when requested.

### **FM DEBUG QUEUES WAITERS**

The QUEUES WAITERS option displays the details of the waiters queue. This information identifies the tape selected by FileManager for assignment to each task.

### **FM DEBUG PORTS**

The PORTS option displays the current status and status history of the port files used by FileManager.

### **FM DEBUG**

The DEBUG command without any options will display the current settings of the debug options.

***Note:** Collection of DEBUG information takes additional processor and I/O time that may be noticeable on heavily loaded systems. It is therefore recommended that DEBUG information not be requested except as requested by DSI or to document a problem.*

# TRACE Command

The TRACE command is the same as the DEBUG TRACE command. It is retained to be compatible with previous releases.

### Syntax

```
FM — TRACE — <boolean> —
```

### Explanation

#### FM TRACE ON

The TRACE command causes the diagnostic tracing function to be activated (ON), deactivated (OFF), or its current status to be displayed. When tracing is active, information about the internal operations of the FileManager are written to a disk file. The size of the disk file(s) is determined by the DEBUG FILESIZE command. If this record count is exceeded the file wraps so that the oldest information is overwritten. Each time tracing is activated a new set of trace files is created. By default, the files are placed on the same family and under the same usercode as the FileManager files. The files will be named as follows:

```
DEBUG/TRACE/<debug sequence #>/<module>/<date>/<time>.
```

This command is the same as the FM DEBUG TRACE <Boolean> command.

#### FM TRACE

The TRACE command without any options will display the current settings of the TRACE/DEBUG options. This command is the same as FM DEBUG.

***Note:** Collection of TRACE information takes additional processor and I/O time that may be noticeable on heavily loaded systems. It is therefore recommended that TRACE information not be collected except as requested by DSI or to document a problem.*

### RESCUE Command

The RESCUE command is used to recover from a possibly hung FileManager system. This command is not a normal FM command, but rather an AX or HI command executed from an Operator Console or remotespo program. The rescue operation causes all active FileManager tasks to be aborted (DSed) with a program dump and restarted. Another task is then started to copy all the program dumps and related information to a tape so that it may be sent to Dynamic Solutions International (DSI) for analysis.

#### Explanation

<mix #> **AX RESCUE**

<mix #> **HI 911**

<mix #> **HI 999**

To start the rescue process, you must know the mix number of the SYSTEM/ FILEMANAGER/SUPPORT library. This can be found using the LIBS system command. For sites with many active libraries, LIBS NAME = FILEMANAGER= will result in a response with fewer selections.

Issuing the AX RESCUE, HI 911, or HI 999 to the SYSTEM/ FILEMANAGER /SUPPORT library will cause all active FileManager processes to take program dumps, be DSed, and restarted. In most cases this will clear up any hung condition.

Once the dump and restart process has completed, FileManager will start a process to copy the dumps, any related traces, the FileManager data files, and current MCP (needed to analyze the dumps). The resulting tape should be sent to the Dynamic Solutions International (DSI) support organization for analysis.

# ABORT Command

The ABORT command is used to bring down FileManager system in an emergency or if the normal QUIT command does not seem to function. This command is not a normal FM command, but rather an AX command executed from an Operator Console or remotespo program. The abort operation causes all active FileManager tasks to be aborted (DSed) and the FileManager support library to be DSed.

### Explanation

**<mix #> AX ABORT**

To start the abort process, you must know the mix number of the SYSTEM/FILEMANAGER/SUPPORT library. This can be found using the LIBS system command. For sites with many active libraries the command "LIBS NAME = FILEMANAGER =" will result in a response with fewer selections.

Issuing the AX ABORT to the SYSTEM/ FILEMANAGER /SUPPORT library will cause all active FileManager processes to be DSed.

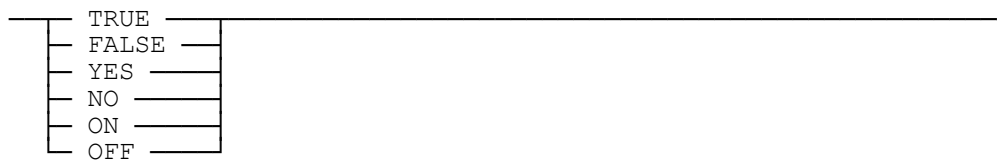
### **WARNING**

*Since ABORT DSes a library stack, any processes linked to that library stack will also be DSed. The RESCUE command is available if needed.*

# Appendix A

## Common Syntax Elements

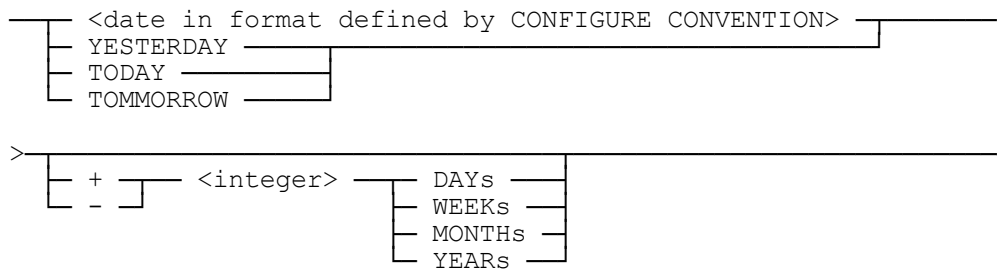
<boolean>



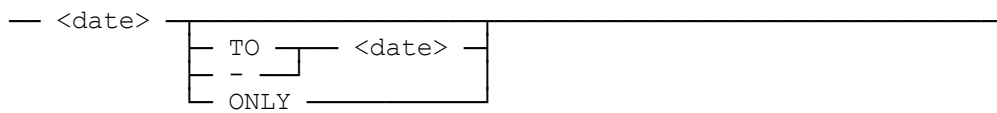
<compare op>



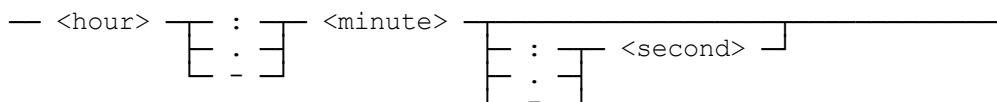
<date>



<date range>



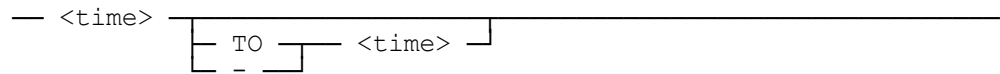
<time>



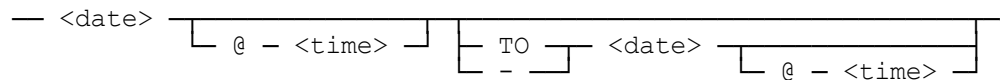
## Common Syntax Elements

---

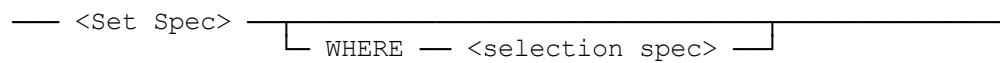
<time range>



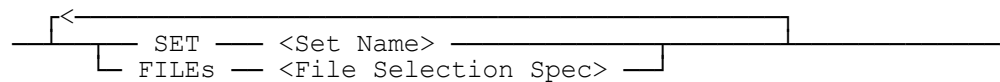
<date/time range>



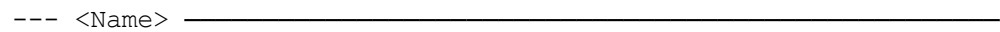
<Set Selection>



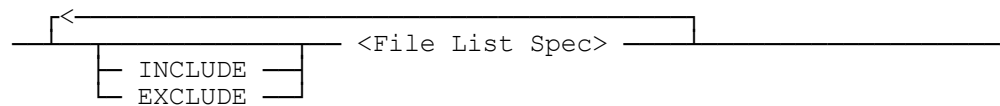
<Set Spec>



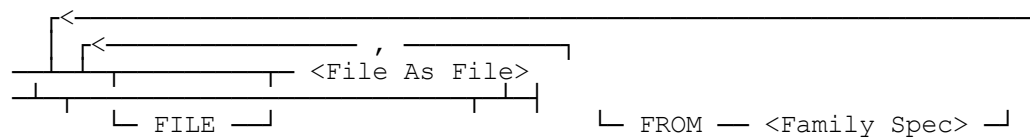
<Set Name>



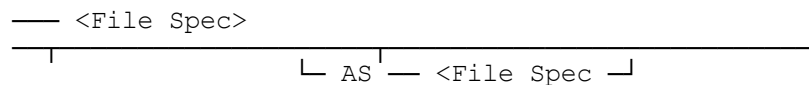
<File Selection Spec>



<File List Spec>

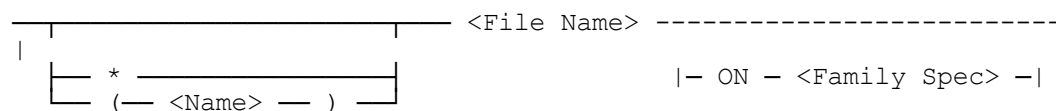


<File As File>



The AS <File Spec> branch is only allowed with certain commands that specifically allow it (eg. RESTORE).

<File Spec>



<Family Name>







# Appendix B

## FileManager Database Fields

FileManager has a data dictionary for use in accessing the fields in backup records. The field names are used in the WHERE specification of many commands. The fields are also used in the SORT specifications of reports and update specifications in the ADD and MODIFY commands. The field names are also used by the FIELD\_VALUE\_FM exported procedure for user interpretation of the FileManager database records. Items in parenthesis ( ) in the following descriptions are mnemonics which may be used in place of numeric values. Field items that show a location of 'virtual' do not actually reside in the database record. They are calculated from other database fields and information, but can be used as database fields. Virtual fields, by default, are read-only.

### Caution

Dynamic Solutions International reserves the right to change the layout and structure of the FileManager database at any time. The FIELD\_VALUE\_FM exported procedure is provided to shield the user from changes that may occur.

### Note

Many of the FileManager record items are MCP file attributes. For a more detailed explanation of MCP file attributes, consult the Unisys MCP File Attributes Manual, document number 8600-0064.

| Field Name   | Type | Location    | RW | Description  |
|--------------|------|-------------|----|--|
| ACCESS       | ts   | [7]         | RW | The last access time of the file in standard timestamp format. |
| ACCESSDATE   | int  | [7].[47:16] | R  | The last time the file was accessed, in Julian date format.    |
| ACCESSDATEUT | int  | Virtual     | R  | Julian date in universal time                                  |

## FileManager Database Fields

---

|              |     |             |    |   |
|--------------|-----|-------------|----|---|
| ACCESSTIME   | int | [7].[31:32] | R  | The last access time of the file, expressed in microseconds since midnight.                   |
| ACCESSTIMEUT | int | Virtual     | R  | The last access time of the file in Universal time, expressed in microseconds since midnight. |
| ACCESSTZ     | int | [63].[23:8] | RW | Integer value of the time zone. See appendix G of the Unisys MCP File Attributes manual.      |
| ALTER        | ts  | [8]         | RW | The timestamp of the last alter date of the file.   |
| ALTERDATE    | ts  | [8].[47:16] | R  | The last alter date of the file, in Julian date format.                                       |
| ALTERDATEUT  | ts  | Virtual     | R  | The last alter date of the file in universal time format.                                     |
| ALERTIME     | int | [8].[31:32] | R  | The last alter time of the file, expressed in microseconds since midnight.                    |
| ALERTIMEUT   | int | Virtual     | R  | The last file alter time expressed in Universal Time format                                   |
| ALTERTZ      | int | [63].[31:8] | RW | Integer value of the time zone. See appendix G of the Unisys MCP File Attributes manual.      |
| BACKUPKIND   | int | [31].[3:4]  | RW | 0 – Invalid<br>1 – FULL<br>2 – INCREMENTAL<br>3 – DIFFERENTIAL                                |
| COPIED       | ts  | [9]         | RW | Timestamp of the last time the file was used as a copy source (COPYSOURCE file attribute)..   |
| COPIEDDATE   | int | [9].[47:16] | R  | The date the file was copied, in Julian date format.  |
| COPIEDDATEUT | int | Virtual     | R  | Same as COPIEDDATE in Universal Time format.  |
| COPIEDTIME   | int | [9].[31:32] | R  | Time the file was copied, expressed in microseconds since midnight.                           |

## FileManager Database Fields

---

|                 |      |              |    |  |
|-----------------|------|--------------|----|--|
| COPIEDTIMEUT    | int  | Virtual      | R  | Same as COPIEDTIME, but in Universal Time format.  |
| COPIEDOK        | bool | [33].[47:1]  | RW | True if the file was successfully copied.  |
| COPYCOMPARE     | bool | [33].[46:1]  | RW | True if the LIB/MAINT COPY&COMPARE syntax was used for the backup.                       |
| COPYVERIFY      | bool | [33].[45:1]  | RW | True if the file was backed up via the LIB/MAINT COPY&VERIFY syntax.                     |
| COPYTZ          | int  | [63].[47:8]  | RW | Integer value of the time zone. See appendix G of the Unisys MCP File Attributes manual. |
| CREATION        | ts   | [10]         | RW | Creation date of the file in Timestamp format  |
| CREATIONDATE    | int  | [10].[47:16] | R  | The date of file creation in Julian date format.   |
| CREATIONDATEUT  | int  | Virtual      | R  | Date of file creation in Julian date format using universal time.                        |
| CREATIONTIME    | int  | [10].[47:32] | R  | Time the file was created, expressed in microseconds since midnight.                     |
| CREATIONTIMEUT  | int  | Virtual      | R  | Time of file creation in universal time, expressed as microseconds since midnight        |
| CREATIONTZ      | int  | [63].[39:8]  | RW | Integer value of the time zone. See appendix G of the Unisys MCP File Attributes manual. |
| TAPECREATED     | ts   | [30]         | RW | The creation timestamp of the tape containing the backup of this file.                   |
| TAPECREATEDDATE | int  | [10].[47:16] | R  | The date portion of TAPECREATED in Julian date format.                                   |
| TAPECREATEDTIME | int  | [10].[31:32] | R  | The time portion of TAPECREATED in microseconds since midnight..                         |

## FileManager Database Fields

---

|                 |     |               |    |  |
|-----------------|-----|---------------|----|--|
| FAMILYNAME      | str | [67..70]      | RW | The family name portion of the file title. The first byte is the length.                     |
| FILENAME        | str | [73..115]     | R  | The file name portion of the file title. The first byte of the field is the length in bytes. |
| FILENUMBER      | int | [65].[47:32]  | RW | The tape file number for this file.  |
| FILESIZE        | int | [64].[47:48]  | RW | The size of the file, in number of bytes.  |
| FIRSTRECORDID   | int | [116].[31:32] | RW | The first linked DB record in a series of DB records.  |
| LABEL           | str | [27..29]      | RW | The label of the tape containing the file, in substandard form..                             |
| NEXTRECORDID    | int | [117].[31:32] | RW | The next linked DB record in a series of DB records. (For files that span multiple reels)    |
| RECORDID        | int | [6].[31:32]   | R  | A unique number assigned by FileManager to identify a particular database record.            |
| RECORDMODIFIER  | int | [2].[39:8]    | R  | The reason for the last change:<br>*   |
| RECORDSTATE     | int | [2].[31:8]    | R  | The record state:<br>*   |
| RECORDTIMESTAMP | ts  | [1].[47:48]   | RW | Timestamp of the last record modification.   |
| RECORDTYPE      | int | [2].[47:8]    | R  | The type of DB Record:<br>*  |
| REELCOUNT       | int | [29].[23:24]  | RW | Number of tapes used during the backup.  |
| REELNUMBER      | int | [39].[47:24]  | RW | The actual reel where this file exists in the LIB/MAINT backup.                              |
| SECTORSIZE      | att | [33].[32:12]  | RW | Size in bytes of an MCP logical sector   |
| SECURITYTYPE    | att | [33].[3:4]    | RW | 0-Private<br>1-Public<br>2-Guarded   |

## FileManager Database Fields

---

|               |     |              |    | 3-Controlled  |
|---------------|-----|--------------|----|---|
| SECURITYUSE   | att | [33].[7:4]   | RW | 0-Secured<br>1-IN<br>2-OUT<br>3-IO  |
| SERIALNO      | str | [0]          | R  | Serial number of the tape holding the backup of the file.                     |
| TAPEDENSITY   | int | [38].[8:9]   | RW | The Density of the tape used.   |
| TIMESTAMP     | ts  | [32]         | R  | Timestamp attribute of the file.  |
| TIMESTAMPDATE | int | [32].[47:16] | R  | The Julian date portion of TIMESTAMP.   |
| TIMESTAMPTIME | int | [32].[31:32] | R  | The Time portion of TIMESTAMP, expressed in microseconds since midnight.      |
| TITLE         | str | Virtual      | R  | Constructed from USERCODE, FILENAME and FAMILYNAME. (eg. (UC)FILE/A ON FAM).. |
| USERCODE      | str | [45..47]     | RW | Synonym for USERNAME.   |
| USERNAME      | str | [45..47]     | RW | The username portion of the file title, expressed as a string.                |

# Appendix C

## Programmatic Interfaces

The FileManager Support library exports a number of functions that allow user written programs to interface with the FileManager system and its database. These should be accessed through the DSISUPPORT intermediate library.

### DSISUPPORT LIBRARY DECLARATION

In order to access these library procedures the library must be declared in your program. Declare the library as follows;

```
LIBRARY DSISUPPORT (LIBACCESS = BYFUNCTION,  
                   FUNCTIONNAME = " DSISUPPORT. " );
```

### FIELD\_VALUE\_FM Procedure

```
BOOLEAN PROCEDURE FIELD_VALUE_FM (FIELD_NAME, RECORD, FORM, VALU,  
                                  VALUE_TYPE) ;  
ARRAY  
    FIELD_NAME [0]  
    , RECORD [0]  
    , VALU [0]  
    ;  
REAL  
    FORM  
    , VALUE_TYPE  
    ;  
LIBRARY DSISUPPORT;
```

**Function:** The FIELD\_VALUE\_FM procedure allows the user to extract data from a FileManager record without needing to know the field location of the data.

**Usage:** The FIELD\_VALUE\_FM procedure is exported from the SYSTEM/FILEMANAGER/SUPPORT library.

|                    |                   |  |
|--------------------|-------------------|--|
| <b>Parameters:</b> | <b>FIELD_NAME</b> | (Input) the name of the field to be extracted from the RECORD. The name must terminate in a NUL (48"00"). The valid field names are listed in Appendix B of this manual. |
|                    | <b>RECORD</b>     | (Input) this is a FileManager database record that was extracted to a file with the FIND/INQUIRE command.  |
|                    | <b>FORM</b>       | (I/O) this field is currently not used and is set to zero.   |

## Programmatic Interfaces

---

**VALU** (Output) this parameter returns the value of the requested field. For all field types except string, the first word of the array returns the raw (binary) value of the field. The second and following words of this parameter contain an EBCDIC string representation of the field value delimited by a NUL. String fields start in the first word and continue until complete. Fields of type word are undefined and do not have a string representation. If needed this parameter will be **RESIZED** to contain all of the field value results.

**VALUE\_TYPE** (Output) this parameter returns the type of the requested field in bits [7:8]. For some fields the length of the string portion of the **VALU** parameter are returned in bits [47:16]. The possible field types are;

- 1 = integer
- 2 = boolean
- 4 = date
- 5 = timestamp (in **TIME(6)** format)
- 7 = string (delimited by a NUL)
- 9 = file attribute value (integer)
- 10 = word
- 15 = time (in **TIME(11)** format)

**Results:** The **FIELD\_VALUE\_FM** procedure returns a value of **FALSE** if the procedure completes correctly. If there is an error in returning the requested value, the procedure returns a **TRUE** with an error value in bits [11:8] of the return value.

**Possible errors:**

- 1 = the field name is unknown or badly constructed
- 2 = the field type was unknown. Contact DSI for a software correction/upgrade.
- 3 = the record passed is invalid
- 4 = the procedure faulted. Contact DSI for a software correction/upgrade
- 5 = a bad parameter was passed. Check that the **FORM** or **VALUE\_TYPE** parameters are not literals.

# Glossary

## A

### **attribute**

1. A quality added to text to make it stand out from surrounding text, such as underlining, boldface, subscript, superscript, or struck out text. Attributes can be combined so that a character or text has several attributes at the same time.
2. A configurable quality used to define a file or station and so on.

## C

### **command**

an instruction to a computer to perform a special task.

### **COMS**

Unisys A Series Communications Management System.

### **CSV**

Comma Separated Values is a file format that is used by spread sheet and other programs for importing and transferring data.

### **cursor**

The blinking underline or block on the screen that indicates where the next character can be entered. As characters are entered, the cursor moves to the right.

## D

### **DELETE**

Display data control key which enables you to delete the character at the current cursor position. Characters to the right of the cursor within the same field and on the same line shift one character position to the left.

## F

### **field**

1. An area on a screen or form in which data is displayed or entered. The delimiters of the field can be visible or invisible to the terminal operator.
2. A consecutive group of bits within a word or a component of a record that represents a logical piece of data.

### **File**

A named group of related records.

### **form**

A special screen containing prompts requesting information and empty form fields in which the requested information can be entered.

## Glossary

---

### H

#### **HOME**

A field cursor movement key that moves the cursor to the home position, the first unprotected character position of the screen. If the display has no unprotected fields or no fields at all, the home position is row 1, column 1.

### I

#### **Input**

Text typed into the computer.

### M

#### **MCP**

The Master Control Program or operating system of a Unisys A Series system.

#### **menu**

A special group of fields that show the user a set of options from which to choose.

#### **message boxes**

Message boxes are pop up windows that provide status and error messages.

#### **message line**

Displays the INSERT operator guidance message.

#### **mix number**

A number assigned to a job or task by the A Series operating system (MCP).

### P

#### **pack**

A disk drive on a Unisys A Series system.

#### **parameter**

A data item provided to or by a program or procedure (subroutine).

#### **Process**

1. Execution of a program or procedure.
2. A software application; activity or series of operations that produces specified results.

### R

#### **railroad diagram**

A graphic representation of the syntax of a command or statement.

### S

#### **screen**

An image that appears on the display area of a terminal or workstation prompting the user to enter data, displays information or presenting options from which to choose.

#### **scroll**

To move forward and backward within a list, within help text, or within other displayed items.

**session**

A session is a Logical connection between two units on a network.

**Specify**

A key on a Unisys T-27 (or compatible) terminal that, when pressed, sends a message containing the current cursor position to the host system.

**station**

The outer end of a communication line. A station can correspond to a single terminal connected on a single line, or several stations can be connected on a line.

**system**

Operating system

## T

**terminal**

I/O device designed to receive or send information in a network.

## U

**user**

Individual accessing the computer.

**User code**

The <string> used to identify the user to the host system.

**user interface**

The appearance of a program to a user.

## **Glossary**

---

# Index

<

<auto reply> · 84  
<body> · 84  
<boolean> · 103  
<compare op> · 103  
<date range> · 55, 103  
<date/time range> · 104  
<date> · 56, 103  
    example · 73  
<Family Name> · 104  
<File As File> · 104  
<File List Spec> · 104  
<File Name> · 105  
<file options> · 72  
<File Selection Spec> · 104  
<File Spec> · 104  
<FILESYSTEM> · 72  
<formal name> · 84  
<formal parameters> · 84  
<intrinsic variable> · 85  
<macro primitive> · 85  
<macro statement> · 84  
<message template> · 84  
<options> · 84  
<query field> · 85  
<selection spec> · 105  
<Set Name> · 104  
<Set Selection> · 104  
<Set Spec> · 104  
<SN list> · 105  
<sort order> · 58  
<string literal> · 84  
<string primary> · 85  
<string> · 85  
<task id> · 48, 92  
<text literal> · 85  
<text> · 85  
<time range> · 104  
<time> · 103, 105  
    example · 73  
<update spec> · 105

## Index

---

### A

ABORT  
  command · 102  
access code · 26  
ad hoc  
  report · 68  
ADD  
  command · 77  
ALL  
  command · 53  
  option · 72  
  report · 58  
ASCENDING · 73  
ATTRIBUTES  
  REPORT · 19  
AUDIT  
  DATABASE · 16  
  report · 67

### B

BACKUP  
  DATABASE · 15  
  DATABASE command · 32  
batch · 91

### C

CLOSE  
  DATABASE · 15  
  DATABASE command · 30  
CONFIGURE  
  CONVENTION · 13  
  DATABASE · 15  
  EMAIL · 17  
  REPORT · 19  
CONVENTION  
  CONFIGURE command · 13  
CSV  
  format · 69, 74

### D

database  
  status · 71  
DATABASE · 37  
  BACKUP · 32  
  CLOSE · 30  
  CONFIGURE command · 15

OPEN · 31  
RECOVER · 36  
REORGANIZE · 34  
RESTORE · 35  
DATE · 13  
    CONVENTION · 13  
DBOPEN · 48  
DEBUG  
    command · 96  
DEFAULT · 27  
DEFINE  
    command · 83  
DELETE  
    command · 78  
DESCENDING · 73  
DUMP · 97

**E**

EMAIL  
    command · 46  
    CONFIGURE command · 17  
EVALUATE · 87  
EXCEPTIONS  
    report · 60

**F**

FA · 88  
FIELD\_VALUE · 74  
FIELD\_VALUE\_FM  
    procedure · 113  
FILE · 47, 57, 70  
    option · 73  
FILESIZE · 98  
FIND  
    command · 72  
FORMAT · 69, 73

**I**

INITIALIZE · 97  
INSTALL · 5

**L**

LAST  
    command · 54  
LibraryManager  
    version · 75  
LIST

## Index

---

command · 55  
LOAD  
  MACROS command · 44  
log · 43  
LOG  
  command · 61, 62, 95  
  report · 62

## M

MAINTENANCE  
  report · 59, 61  
MARC · 7, 9  
MODIFY  
  command · 80

## N

NEWEST  
  command · 52

## O

ODT · 9  
OF  
  DATABASE · 15  
OLDEST  
  command · 51  
OPEN  
  DATABASE command · 31  
**OPERATOR** · 47  
OVERRIDE  
  macro option · 86

## P

PAGESIZE · 20  
PORTS  
  debug · 99  
PRINT  
  trace files · 97  
PRINTER  
  option · 74  
PRIORITY  
  REPORT · 19  
PRIVILEGED · 27  
PUBLIC · 27  
PURGE  
  dump files · 98  
  trace files · 97

**Q**

QUERY · 87  
  option · 74  
QUEUES · 99  
QUIT  
  command · 45

**R**

RAW  
  format · 70, 73  
RECOVER  
  DATABASE · 16  
REMOTESPO · 9  
REORGANIZE  
  DATABASE command · 34  
  report · 65  
REPLACE  
  macro option · 86  
REPLY  
  macro option · 86  
REPORT  
  command · 55  
  CONFIGURE command · 19  
RESCUE  
  command · 101  
RESPONSE  
  macro option · 86  
RESTORE  
  DATABASE command · 35, 36

**S**

SECURE  
  command · 25  
SITENAME  
  REPORT · 19  
SKIP · 86  
SORT · 69, 73  
STATION · 26  
STATUS  
  command · 71  
  DB · 79  
SUMMARY  
  report · 66  
SUPERVISOR · 46  
SUPPORT · 47

## Index

---

### T

TapeManager  
  status · 71  
TEMPORARY · 97  
TEXT  
  format · 69, 74  
TIME · 13  
  CONVENTION · 13  
TITLE · 69  
TL  
  command · 43  
TRACE · 47, 96, 100  
  command · 100

### U

UNDO · 78  
UNIT · 26  
USER · 26

### V

VERSION  
  command · 75

### W

WAIT · 92  
  command · 48  
WHERE · 69, 78, 80  
WITH · 77, 80

# Document Evaluation Form

DSI is interested in receiving your comments and suggestions regarding this document. Comments will be utilized in subsequent revisions to improve this document.

Manual Title:

**FileManager Operations Guide  
for  
Unisys MCP Systems**

Version No: **8.068**

Date: **September 2009**

**Please check type of suggestion:**

Addition       Deletion       Revision       Error

**Comments / Suggestion:**

---

---

---

---

---

---

---

**From:**

Name: \_\_\_\_\_

Title: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

City/ST/ZIP: \_\_\_\_\_

Phone: (\_\_\_\_)\_\_\_\_-\_\_\_\_ Date: \_\_\_\_/\_\_\_\_/\_\_\_\_

**Remove form and mail preaddressed overleaf, or FAX to:**

Dynamic Solutions International (DSI)  
Product Development Group

FAX Number: (303)754-2020

FOLD HERE AND TAPE

---

PUT  
STAMP  
HERE

**Dynamic Solutions International (DSI)**  
1 Inverness Drive East  
Englewood, Colorado 80112  
U.S.A.

Attn: Product Development Group