



Rocket Model 204 SirFile

User's Guide

November 2014
FIL-0705-UG-01

Notices

Edition

Publication date: November 2014

Book number: FIL-0705-UG-01

Product version:

Copyright

© Rocket Software, Inc. or its affiliates 1995-2014. All Rights Reserved.

Trademarks

Rocket is a registered trademark of Rocket Software, Inc. For a list of Rocket registered trademarks go to: www.rocketsoftware.com/about/legal. All other products or services mentioned in this document may be covered by the trademarks, service marks, or product names of their respective owners.

Examples

This information might contain examples of data and reports. The examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

License agreement

This software and the associated documentation are proprietary and confidential to Rocket Software, Inc. or its affiliates, are furnished under license, and may be used and copied only in accordance with the terms of such license.

Note: This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when exporting this product.

Corporate Information

Rocket Software, Inc. develops enterprise infrastructure products in four key areas: storage, networks, and compliance; database servers and tools; business information and analytics; and application development, integration, and modernization.

Website: www.rocketsoftware.com

Rocket Global Headquarters
77 4th Avenue, Suite 100
Waltham, MA 02451-1468
USA

Contacting Technical Support

If you have current support and maintenance agreements with Rocket Software and CCA, contact Rocket Software Technical support by email or by telephone:

Email: m204support@rocketsoftware.com

Telephone :

North America +1.800.755.4222

United Kingdom/Europe +44 (0) 20 8867 6153

Alternatively, you can access the Rocket Customer Portal and report a problem, download an update, or read answers to FAQs. You will be prompted to log in with the credentials supplied as part of your product maintenance agreement.

To log in to the Rocket Customer Portal, go to:

www.rocketsoftware.com/support

Contents

Proprietary Notices	iii
Contents	v
Chapter 1: Introduction to UL/SPF	1
UL/SPF packaging and installation requirements	2
Integrating UL/SPF with other subsystems	2
Related documentation	3
Chapter 2: Overview and Setup for SirFile	5
Files available for SirFile Monitoring	6
File enqueueing	6
Getting Started	7
Chapter 3: The Refresh Process	11
Chapter 4: Main File Menu	13
Chapter 5: File Thresholds Screen	17
Chapter 6: File Open Screen	21
Chapter 7: File Detail Screen	23
Chapter 8: File Tables Screen	25
Chapter 9: Messages Screen	29
Chapter 10: Background Monitor Specification Screen	31
Chapter 11: Invocation Techniques for Background Monitoring	33
Chapter 12: Running SirFile under BATCH204	35
Appendix A: File statistics definitions	37
Appendix B: Date Processing	53

Index **55**

Introduction to UL/SPF

UL/SPF is a family of products implemented as User Language application subsystems and designed to function together in a *Model 204* Online. Each product in the family can be installed and run independently, or each can be installed and run as a component of the integrated *UL/SPF* (User Language / System Productivity Facilities) framework.

All *UL/SPF* products share a common "look and feel" that is modelled upon IBM's ISPF facilities. *UL/SPF* supplants the sometimes arcane command language of *Model 204*, providing a highly productive full screen interface to a variety of common *Model 204* functions. *UL/SPF* enhances the capabilities of *Model 204* and bypasses many of its restrictions, enabling the performance of routine tasks that were previously impossible or prohibitively time-consuming.

UL/SPF comprises the following products:

- SirDBA*** A system that analyzes *Model 204* databases to determine their logical structure, populating an internal catalog. *SirDBA* is distributed as a component of the *Sir2000 Database Analysis Tools*.
- SirFile*** A comprehensive facility both for monitoring the physical storage utilization of *Model 204* database files and for warning users of the need for file reorganizations. *SirFile* maintains historical information that allows it to predict when file sizing problems *will* occur, allowing a DBA to take preventative action before an application outage results.
- SirLib*** A system that provides change management and configuration control for *Model 204* User Language applications. Fully integrated within the programming environment, *SirLib* supports unique *Model 204* constructs such as file groups, while remaining nearly transparent to programmers.
- SirMon*** A comprehensive facility for monitoring the performance and availability of *Model 204* online systems. *SirMon* combines the real time monitoring of *Model 204* performance with intelligent full screen displays that facilitate System Manager duties.
- SirPro*** A collection of powerful and easy to use tools for programmers, database administrators, and application managers. *SirPro* provides programmers with powerful facilities for managing large libraries of User Language procedures, and it provides system managers with intuitive ISPF-like front ends to many *Model 204* system management commands.
- SirScan*** A high performance utility that allows users in a *Model 204* Online to browse the contents of its journal in real time. *SirScan* permits ordinary

users to view journal entries generated by their own online session, and it allows users in ADMIN SCLASSES to browse journal entries for any set of users. The data is displayed in a full-screen browser with powerful searching commands and filtering options.

In addition, a number of subsystems that are not linked into the *UL/SPF* menuing structure may be accessed via APSY-transfer from the *UL/SPF* applications. One such subsystem is **FACT**, a utility for browsing *SirFact* dumps.

There are also many sample web and client-server applications developed for the Janus product family that are distributed and installed along with the *UL/SPF* products. These include **JANCAT**, an application that builds normalized views of *Model 204* data for use by *Janus Specialty Data Store* applications, and **JANSSL**, a system for creating and managing SSL certificate requests.

1.1 UL/SPF packaging and installation requirements

All of the *UL/SPF* products are *Model 204* application subsystems written in User Language. *UL/SPF* is distributed as a set of *Model 204* files in a backup format produced by the *Model 204* DUMP command. All User Language based products are distributed in a single *Model 204* procedure file called **SIRIUS**.

UL/SPF makes extensive use of specialized User Language \$functions that enable the creation of User Language application systems that can support complex environments with minimal server size requirements. Prior to version 7.5 of *Model 204*, the \$functions were part of the *Sirius Mods*, the installation of which (*Sirius Mods Installation Guide*) was a prerequisite for any *UL/SPF* product.

The *UL/SPF* \$functions are included in the [http://m204wiki.rocketsoftware.com/index.php/List_of_\\$functions](http://m204wiki.rocketsoftware.com/index.php/List_of_$functions), and their use is controlled by a product authorization mechanism. Many functions require that a specific *UL/SPF* product be authorized. Some functions may only be invoked from an authorized procedure “signed” by Sirius Software or Rocket Software. The **SIRIUS** command examines the status of all Sirius products installed in a *Model 204* Online.

1.2 Integrating UL/SPF with other subsystems

UL/SPF and any of its constituent products can be easily integrated with other User Language subsystems. Whenever a *UL/SPF* component product is exiting, it first checks to see if the global variable **SIRIUS.COMM** exists and has a non-null value. If so, the *UL/SPF* product performs a subsystem transfer using the value in **SIRIUS.COMM** as the name of the target subsystem.

For example, the following code fragment lets you transfer into SIRMON. When SIRMON exits, control is transferred to the application subsystem MENUSYS, provided that NEXTPROC is the current subsystem's communication global variable:

```
%RC = $SETG('SIRIUS.COMM', 'MENUSYS')
%RC = $SETG('NEXTPROC', 'XFER')
%RC = $SETG('XFER', 'SIRMON')
STOP
```

In addition to the individual User Language subsystems that implement the *UL/SPF* products, an umbrella *UL/SPF* subsystem, **ULSPF**, provides a menu that contains entries for all *UL/SPF* components installed at a site.

The *UL/SPF* subsystems support fast path navigation. For example, a *SirMon* user can transfer into *SirPro* Option 1 by typing:

```
=M.1.1
```

1.3 Related documentation

The following documentation is available from <http://docs.rocketsoftware.com> (M204 folder) or from <http://m204wiki.rocketsoftware.com>:

- ***SirDBA User's Guide***
- ***SirFile User's Guide***
- ***SirLib User's Guide***
- ***SirMon User's Guide***
- ***SirPro User's Guide***
- ***SirScan User's Guide***
- ***Sirius Mods Installation Guide***
- ***Model 204 System Manager's Guide***

The following documentation is available from the Model 204 documentation wiki (<http://m204wiki.rocketsoftware.com>):

- UL/SPF installation
(http://m204wiki.rocketsoftware.com/index.php/UL/SPF_installation_guide)
- \$functions
(http://m204wiki.rocketsoftware.com/index.php/M204wiki_main_page#.24Functions)
- Model 204 commands
(http://m204wiki.rocketsoftware.com/index.php/List_of_Model_204_commands)
- Model 204 files
(http://m204wiki.rocketsoftware.com/index.php/Category:File_architecture_and_management)
- UL/SPF product messages
(http://m204wiki.rocketsoftware.com/index.php/Category:Sirius_Mods_messages)
- User Language/SOUL (<http://m204wiki.rocketsoftware.com/index.php/Category:SOUL>)

Overview and Setup for SirFile

SirFile is a comprehensive facility for monitoring the physical storage utilization of *Model 204* database files and warning users of the need for file reorganizations.

SirFile requires only a single screen of setup information to determine “thresholds,” after which it automatically performs a pass through the online database files, checking to see if any file has exceeded a threshold. The process that evaluates files is called the **refresh process**. The refresh process performs two types of checks:

- It compares the current state of each file against the threshold settings to see if any file statistic is above the critical value for that statistic.
- It maintains a database of table usage information which it uses to determine rate of growth and to predict when thresholds will be exceeded.

During the initial database load, all files added to the *SirFile* database are compared to the system default thresholds. After the initial load, the user may override the system default thresholds with file-specific thresholds for files that require higher or lower levels of monitoring.

If any file table is above a threshold, *SirFile* flags the file with a warning, causing it to be highlighted in *SirFile*. Once *SirFile* has stored at least one sample snapshot of a file, it begins calculating the date at which each table will fill — assuming any table is showing growth. If the predicted fill date falls within a user-settable number of days, *SirFile* posts a prediction to the file, and the file is highlighted in the *SirFile* screens. When a prediction or warning is posted using a periodic refresh, a message can also be sent to the audit trail, as well as to a list of user IDs.

The file parameters that are evaluated for table-full status are ARETRIES, BFULLP, CRETRIES, CFULLP, DFULLP, EXTNADD, OVFLADD, EOVLADD, and SPILLADD.

The database refresh portion of *SirFile* is quite efficient, however it is the most expensive operation in *SirFile*. Therefore, several options are provided for determining when to perform a refresh, and for determining when to calculate CFULLP, generally the most expensive of the refresh calculations.

For more information, see:

- [“The Refresh Process” on page 11](#), for a complete description of the refresh process
- [“Invocation Techniques for Background Monitoring” on page 33](#), for methods that invoke a periodic refresh

- “Running SirFile under BATCH204” on page 35, for a method that can be useful for calculating CFULLP

The *SirFile* refresh process is always in “learn” mode. Any file that is open when a *SirFile* refresh is run is automatically added to the *SirFile* database, using the default thresholds for comparison.

2.1 Files available for SirFile Monitoring

For *SirFile* to collect information on a file, it must be open in the region by some user or subsystem. To guarantee that certain files are included in the initial load, those files can be opened manually before initiating the refresh process. Because storing user-entered passwords would violate most sites' security, *SirFile* uses a variety of methods to make files available to itself without needing to store passwords.

1. If a file is open in the region, or *SirFile* manages to open the file with default privileges, the file is added to the database and no user intervention is required.
2. The first time *SirFile* collects data for a file it adds the file to its Application Subsystem definition (with low privileges in the “USER” SCLASS and high privileges in the “ADMIN” SCLASS). Thereafter, no password is required for *SirFile* to open the file. Files that are prefixed “CCA” are exempted from this operation, as these files cause problems in SUBSYSGMT.
3. If *SirFile* has a file in its database that requires a password, and a non-periodic refresh is running (other than SIRFILE BATCH), you are prompted for the password.
4. If SIRFILE BATCH or a periodic refresh is running, *SirFile* skips any file it cannot open.

2.2 File enqueueing

Besides its APSY procedure file and the *UL/SPF* file SIRLOCAL, *SirFile* requires update access to CCASYS. All other files, even if they are defined to *SirFile* as optional APSY files, are closed and freed from the subsystem at the end of each user's *SirFile* session. This is done to reduce enqueueing problems.

2.3 Getting Started

Installation of *SirFile* is described in http://m204wiki.rocketsoftware.com/index.php/UL/SPF_installation_guide. *SirFile* should be installed as a private application subsystem. To access the system, a privileged user types “SIRFILE” on the *Model 204* command line of the region containing the software. An initial System Default thresholds screen is presented.

```

----- * * * Set threshold values for file warnings * * * -----
==> File ==> *DEFAULT

                Thresholds                                Number of historical
                ARETRIES ==> 1                             records to keep per file.
                BFULLP ==> 90                               ==> 99999
                CFULLP ==> 80                               Minimum number of days
                CRETRIES ==> 100                           between stored samples.
                DFULLP ==> 90                               ==> 7
                EXTNADD ==> 5000
                OVFLADD ==> 5000
                EOVLADD ==> 5000                            # of days advance warning
                SPILLADD ==> 1000                           on threshold exceeded.
                                                         ==> 7

Collect CFULLP only on specified files (Y/N)? ==> N
-----
1/Help          3/Quit
                9/Repeat
                12/Save-Refresh

```

System Default thresholds screen

This screen is also presented by pressing **PF11** from either the Main File menu or from the File Tables screen, if the cursor is not on a file name or file statistics line, respectively. The System Default thresholds screen is initially populated with a set of defaults that are adequate for most file monitoring needs. The thresholds are:

ARETRIES

Table A Retries indicate a failure of the Table A hashing algorithm to find a slot on a Table A page to store field information. Table A is usually the smallest table in a *Model 204* database file, but it is accessed every time a field name is referred to in User Language or IFAM code, so it is important that Table A operate as efficiently as possible. In most cases, *any* ARETRIES are too many, and for this reason, the ARETRIES default is very small.

BFULLP Percentage of Table B full. This is the percentage of record slots currently used, calculated as $BHIGHPG/BSIZE$. Table B holds the actual data in a *Model 204* file and is usually the largest table.

CRETRIES

Table C hashing algorithm retries. This statistic indicates that Table C, a hashed data structure, is becoming full, so the algorithm which determines placement of new pointers in the table must run repeatedly in order to find an empty slot. There is no fixed measure of how many retries indicate a critical condition for Table C.

CFULLP

Percentage of Table C full. Like Table A, Table C begins to exhibit hash retries when the data in it reaches a certain density. There is no strict formula for the maximum fullness of Table C for a given file, but a rule of thumb is that retries will begin to appear when Table C is 70 to 80% full. CFULLP is a more accurate measure of Table C full status than CRETRIES, but it is very expensive to calculate. *For this reason, unless you determine CFULLP calculation file-by-file, it is only calculated by the SIRFILE BATCH command.*

DFULLP

Table D percentage full. Table D can fill either as a result of index information or User Language procedures. Percentage full in Table D is calculated as the number of D pages used (DPGUSED) divided by DSIZE.

EXTNADD

Number of Extension Records. Extension records occur when the information added to an existing record exceeds the usable space left on a Table B page. Extension records are not a sign of a file full condition pending, and in some record structures they may be unavoidable. For this reason the EXTNADD threshold may be set very high. However, extension records cause inefficiencies in storage and retrieval, and for some file designs should be taken as a sign that Table B parameters are out of balance.

OVFLADD

Addition of records to overflow areas. Overflow records may occur in either sorted or hash files.

EOVFLADD

Addition of records to extra overflow areas. Records may be added to Extra Overflow areas in sorted files when the Overflow area is full.

SPILLADD

Number of records that have “spilled” from the preferred overflow area into secondary overflow areas or the overflow area prior to the appropriate sort group. Spilled records are stored and retrieved less efficiently than other records.

Number of historical records to keep

Number of sample records to keep for each file. *SirFile* requires at least 1 sample to be stored for any given file in order to make predictions about file-full conditions. The default value of 99999 records means that *SirFile* will

never throw away old data, as long as the file CREATE date does not change.

Minimum number of days between stored samples

Minimum number of days that must pass between samples being stored for any particular file. This setting allows users to run the refresh as often as they like without *SirFile* storing excessive amounts of data. The default of 7 ensures that no more than one record a week is maintained for any file unless that file breaks a threshold.

Days Warning

Number of days prior to a predicted “threshold exceeded” condition *SirFile* should post a warning. During a Refresh, *SirFile* uses the current file table values and the historical file information to calculate when the file will exceed each threshold (if positive growth is detected in a threshold value). If the number of days predicted falls within this value, a prediction and message are posted to the file, and the file appears highlighted on the Main File menu.

Collect CFULLP only on specified files?

Specifying “Y” to this prompt allows *SirFile* to collect the CFULLP statistic only on files specified by the user. Collecting CFULLP requires a TABLEC command, which can be very I/O intensive.

- When “Y” is specified at this prompt, the user must specifically request each file the TABLEC command is to be run against, entering Y at the following prompt in the File thresholds screen:

Collect CFULLP for this file

- When “N” is specified at this prompt, CFULLP calculations are performed only by the SIRFIELD BATCH command.

The user may accept the defaults or change any of the supplied settings. Pressing **PF12** saves the settings and initiates the *SirFile* database Refresh process.

SirFile records are fairly small and you shouldn't be too concerned about minimizing the amount of stored data: A year's worth of data for a single file will occupy only about 1 *Model 204* page if a record is kept per week.

The following commands are allowed:

OPEN xxxxxxxx Opens a file. As data can only be collected for files that *SirFile* already knows about, or files that are currently open by any user, a *SirFile* user may wish to manually open a file before running a refresh to guarantee the file is included in the sample.

CLOSE xxxxxxxx Closes a file. Any file that the *SirFile* user has open can be closed, except SIRFILE, SIRLOCAL and CCASYS (the required application subsystem files for *SirFile*).

PF keys:

- PF1** Accesses full-screen help.
- PF3** Return to command level or previous screen.
- PF9** Repeats the last command-line command.
- PF12** Saves the file specific thresholds and invokes the *SirFile* database refresh operation.

The Refresh Process

The Refresh process checks file tables and updates the *SirFile* database. You can request a refresh operation to run once or periodically.

1. Running a refresh once always runs in the *Model 204* thread of the user requesting the refresh, and can be specified either for a single file or for all files that *SirFile* can access. This can be invoked via:
 - The Main File menu, using either the ADD or REFRESH command or using **PF12**
 - Any of the File Tables, System Default thresholds, or File thresholds screens, using **PF12**
 - The SIRFILE BATCH command. The SIRFILE BATCH command is usually invoked only in a BATCH204 jobstream, to be run when files are not open for update by any online. See [“Running SirFile under BATCH204” on page 35](#).
2. A periodic refresh, which repeats at a specified time interval, is also called the **Background Monitor**. This can be invoked via:
 - The Background Monitor specification screen, using **PF5**. This runs the refresh on an sdaemon thread, that is, a different thread from the requesting user.
 - The **SIRFILE BACKGROUND interval** command. This runs the refresh on an sdaemon thread, that is, a different thread from the requesting user, and it may be useful in the online's User 0 stream.
 - The **SIRFILE SDAEMON interval** command. This runs the refresh on the thread of the requesting user, and it is most often invoked by a BATCH2 or IODEV=3 thread.

[“Invocation Techniques for Background Monitoring” on page 33](#) and [“Running SirFile under BATCH204” on page 35](#) further discuss considerations of periodic refreshes.

Whenever the refresh operation is performed, it can consume considerable resources if it calculates the CFULLP (Table C percentage full) value. One of the values specified on the System Default thresholds screen is whether to allow file-by-file determination of CFULLP calculation. If file-by-file CFULLP determination is allowed, then CFULLP is calculated during any type of refresh, for exactly those files which are specified as having CFULLP calculation performed. If CFULLP calculation is not determined on a

file-by-file basis, then CFULLP calculation is performed for all files by the SIRFILE BATCH command, and is not performed by any other type of refresh.

Messages to users and the audit trail are only sent during a periodic refresh.

The Refresh process makes a list of all files already in its database, then adds to the list any files that are currently open in the online. Then it runs through the following process for each file on the list:

1. It checks if a file-specific threshold exists. If not, it compares the file to the system default thresholds.
2. If any value in the file exceeds a threshold value, a record is stored for the file, and a warning is posted which causes the file to be highlighted on the Main File menu.
3. If the file has not exceeded any thresholds and at least 1 sample record has been stored for the file, *SirFile* performs a simple linear regression to predict when the file will fill. If the file-full date falls within the number of days of advance warning (as shown on the File threshold screen), a prediction and a message is posted to the file. The file will now appear highlighted on the Main File menu and the File Tables screen.
4. If no prediction or warning is posted to the file, *SirFile* performs a cleanup of any old warnings, predictions, or messages for the file.
5. If the refresh process is running online, and *SirFile* encounters a file in its database that it cannot open — either because it did not get added to its APSY definition, or because the file is not allocated to the region — the user is prompted for a password.

CHAPTER 4 *Main File Menu*

The Main File menu is presented on entering *SirFile* any time but the first; it is also presented when **PF3** is pressed from most *SirFile* screens. The first time entering *SirFile*, the System Default thresholds screen is presented, an initial database load is performed, then the Main File menu is presented. If any file has a warning or prediction posted against it, it appears highlighted on the Main File menu. Because certain terminal devices will not support extended display attributes, a pointer is also placed to the left of any highlighted file name.

```

----- * * * Sirius File Monitor * * * -----
==> _                                     Files: 38

  1 ALANPROC => 17 MOVE875      33 SIRSCAN
  2 ALANWEB   18 M204DCTL     34 SIRULSPF
  3 ALEXPROC  19 M204PROC     35 SIRXREF
  4 CCASYS    20 M204TEMP     36 TOMPROC
  5 CHARPROC  21 PUBS2       37 TOMWEB
  6 DATALINK  22 SIRDBA      38 TOOLS
  7 DEVFILE   23 SIRDBAD
  8 DMEDTC    24 SIRFILE
  9 DMEPROC   25 SIRFILED
 10 DMEWORK   26 SIRFIXES
 11 DMEWORK1  27 SIRLIB
 12 DMEWORK2  28 SIRLIBD
 13 JANCAT    29 SIRLIBP
 14 JANSSL    30 SIRLOCAL
 15 JANUS     31 SIRMON
 16 METADATA  32 SIRPRO

----- ULSPF.016: Highlighted file names have a warning posted against them.
1/Help  2/Background  3/Quit  4/V TABLES  6/Msgs
7/Up    8/Down        9/Repeat 10/Detail 11/Thresholds 12/Refresh

```

Main File menu

As many as 64 files are shown at once on the Main File menu. The following commands are allowed:

- n** Where **n** is any number. Scrolls the display forward the specified number of files.
- n** Where **n** is any number. Scrolls the display backward the specified number of files.
- F xxxxxxxx or /xxxxxxx** Searches forward in the file list for the requested file name "xxxxxxx".
- F xxxxxxxx or -/xxxxxxx** Searches backward in the file list for the requested file name "xxxxxxx".

Top	Moves the display to the top of the file list.
Bottom	Moves the display to the bottom of the file list.
M	When using PF7 or PF8 , scroll up or down to the top or bottom of the list of files.
ADD xxxxxxxx	Add the specified file to the database. <i>SirFile</i> attempts to open the file. If the file cannot be opened the user is prompted for a password on the separate password screen. Once the file is open <i>SirFile</i> analyzes the file tables, stores a file record and returns to the Main File menu, where the new file will appear.
REFRESH xxxxxxxx	REFRESH performs the same function as the ADD command if the specified file is not already in the database. If the file <i>is</i> in the database its tables are checked, and a record is stored for the file if the specified number of days has passed since the last sample, or if a threshold has been exceeded.
OPEN xxxxxxxx	Opens a file. As data can only be collected for files that <i>SirFile</i> already knows about, or files that are currently open by any user, a <i>SirFile</i> user may wish to manually open a file before running a refresh to guarantee the file is included in the sample.
CLOSE xxxxxxxx	Closes a file. Any file that the <i>SirFile</i> user has open can be closed, except SIRFILE, SIRLOCAL and CCASYS (the required application subsystem files for <i>SirFile</i>).
DELETE xxxxxxxx	Removes the file from the <i>SirFile</i> database. It is never necessary to delete a file from <i>SirFile</i> , but if a file is obsolete, or is never allocated to the online region, the <i>SirFile</i> administrator may find it inconvenient to always be prompted for passwords when the file cannot be opened during a refresh. DELETE removes all records associated with the file: warnings, history and thresholds.
TIME xxxx yy	Puts the screen into auto-refresh mode. The screen will refresh xxxx times every yy seconds. This command does not invoke the refresh process each time it wakes up, so it is only useful if a background task (or some other user) is running occasional refreshes, and this screen is used to view the changed information.

PF keys:

- PF1** Accesses full-screen help.
- PF2** Accesses the Background Monitor specification screen. This screen controls background monitoring, and specifies actions taken when the monitor notices file-full-pending conditions.
- PF3** Return to command level or the *UL/SPF* main menu.
- PF4** Accesses a scrollable list that displays the table-full status of files currently open by any user or subsystem.
- PF6** Accesses a screen of messages posted to the file. The only messages posted to files are prediction messages for when a file will fill. Only files that are highlighted on the Main File menu will have messages posted to them. If a file is highlighted and has no messages posted to it, then it has *already* exceeded some table threshold, and the File Detail screen (**PF10**) should be used to view the status of the file.
- PF7** Scrolls the list of files upward.
- PF8** Scrolls the list of files downward.
- PF9** Repeats the last command-line command.
- PF10** The cursor must be on a filename for this PF key to work. The user is presented with a detail screen that shows the current state of the file, and shows the current rate of growth for a wide variety of file statistics. If the file requires a password to open, and is not already open in the region, the *SirFile* user is prompted for the password before the Detail display can be shown.
- PF11** Accesses one of the threshold screens to allow the user to enter, update or delete threshold information. If the cursor is not on a file when **PF11** is pressed the user is presented with the System Default thresholds screens, showing defaults used by every file that does not have a specific set of thresholds. If the cursor is on a particular file when **PF11** is pressed, the thresholds for that file are presented.
- PF12** Invokes the *SirFile* database refresh operation.

CHAPTER 5 *File Thresholds Screen*

The user may specify a set of thresholds that apply to only a single file, by placing the cursor on the filename and pressing **PF11**, from either the Main File menu or from the File Tables screen. The File thresholds screen shows the same options as on the System Default thresholds screen, and it also displays some file-specific information.

```

----- * * * Set threshold values for file warnings * * * -----
==> File ==> DATALINK          CREATE/Reorg Date ==> 01/11/91 07:59
                                Last sample taken ==> 02/24/99 23:10

                                # days until
                                threshold
                                exceeded
                                Number of historical
                                records to keep per file.

ARETRIES ==> 1                0                >9999          ==> 99999
BFULLP   ==> 90              23.0           >9999
CFULLP   ==> 80              N/A            N/A            Minimum number of days
CRETRIES ==> 100             0                >9999          between stored samples.
DFULLP   ==> 90              26.6           >9999          ==> 7
EXTNADD  ==> 5000            0                >9999
OVFLADD  ==> 5000            0                >9999
EOVFLADD ==> 5000            0                >9999          # of days advance warning
SPIILLADD ==> 1000           0                >9999          on threshold exceeded.
                                                ==> 7

Collect CFULLP for this file (Y/N)? ==> N
----- ULSPF.063: Threshold values taken from system default settings.
1/Help          3/Quit          9/Repeat          5/Delete          12/Save-Refresh

```

File thresholds screen

See the System Default thresholds screen (in [“Overview and Setup for SirFile”](#) on page 5) for information on the threshold settings themselves. Other fields on this screen are:

File Name

File Name

File Creation/Reorganization Date

Date of the last CREATE run against the file. This date is taken from page trailers in the file. *SirFile* automatically discards its own history records for a file if it detects that a CREATE has been run subsequent to the date of the history records.

Last Sample Kept

The data and time of the last record stored for the file in the *SirFile* database.

Current Table Settings

The column to the right of the threshold settings contains the current file status for ARETRIES, BFULLP, CRETRIES, DFULLP, EXTNADD, OVFLADD, EOVLADD, and SPILLADD.

CFULLP is not calculated in this column because of resource considerations. If however CFULLP was collected on any sample after the first, the latest CFULLP is presented and is used in the calculation of the value in the next column.

days until threshold exceeded

If *SirFile* has captured at least one previous sample for the target file, a third column of values is displayed, which shows the number of days until each of the table parameters exceeds its matching threshold. If a parameter is showing little or no growth, or the table is empty, *SirFile* displays >9999. If the value cannot be calculated, an "N/A" is displayed (such as for CFULLP, if it has not been collected).

Collect CFULLP for this file?

Selecting "Y" at this prompt allows *SirFile* to run the TABLEC command for the file. This prompt is not presented if the user has specified "N" for the following prompt on the System Default thresholds screen:

Collect CFULLP only on specified files

The following commands are allowed:

- OPEN xxxxxxxx** Opens a file. As data can only be collected for files that *SirFile* already knows about, or files that are currently open by any user, a *SirFile* user may want to manually open a file before running a refresh to guarantee the file is included in the sample.
- CLOSE xxxxxxxx** Closes a file. Any file that the *SirFile* user has open can be closed, except SIRFILE, SIRLOCAL, and CCASYS (the required application subsystem files for *SirFile*).

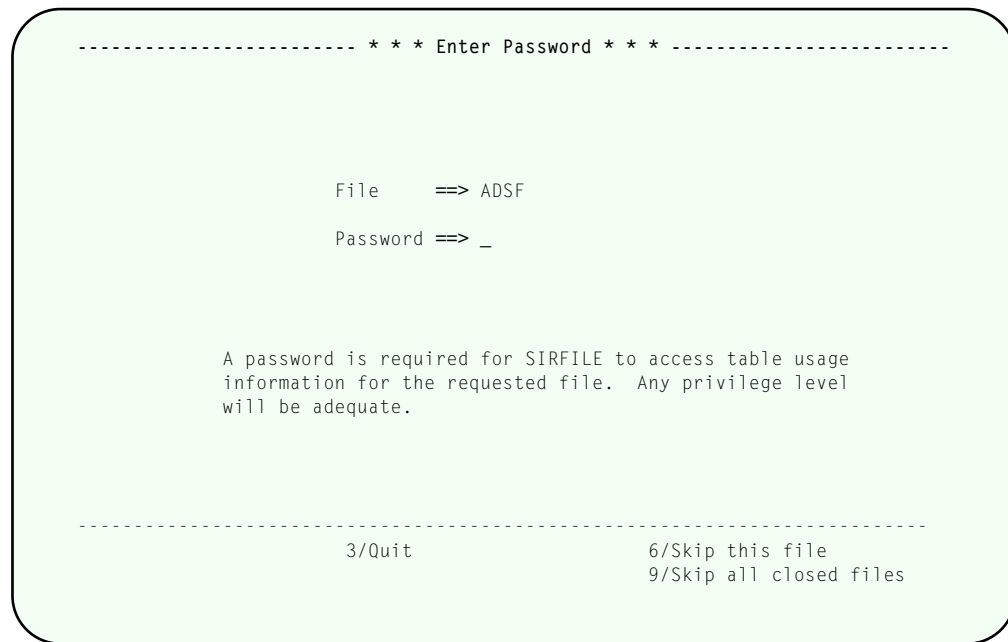
PF keys:

- PF1** Accesses full-screen help.
- PF3** Return to command level or previous screen.
- PF5** Delete this file-specific threshold. Deleting a file-specific threshold allows the file to be evaluated against the system default thresholds on all subsequent refresh operations.
- PF5** Delete the file-specific threshold being viewed. This operation returns the user to the previous screen once the threshold is deleted.

- PF9** Repeats the last command-line command.
- PF12** Saves the file specific thresholds and, for this file only, invokes the *SirFile* database refresh operation.

File Open Screen

When *SirFile* needs to open a file, it first tries to do it without a password. If that fails the user is prompted for a password with the following screen:



File open screen

SirFile requires only read privileges in files being monitored.

If the user is being prompted for passwords in the middle of a refresh operation, the PF key choices are as shown in the figure, and have the following actions:

- PF3** Cancel the refresh and return to the previous screen.
- PF6** Skip the current file in the refresh.
- PF9** Skip the this and the rest of the files in the refresh that require manual entry of passwords.

Otherwise, for example if the user is prompted for a password due to an OPEN command, the only PF key choice is:

- PF3** Cancel the open and return to the previous screen.

CHAPTER 7 *File Detail Screen*

The File Detail screen may be accessed from the Main File menu or the File Tables screen, by placing the cursor on a file name and pressing **PF10**.

```

-----ULSPF500 / 4.1.1E / CMS ----- INTERVAL: * SEC----
==> _
FILENAME  ENQSHR  ENQEXC  NREQ    NUPD  BUFPAGE  CFRQUEU  FISTAT  FRCVOPT
ALANPROC      0      1        0        0      1        0        0        4
  ASIZE   BSIZE   CSIZE   DSIZE  FREESIZ  NRECMAS  NRECEXT  FILEORG  FOPT
    3     10     10    1350    219     13        0        0        0
  RETRYA  BFULLP  RETRYC  DFULLP
    0     0.00    0     44.74

      DKRD   DKWR  DIRRCD  DUPDTS    REQ  CFRCONF
SEC   0.00   0.00   0.00   0.00   0.00  0.00
TOT    2     2     0     0     0     0

      RECADD  RECDEL  IXADD  IXDEL  BADD  BCHG  BDEL
SEC   0.00   0.00   0.00   0.00   0.00  0.00  0.00
TOT    0     0     0     0     0     0     0

      BXCHNG  BXDELE  BXNEXT  BXFIND  BXINSE  BXSPLI  BXRFDN  BXFREE
SEC   0.00   0.00   0.00   0.00   0.00  0.00  0.00  0.00
TOT    0     0     0     0     0     0     0     0
-----
1/Help                3/Quit                6/Msgs

```

File Detail screen

“File statistics definitions” on page 37 contains statistic descriptions for this display.

The File Detail screen may be placed into auto-refresh mode via the TIME command:

TIME xx yy This command takes a snapshot of the file every yy seconds until xx snapshots have been taken. The rate and total information displayed is updated with each snapshot. The snapshots are not stored, but they allow the user to view precise per-second changes in the state of the file. The automatic refresh may be interrupted via **PA1**.

This screen is especially useful when a warning is posted against a file, as it allows the user to view all relevant file table information.

PF keys:

- PF1** Accesses full-screen help.
- PF3** Return to command level or previous screen.
- PF6** Accesses the messages screen if any messages are posted against this file.

PF4 from the Main File menu presents a scrollable list of the table-full status of files currently open by any user or subsystem.

```

-----FILE 1/22 ----- ULSPF500/4.1.1E/CMS -----
====> _
  FILENAME NRECMAS NRECEXT BFULLP DFULLP  RETRYA  RETRYC
  DATALINK 2376      0 23.000 26.666      0      0
  METADATA 1966      6 33.500 22.000      0      0
  SIRLIBD  1683     220  4.000  7.666      0      0
  CCASYS   1432     96 22.941 24.375      0      0
  M204CTL  567      6 35.555 10.000      0      0
  SIRFILED 423      0  1.333  3.428      0      0
  SIRLOCAL 230      2  5.000  5.000      0      0
  M204TEMP 29      0  0.000 20.000      0      0
  SIRULSPF 16      0  0.000 70.843      0      0
  > MOVE875 15      0 75.000 86.363      0      0
  M204PROC 3      0  0.000 60.031      0      0
  SIRLIBP  0      0  0.000  1.400      0      0
  SIRFIXES 0      0  0.000 54.300      0      0
  SIRLIB   0      0  0.000 30.344      0      0
  SIRXREF  0      0  0.000 54.187      0      0
  SIRMON   0      0  0.000 49.600      0      0
  SIRFILE  0      0  0.000 48.400      0      0
-----
  1/Help      3/Quit      5/SortFile  6/Msgs
  7/Up       8/Down     9/Repeat 10/Detail 11/Thresh 12/Refresh

```

File Tables screen

Like the File Detail screen, the File Tables screen displays information about open files in the region using statistics taken directly from the files' tables. If a file has a warning or prediction stored against it, it is highlighted on this display.

The following commands work on this screen:

USER xxxx

restricts the view to files held open by the specified user NUMBER (userid will not work).

SUBSYS xxxxx

restricts the view to files held open by the specified application subsystem. An asterisk (*) can be used as a wildcard character, so "SUBSYS SIR*" will display files that have an enqueue on them by any subsystem beginning with the letters "SIR".

FILE xxxxxxxx

restricts the display to the named file. An asterisk (*) can be used as a wildcard character, so "FILE *ADAC*" will display any files whose names contain the string "ADAC".

n	Where n is any number. Scrolls the display forward the specified number of files.
-n	Where n is any number. Scrolls the display backward the specified number of files.
F xxxxxxxx or /xxxxxxx	Searches forward in the file list for the requested file name “xxxxxxx”, and makes it the first line on the display.
-F xxxxxxxx or -/xxxxxxx	Searches backward in the file list for the requested file name “xxxxxxx”, and makes it the first line on the display.
Top	Moves the display to the top of the file list.
Bottom	Moves the display to the bottom of the file list.
M	When using PF7 or PF8 , scroll up or down to the top or bottom of the list of files.
SORT stat x	Where stat is the screen name of any of the displayed statistics, and x is A or D, for ascending or descending. Multiple sort criteria may be specified, as in the following example: <pre>SORT RETRYC A, NRECMAS D</pre>
OPEN xxxxxxxx	Opens a file. This is useful if you want to guarantee that a file is included in a refresh operation you are about to initiate.
CLOSE xxxxxxxx	Closes a file. Any file that the <i>SirFile</i> user has open can be closed, except SIRFILE, SIRLOCAL and CCASYS (the required application subsystem files for <i>SirFile</i>).

PF keys for this screen are:

- PF1** Accesses full screen help.
- PF3** Return to previous screen.
- PF6** Accesses table-full prediction messages for files that have them posted.
- PF7** Scrolls the display up.
- PF8** Scrolls the display down.
- PF9** Repeats the previous command-line command.

- PF10** Accesses the File Detail screen, if the cursor is positioned over a file statistic line when the PF key is pressed.
- PF11** Accesses the threshold screen for a selected file.
- PF12** Performs the Refresh process, either for all files, or for a single file if the cursor is placed over a file statistic line when the PF key is pressed.

CHAPTER 9 *Messages Screen*

Placing the cursor on a file name in the Main File menu or the File Tables screen, and pressing **PF6**, allows the user to view messages posted to the file.

```
----- * * * Messages and warnings posted to file * * * -----  
==> _                               File:  ALANPROC  
  
03/01/99: Table B is likely to exceed the threshold on MAR 17, 1999  
03/01/99: EXTNADD threshold is likely to be exceeded MAR 25, 1999  
  
-----  
1/Help          3/Quit          6/Main  
7/Up            8/Down         9/Repeat 10/Detail
```

SIRFILE.TRC # 2

This screen can only be accessed for files that have messages posted. Messages are only posted when a file is predicted to fill. If the file has already exceeded a threshold, the user should view the file tables and parameters via the File Detail screen or the File thresholds screen.

PF keys:

- PF1** Accesses full-screen help.
- PF3** Return to previous screen.
- PF5** Delete the file-specific threshold being viewed. This operation returns the user to the previous screen once the threshold is deleted.

CHAPTER 10 *Background Monitor Specification Screen*

PF2 from the Main File menu presents the Background Monitor specification screen.

```

----- * * * Background File Monitor * * * -----
==> _

Default monitoring increment (minutes)           ==> 9

----- Warnings -----
Send warnings to the journal? (Y/N)             ==> N
Warn: 0 = Never, 1 = First time only, 2 = Every time ==> 1
Other users to receive warning messages (separate by blanks):
==> OPR

----- Predictions -----
Send predicts to the journal? (Y/N)             ==> N
Predict: 0 = Never, 1 = First time only, 2 = Every time ==> 1
Other users to receive predict messages (separate by blanks):
==> OPR

-----
1/Help           3/Quit           5/Init-Backgrnd           12/Save
9/Repeat

```

Background Monitor specification screen

Use this screen to control *SirFile* background monitoring, and to specify the actions taken when a file-full-pending condition is detected by any refresh process. Warnings and predictions are treated separately. Entry fields are:

- Monitoring Increment** Number of minutes the background *SirFile* task should wait between checking files.
- Warnings to Journal** Y or N. Specifying Y causes *SirFile* to send a user audit line (record type US) to the journal each time a warning condition is noticed for any file by the background task.
- Warning frequency** 0, 1 or 2. Specifying “0” causes *SirFile* to not send warnings on threshold-exceeded conditions. “1” causes *SirFile* to warn users specified in the warn list the first time a condition is noticed for a file. “2” causes *SirFile* to send a warning to each user id on the warn list each time the background monitoring task notices the condition.
- Warn List** Specify a list of user IDs and/or thread numbers to receive a WARN message by the background task. User IDs and numbers must be separated by blanks or commas.

Predictions to Journal

Y or N. Specifying Y causes *SirFile* to send a user audit line (record type US) to the journal each time a prediction condition is noticed by the background task. (Prediction conditions mean the file is predicted to fill within the number of days of advance warning as shown on the File threshold screen.)

Predict frequency

0, 1 or 2. "0" causes *SirFile* to disregard predictions. "1" causes *SirFile* to warn users on the predict list the first time a prediction condition is noticed. "2" causes *SirFile* to send a warning to each user ID on the predict list each time the background task notices the condition.

Predict List

Specify a list of user IDs and/or thread numbers to receive prediction messages by the background task. User IDs and numbers must be separated by blanks or commas.

The following commands work on this screen:

OPEN xxxxxxxx Opens a file. As data can only be collected for files that *SirFile* already knows about, or files that are currently open by any user, a *SirFile* user may wish to manually open a file before running a refresh to guarantee the file is included in the sample.

CLOSE xxxxxxxx Closes a file. Any file that the *SirFile* user has open can be closed, except SIRFILE, SIRLOCAL and CCASYS (the required application subsystem files for *SirFile*).

PF keys:

PF1 Accesses full-screen help.

PF3 Return to previous screen.

PF5 Save the new background task specifications, and initialize the background task. If the background task is already running, the user is notified. To make the background task accept the new settings, the sdaemon background user should be bumped and restarted.

PF9 Repeats the previous command-line command.

PF12 Save the new background task specifications.

CHAPTER 11 *Invocation Techniques for Background Monitoring*

SirFile's background monitor runs as a special user in the SIRFILE APSY; this performs the **periodic** refresh operation. Messages to users and the audit trail are only sent during a periodic refresh (that is, by the background task).

Any user ID can be the background monitor, and any *SirFile* user can initiate the background task. The most useful way to run the background monitor is in one of the sdaemon IODEVs set up in the online. If you are going to run the background file monitor in an sdaemon, you should allocate an extra one explicitly for this task. Failing to allocate an extra sdaemon can slow down other jobs initiated from Sirius products which require sdaemon servicing.

When the background monitoring task is sent to an sdaemon, the task appears as the sdaemon thread number and the user ID of the submitting user. The sdaemon sits in a WAIT state until the specified number of minutes has passed, then it wakes up and runs a *SirFile* database refresh, sends warnings as appropriate, then returns to a “sleep” state. The background task may be BUMPed the same as any user. To change the time parameter for the background task, it should be bumped and resubmitted.

PF5 from the Background Monitor specification screen (see “[Background Monitor Specification Screen](#)” on page 31) will run the monitoring task in an sdaemon; it can also be initiated via one of the following commands:

SIRFILE BACKGROUND interval

This command, typed at command level or inserted in the USER0 stream, initiates the background task under an sdaemon. *Interval* specifies the number of minutes the background task should sleep between refresh operations; it is a required value and must be between 1 and 1440 (24 hours).

If no sdaemon is available or if the background task is already running, the submitting user is warned, and the request is ignored.

SIRFILE SDAEMON interval

This command, typed at command level, causes the session to become the background task. *Interval* specifies the number of minutes the background task should sleep between refresh operations; it is a required value and must be between 1 and 1440 (24 hours).

The command may be placed in a BATCH2 or IODEV3 input stream, or it may be typed at command level from any terminal. Typing this command on

a terminal will cause that session to become the background *SirFile* task: that is, it will enter a “sleep” state that is not interruptable unless it is bumped.

You can also use the **SIRFILE BATCH** command to invoke a refresh once; see [“Running SirFile under BATCH204” on page 35](#), which describes the most useful way to do this.

 CHAPTER 12 *Running SirFile under BATCH204*

Running *SirFile*'s database refresh in batch has two advantages over running it in the online:

1. It allows the database refresh to be scheduled on a regular basis by a job scheduling system.
2. It allows the most expensive processing in *SirFile* to be run in off hours.

In addition, if you have specified “N” for the following prompt on the System Default thresholds screen, the CFULLP statistic is only calculated if the refresh is run via the SIRFIELD BATCH command:

Collect CFULLP only on specified files

CFULLP is a more accurate measure of the full status of Table C, but it requires that a TABLEC command be executed, which can be a very resource-intensive operation.

Following is a sample job fragment and CCAIN stream showing the *SirFile* refresh being run in BATCH204.

```
//SIRFILE JOB (Ø) 'Batch Refresh',CLASS=X
//REFRESH EXEC PGM=BATCH204,SYSOPT=187
//...
//CCASYS DD DSN=M204.CCASYS,DISP=SHR
//CCAGRP DD DSN=M204.CCAGRP,DISP=SHR
//SIRFILE DD DSN=M204.SIRFILE,DISP=SHR
//SIRLOCAL DD DSN=M204.SIRLOCAL,DISP=SHR
//PROCFIL DD DSN=M204.PROCFIL,DISP=SHR
//PROCFIL2 DD DSN=M204.PROCFIL2,DISP=SHR
//...
//CCAIN *
NFILES=99,NDIRS=99,NDCBS=99,NUSERS=2,NSERVS=2,...
IODEV=15,POLLNO=1,NOTERM=1
LOGON filemanagr
password
OPEN FILE PROCFILE1
password
OPEN FILE PROCFIL2
password
...
SIRFILE BATCH
EOJ
/*
//
```

The job is simply a BATCH204 job that contains CCASYS and CCAGRP, and it has a DD card for any file *SirFile* is going to monitor. The IODEV=15 statement is required; change it to IODEV=*nn* if you reset the SDAEMDEV parameter to *nn*. SYSOPT for the job must allow for use of Groups and Application Subsystems. If any file is not yet in the *SirFile* database or requires a password to open, it should be opened explicitly in the CCAIN stream.

The command SIRFILE BATCH tells the SIRFILE APSY to run through the Refresh process, including TABLEC calculations for CFULLP (Table C percentage full) if CFULLP allocation is not determined file-by-file.

Note: A file must be open with File Manager privileges (PRIVDEF must contain the x'8000' bit) in order for the TABLEC command to operate. When *SirFile* adds files to its APSY definition, it adds them with privileges x'0221' to SCLASS "USER" and x'8221' to SCLASS "ADMIN". If a site wants to run the batch job with TABLEC commands for each file, the logged-on user in the batch job must be in the ADMIN SCLASS.

Users may customize their run by manually changing privileges for specific files in the USER SCLASS to x'8221'. Then, running SIRFILE BATCH with a user ID from the USER SCLASS will only calculate TABLEC for the files with high privileges. (Note that the output will show a failure of the TABLEC command for all files with too low of privileges).

 APPENDIX A *File statistics definitions*

File statistics provide information about each file open in the Online system. Each file has a specific value for each statistic based on the file's current activity and state within the Online region. Many of these statistics are viewable on the File Tables screen and the File Detail screen in *SirFile*. Others may be viewable only at command level or through *SirMon*.

ASIZE	Size of table A in number of pages.
ASTRPPG	Table A strings per page.
ATRPG	Number of attribute pages in Table A.
BACKOUTS	Number of backed out update units since the file was opened. The stat begins incrementing when the file is first opened, and is reset to 0 when the last user closes it. If the file is reopened the incrementing begins again from 0 (rate or total).
BADD	Fields added to Table B (rate or total).
BCHG	Fields changed in place in Table B (rate or total).
BDEL	Fields deleted from table B (rate or total).
BFULLP	Estimate of the percentage of table B that is full. This statistic is equal to the table B highest active page (BHIGHPG) divided by the total number of table B pages (BSIZE).
BHIGHPG	The current table B high water mark. Same as the Model204 BHIGHPG parameter.
BLOWPG	The lowest active table B page. Same as the Model204 BLOWPG parameter.
BQLEN	The number of pages in the table B reuse queue. Same as the Model204 BQLEN parameter.
BRECPPG	Maximum number of records per Table B page.
BRESERVE	The amount of space in bytes on a Table B page on which new records cannot be started. The space is reserved for the expansion of existing records on the Table B page.

BREUSED	The number of records added to the file that reused the record number of a previously deleted record. Same as the Model204 BREUSED parameter.
BSIZE	Size of table B in number of pages.
BUFF.	Total number of pages in the disk buffer pool currently occupied by pages from all tables in the database file. Equal to BUFFF + BUFFA + BUFFB + BUFFC + BUFFD + BUFFE + BUFFX.
BUFF.G	Total number of above-the-bar pages in the disk buffer pool occupied by pages from all tables in the database file. Equal to BUFFFFG + BUFFAG + BUFFBG + BUFFCG + BUFFDG + BUFFEG + BUFFXG. Above-the-bar buffers reside at virtual addresses above two gigabytes and are only used when the NUMBUFG parameter is set to a non-zero value under Model 204 V6R2 or later.
BUFF.L	Total number of below-the-bar pages in the disk buffer pool occupied by pages from all tables in the database file. Equal to BUFFF + BUFFA + BUFFB + BUFFC + BUFFD + BUFFE + BUFFX. Below-the-bar buffers reside at virtual addresses below two gigabytes. BUFF.L has the same value as BUFF. unless the NUMBUFG parameter is set to a non-zero value under Model 204 V6R2 or later.
BUFFA	Total number of pages in the disk buffer pool currently occupied by table A pages from the database file.
BUFFAG	Total number of above-the-bar pages in the disk buffer pool currently occupied by table A pages from the database file.
BUFFAL	Total number of below-the-bar pages in the disk buffer pool currently occupied by table A pages from the database file.
BUFFB	Total number of pages in the disk buffer pool currently occupied by table B pages from the database file.
BUFFBG	Total number of above-the-bar pages in the disk buffer pool currently occupied by table B pages from the database file.
BUFFBL	Total number of below-the-bar pages in the disk buffer pool currently occupied by table B pages from the database file.
BUFFC	Total number of pages in the disk buffer pool currently occupied by table C pages from the database file.
BUFFCG	Total number of above-the-bar pages in the disk buffer pool currently occupied by table C pages from the database file.

BUFFCL	Total number of below-the-bar pages in the disk buffer pool currently occupied by table C pages from the database file.
BUFFD	Total number of pages in the disk buffer pool currently occupied by table D pages from the database file. Note that CCATEMP pages are listed as being in table D, though they're not really table D. Note also that in the system buffer usage statistics, CCATEMP pages are not included in the table D totals and are instead included in a separate value called BUFFT.
BUFFDG	Total number of above-the-bar pages in the disk buffer pool currently occupied by table D pages from the database file. Note that CCATEMP pages are listed as being in table D, though they're not really table D. Note also that in the system buffer usage statistics, CCATEMP pages are not included in the table D totals and are instead included in a separate value called BUFFTG.
BUFFDL	Total number of below-the-bar pages in the disk buffer pool currently occupied by table D pages from the database file. Note that CCATEMP pages are listed as being in table D, though they're not really table D. Note also that in the system buffer usage statistics, CCATEMP pages are not included in the table D totals and are instead included in a separate value called BUFFTL.
BUFFE	Total number of pages in the disk buffer pool currently occupied by Table E pages from the database file.
BUFFEG	Total number of above-the-bar pages in the disk buffer pool currently occupied by table E pages from the database file.
BUFFEL	Total number of below-the-bar pages in the disk buffer pool currently occupied by table E pages from the database file.
BUFFF	Total number of pages in the disk buffer pool currently occupied by FCT pages from the database file.
BUFFFG	Total number of above-the-bar pages in the disk buffer pool currently occupied by FCT pages from the database file.
BUFFFL	Total number of below-the-bar pages in the disk buffer pool currently occupied by FCT pages from the database file.
BUFFX	Total number of pages in the disk buffer pool currently occupied by Table X pages from the database file.
BUFFXG	Total number of above-the-bar pages in the disk buffer pool currently occupied by table X pages from the database file.

BUFFXL	Total number of below-the-bar pages in the disk buffer pool currently occupied by table X pages from the database file.
BUFM.	Total number of pages in the disk buffer pool currently occupied by modified pages from all tables in the database file. Equal to BUFMF + BUFMA + BUFMB + BUFMC + BUFMD + BUFME + BUFMX.
BUFMA	Total number of pages in the disk buffer pool currently occupied by modified table A pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
BUFMAG	Total number of above-the-bar pages in the disk buffer pool currently occupied by modified table A pages from the database file. These must pages be written to disk before they can be flushed from the buffer pool.
BUFMAL	Total number of below-the-bar pages in the disk buffer pool currently occupied by modified table A pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
BUFMB	Total number of pages in the disk buffer pool currently occupied by modified table A pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
BUFMBG	Total number of above-the-bar pages in the disk buffer pool currently occupied by modified table B pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
BUFMBL	Total number of below-the-bar pages in the disk buffer pool currently occupied by modified table B pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
BUFMC	Total number of pages in the disk buffer pool currently occupied by modified table A pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool. Total number of pages in the disk buffer pool currently occupied by modified table A pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
BUFMCG	Total number of above-the-bar pages in the disk buffer pool currently occupied by modified table C pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
BUFMCL	Total number of below-the-bar pages in the disk buffer pool currently occupied by modified table C pages from the database file. These

pages must be written to disk before they can be flushed from the buffer pool.

- BUFMD** Total number of pages in the disk buffer pool currently occupied by modified table A pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool. Note that CCATEMP pages are listed as being in table D, though they're not really table D. Note also that in the system buffer usage statistics, CCATEMP pages are not included in the table D totals and are instead included in a separate value called BUFMT.
- BUFMDG** Total number of above-the-bar pages in the disk buffer pool currently occupied by modified table D pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool. Note that CCATEMP pages are listed as being in table D, though they're not really table D. Note also that in the system buffer usage statistics, CCATEMP pages are not included in the table D totals and are instead included in a separate value called BUFMTG.
- BUFMDL** Total number of below-the-bar pages in the disk buffer pool currently occupied by modified table D pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool. Note that CCATEMP pages are listed as being in table D, though they're not really table D. Note also that in the system buffer usage statistics, CCATEMP pages are not included in the table D totals and are instead included in a separate value called BUFMTL.
- BUFME** Total number of pages in the disk buffer pool currently occupied by modified table E pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
- BUFMEG** Total number of above-the-bar pages in the disk buffer pool currently occupied by modified table E pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
- BUFMEL** Total number of below-the-bar pages in the disk buffer pool currently occupied by modified table E pages from the database file. These pages be must written to disk before they can be flushed from the buffer pool.
- BUFMF** Total number of pages in the disk buffer pool currently occupied by FCT pages from the database file. These pages must be written to to disk before they can be flushed from the buffer pool.
- BUFMFG** Total number of above-the-bar pages in the disk buffer pool currently occupied by modified FCT pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.

BUFMFL	Total number of below-the-bar pages in the disk buffer pool currently occupied by modified FCT pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
BUFMX	Total number of pages in the disk buffer pool currently occupied by modified table X pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
BUFMXG	Total number of above-the-bar pages in the disk buffer pool currently occupied by modified table X pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
BUFMXL	Total number of below-the-bar pages in the disk buffer pool currently occupied by modified table X pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
BUFPAGA	Total number of pages in the disk buffer pool currently occupied by table A pages from the database file.
BUFPAGB	Total number of pages in the disk buffer pool currently occupied by table B pages from the database file.
BUFPAGC	Total number of pages in the disk buffer pool currently occupied by table C pages from the database file.
BUFPAGD	Total number of pages in the disk buffer pool currently occupied by table D pages from the database file.
BUFPAGE	Total number of pages in the disk buffer pool currently occupied by pages from all tables in the database file. Equal to BUFPAGA + BUFPAGB + BUFPAGC + BUFPAGD + BUFPAGF + BUFPAGL + BUFPAGX + the number of table X pages which can be obtained via the BUFFX statistic.
BUFPAGF	Total number of pages in the disk buffer pool currently occupied by FCT pages from the database file.
BUFPAGL	Total number of pages in the disk buffer pool currently occupied by Table E pages from the database file. The L stands for LOBs, or large objects, which is what is stored in table E.
BUFPAGX	Total number of pages in the disk buffer pool currently occupied by pages from CCATEMP. Note: these are *not* pages from table X. The preferred set of buffer usage statistics that includes the table X pages is the sets of stats that begin with the letters "BUFF".

BXCHNG	New segments of records starting or local lists deleted in the ordered index (rate or total).
BXDELE	Removals from tree structure in the ordered index. (rate or total).
BXFIND	Searches to locate FIELD NAME = VALUE pairs in the ordered index (rate or total).
BXFREE	Nodes emptied in the ordered index (rate or total).
BXINSE	New records inserted with an unused FIELD NAME = VALUE pair in the ordered index (rate or total).
BXNEXT	FIELD NAME = VALUE pairs touched in the ordered index during range retrievals (rate or total).
BXRFND	Calls for range retrieval from the ordered index (rate or total).
BXSPLI	Node splits in the ordered index (rate or total).
CFRCDEX	Conflicts on the "DIRECT" and "EXISTS" critical file resources (rate or total). Equal to CFRCDIR+CFRCEXS.
CFRCDIR	Conflicts on the "DIRECT" critical file resource (rate or total). This resource protects the integrity of Table B.
CFRCEXS	Conflicts on the "EXISTS" critical file resource (rate or total). This resource protects the integrity of the existence bit map.
CFRCIND	Conflicts on the "INDEX" critical file resource (rate or total). This resource protects the integrity of the database indexes.
CFRCONF	Conflicts on all critical file resources (rate or total).
CFRCREC	Conflicts on the "RECENQ" critical file resource (rate or total). This resource protects the integrity of the record enqueueing tables.
CFRQDEX	Number of users waiting for either the "DIRECT" or "EXISTS" critical file resource. Equal to CFRQDIR+CFRQEXS.
CFRQDIR	Number of users waiting for the "DIRECT" critical file resource. This resource protects the integrity of table B.
CFRQEXS	Number of users waiting for the "EXISTS" critical file resource. This resource protects the integrity of the existence bit map.
CFRQIND	Number of users waiting for the "INDEX" critical file resource. This resource protects the integrity of the database indexes.

CFRQREC	Number of users waiting for the "RECENQ" critical file resource. This resource protects the integrity of the record enqueueing tables.
CFRQUEU	Number of users waiting for any critical file resource.
COMMITTS	Number of committed update units since the file was opened. The stat begins incrementing when the file is first opened, and is reset to 0 when the last user closes it. If the file is reopened the incrementing begins again from 0. (rate or total.)
CSIZE	Size of table C in number of pages.
DFULLP	Percentage of table D that is full. This statistic is equal to the number of table D pages currently in use (DPGSUSED) divided by the total number of table D pages (DSIZE).
DHIGHPG	The current table D high water mark. Same as the Model204 DHIGHPG parameter.
DIRRCD	Number of records scanned in direct searches (rate or total).
DKIO	Physical page reads/writes to/from the Model204 database file (rate or total). This value is equal to DKRD + DKWR.
DKRD	Physical pages reads from the Model204 database file (rate or total).
DKUPTIME	Number of milliseconds spent writing a file's pages to disk, and marking it "physically consistent". It includes time writing pages even if the disk update was interrupted. This stat is presented by SirMon in seconds.
DKWR	Physical page writes to Model204 database file (rate or total).
DPGSUSE	The current number of table D pages in use. Same as the Model204 DPGSUSED parameter.
DSIZE	Size of table D in number of pages.
DUPDTS	Records written to the deferred update data set for the Model204 database file (rate or total).
EFULLP	Percentage of table E that is full. This statistic is equal to the number of table E pages currently in use (EPGSUSED) divided by the total number of table E pages (ESIZE).
EHIGHPG	The current table E high water mark. Same as the Model204 EHIGHTPG parameter.

ENQEXC	Total number of users and subsystems that have the database file opened in update mode.
ENQSHR	Total number of users and subsystems that have the database file opened in read/only mode.
EOVFADD	Number of records added to the extra overflow areas. Same as the Model204 EOVLADD parameter (rate or total).
EOVFDEL	Number of deleted from to the extra overflow areas. Same as the Model204 EOVLDEL parameter (rate or total).
EPGSUSE	The current number of table E pages in use. Same as the Model204 EPGUSED parameter.
ESIZE	Size of table E in number of pages.
EXTNADD	Number of extension records added to table B. Same as the Model204 EXTNADD parameter (rate or total).
EXTNDEL	Number of extension records deleted from table B. Same as the Model204 EXTNDEL parameter (rate or total).
FCTB	Total number of pages in the disk buffer pool currently occupied by FCT pages from the database file.
FCTM	Total number of pages in the disk buffer pool currently occupied by modified FCT pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
FICREATE	Release of Model 204 in which the file was created. Values are: <ul style="list-style-type: none"> 0 - Release 7.1 or earlier release of Model 204 1 - Release 8.0 of Model 204 2 - Release 8.1 of Model 204 3 - Release 9.0 (or later) of Model 204 4 - Version 2 Release 2 (or later) of Model 204 and also <ul style="list-style-type: none"> has a non-zero setting of FILEMODL parameter. 5 - Version 3 Release 2 (or later) of Model 204 and also <ul style="list-style-type: none"> has a non-null setting of LANGFILE parameter. 6 - Version 6 Release 1 (or later) of Model 204. 7 - Version 6 Release 2 (or later) of Model 204. 8 - Version 6 Release 3 (or later) of Model 204.

FIFLAGS Summed hex value representing file status detail. Valid settings are sums of the following values:

X'80' - Group index entry required (sorted files).
X'40' - Table B appends page full.
X'20' - Currently active Table D inversion page full.
X'10' - Field level constraints (file has at least one field with a UNIQUE attribute).
X'08' - New Table C logic for initial property entry.
X'04' - Ordered Index list page full.
X'01' - File is post release 3.16.

FILEORG Summed hex value representing the organization of Table B. Valid settings are sums of the following values:

X'40' - File skewing is enabled.
X'20' - Unordered file.
X'08' - Hash key file.
X'04' - Reuse Record Number file (RRN).
X'02' - Sort or hash key required in every record.
X'01' - Sorted file.
X'00' - Entry order file.

FISTAT The current value of the database file's status byte. Same as the Model204 FISTAT parameter. Values can be summed:

X'40' - File might be logically inconsistent.
X'20' - File is in deferred update mode.
X'10' - File has been recovered.
X'08' - File is full.
X'02' - File is physically inconsistent.
X'01' - File is not initialized.

Model204 FISTAT parameter.

FOPT Summed hex value representing file options, as follows:

X'80' - Prohibit statement numbers in procedures.
X'40' - Prohibit statement labels in procedures.
X'08' - Append-first mode in RDFS file.
X'02' - Disable lock pending updates.
X'01' - Disallow new field name definitions.
X'00' - Allow labels and statement numbers in procedures.

FRCVOPT	Summed hex value indicating file recovery options, as follows: X'80' - Updates not allowed without roll forward active. X'40' - Updates not allowed unless checkpointing active. X'20' - File does not participate in checkpointing. X'10' - Discontinuities not allowed. X'08' - Transaction backout disabled. X'04' - File does not take part in roll forward logging. X'02' - File does not participate in roll forward. X'01' - Transaction boundaries are ignored when applying roll forward updates.
FREESIZ	Number of unassigned pages in the database file. Same as the Model204 FREESIZE parameter.
FVFPG	Number of Table A pages used for storing the values from "few valued" coded fields.
HIGHSRT	Record number containing the highest sort key. Same as the Model204 HIGHSORT parameter.
IXADD	Index entries added to tables C and D, including attempts to add duplicates (rate or total).
IXDEL	Index entries deleted from tables C and D (rate or total).
MODPAGA	Total number of pages in the disk buffer pool currently occupied by modified table A pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
MODPAGB	Total number of pages in the disk buffer pool currently occupied by modified table B pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
MODPAGC	Total number of pages in the disk buffer pool currently occupied by modified table C pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
MODPAGD	Total number of pages in the disk buffer pool currently occupied by modified table D pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
MODPAGE	Total number of pages in the disk buffer pool currently occupied by modified pages from all tables in the database file. These pages must be written to disk before they can be flushed from the buffer pool. Equal to MODPAGA + MODPAGB + MODPAGC + MODPAGD + MODPAGF + MODPAGL.

MODPAGF	Total number of pages in the disk buffer pool currently occupied by modified FCT pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool.
MODPAGL	Total number of pages in the disk buffer pool currently occupied by modified table E pages from the database file. These pages must be written to disk before they can be flushed from the buffer pool. The L stands for LOBs, or large objects, which is what is stored in table E.
MODPAGX	Total number of pages in the disk buffer pool currently occupied by modified pages from CCATEMP. Note: these are *not* modified pages from table X. The preferred set of buffer usage statistics that includes the table X pages is the sets of stats that begin with the letters "BUFM".
MSTRADD	Number of records added to table B. Same as the Model204 MSTRADD parameter (rate or total).
MSTRDEL	Number of records physically deleted from table B. Same as the Model204 MSTRDEL parameter (rate or total).
MVFPFG	Number of Table A pages used for storing the values from "many valued" coded fields.
NDPU	The current number of table D pages in use. Same as the Model204 DPGSUSED parameter.
NPTA	Size of table A in pages. Same as ASIZE.
NPTB	Size of table B in pages. Same as BSIZE.
NPTC	Size of table C in pages. Same as CSIZE.
NPTD	Size of table D in pages. Same as DSIZE.
NPTE	Size of table E in pages. Same as ESIZE.
NPTF	Size of FCT in pages.
NRECEXT	Number of extension records currently in table B.
NRECMAS	Number of master records currently in table B.
NRECOVF	Number of records currently in overflow areas in table B.
NRECSPL	Number of records currently in spill areas in table B.
NREQ	Current number of active transactions which reference the file.
NUPD	Current number of active transactions which update the file.

OIDEPTH	Number of Ordered Index B-tree levels (depth of the B-tree).
OILEAFP	Percentage of ordered index leaf pages that contain data. $OILEAFP = 100 * (OINBYTES / OILEAVS) / PAGESZ$. In general, the higher this value, the better. After a reorg, this value will generally be close to $100 - LRESERV$. A value close to 50 or lower is a sign of relatively low ordered index density and suggests a reorg.
OILEAVS	The number of Ordered Index B-tree leaf-level nodes. The number of Table D pages used for the B-tree leaves.
OILPACT	Page number of the Order Index B-tree page which is active.
OINBYTES	Number of Ordered Index B-tree entry bytes. .
OINENTR	The number of distinct field name = value pairs stored in the Ordered Index B-tree.
OINODEP	Percentage of ordered index node pages that contain data. $OINODEP = 100 * (OINBYTES / OINODES) / PAGESZ$. In general, the higher this value, the better. After a reorg, this value will generally be close to $100 - LRESERV$. A value close to 50 or lower is a sign of relatively low ordered index density and suggests a reorg. This value will usually be close to OILEAFP except in the case where there are few leaf nodes per higher level nodes. This situation would suggest a reorg.
OINODES	The number of Ordered Index B-tree nodes. The total number of Table D pages used for the Ordered Index B-tree.
OPENCTL	PUBLIC, PRIVATE or SEMIPUBLIC status of file.
OVFLADD	Number of records added to overflow areas in table B. Same as the Model204 OVFLADD parameter (rate or total).
OVFLDEL	Number of records deleted from overflow areas in table B. Same as the Model204 OVFLDEL parameter (rate or total).
PDCHUNK	Number of "chunks" allocated for the procedure dictionary. The procedure dictionary is allocated in chunks of PDSIZE pages. When an attempt to store a name in the dictionary fails because all chunks are full, a new chunk is allocated.
PDPAGES	Total number of pages allocated for the procedure dictionary. Equal to $PDCHUNK * PDSIZE$.
PDSIZE	Procedure Dictionary size.
PDSTRPG	Procedure Dictionary Strings per page.

PNDGTIME	Number of milliseconds since the last update unit completed that a file's pages have been waiting to be written. The stat is only accumulated when DKUPDTWT is non-zero.
PRIVDEF	Default privileges for the file (summed hex value).
RCVYTIME	Date/time recovery was last performed or REGENERATE used. RCVYTIME is formatted "YYYYJJJHHMMSSTH", for year, Julian date, hours, minutes, seconds and tenths of seconds. This value is the same as that returned by DTSLRCVY.
RECADD	Records started in table B not including extension records (rate or total).
RECDEL	Records deleted from table B not including extension records or records deleted by DELETE RECORDS or IFDSET (rate or total).
REQ	Number of requests and DISPLAY PROCEDURE commands run on the file (rate or total).
RETRYA	Number of table A rehashes required to add an item to table A. This statistics returns the total for the file since it was initialized. A high value for RETRYA could indicate that table A is getting close to being full.
RETRYC	Number of table C rehashes required to add an item to table C. This statistics returns the total for the file since it was initialized. A high value for RETRYC could indicate that table C is getting close to being full.
SECTY	Summed hex value indicating type of security in effect for file: X'01' - Procedure security. X'04' - Procedures secured after Release 4.
SPILADD	Number of records added to spill areas in table B. Same as the Model204 SPILLADD parameter (rate or total).
SPILDEL	Number of records deleted from spill areas in table B. Same as the Model204 SPILLDEL parameter (rate or total).
UPDTDUR	Number of milliseconds the file has been part of at least on update unit. This includes the time for Model 204 to mark the file physically consistent (the action that completes the update unit). Presented by SirMon in seconds.
UPDTIME	Date and time file was last updated. UPDTPIME is formatted "YYYYJJJHHMMSSTH", for year, Julian date, hours, minutes, seconds and tenths of seconds. This stat is the same as Model 204's file statistic DTSLUPDT.

XFULLP	Estimate of the percentage of table X that is full. This statistic is equal to the table X highest active page (XHIGHPG) divided by the total number of table B pages (XSIZE).
XHIGHPG	The current table B high water mark. Same as the Model204 XHIGHPG parameter.
XQLEN	The number of pages in the table X reuse queue. Same as the Model204 XQLEN parameter.
XRECPPG	Maximum number of records per Table X page.
XRESERVE	The amount of space in bytes on a Table B page on which new records cannot be started. The space is reserved for the expansion of existing records on the Table B page.
XREUSED	The number of records added to the file that reused the record number of a previously deleted record. Same as the Model204 XREUSED parameter.
XSIZE	Size of table X in number of pages.

APPENDIX B *Date Processing*

SirFile uses dates in the following ways:

- To examine the CPU clock (as returned by the STCK hardware instruction) to determine the current date, in case *SirFile* is under a rental or trial agreement
- To display the current date, as returned by the TIME SVC, modified by the SYSDATE parameter or the the *Sir2000 User Language Tools* APPDATE clock, as page headers in various end-user displays
- To keep track of the date and time of various changes to a file, to enable predictions about file conditions, using the TIME SVC, modified by the SYSDATE parameter or the the *Sir2000 User Language Tools* APPDATE clock

For headers on pages or rows that occur on printed pages or displayed screens, Sirius Software products generally use a full four-digit year format, although they may display dates with two-digit years in circumstances where the proper century can be inferred from the context.

Index

B

Background monitor ... 31-33
 Invocation techniques ... 33
 Predict frequency ... 32
 Predictions to journal ... 31
 time interval setting ... 31
 User list for predictions ... 32
 User list for warnings ... 31
 Warning frequency ... 31
 Warnings to journal ... 31
 Background Monitor Specification Screen ... 31
 Batch Refresh ... 35

C

Commands ... 9, 13-14, 18, 23, 25-26, 32
 -F (Find backward) ... 13, 26
 -n (scroll backward) ... 13, 26
 Add ... 14
 Close ... 9, 14, 18, 26, 32
 DELETE ... 14
 F (Find forward) ... 13, 26
 n (scroll forward) ... 13, 25
 Open ... 9, 14, 18, 26, 32
 Refresh ... 14
 Sort ... 26
 Time ... 14, 23
 auto-refresh mode ... 23

F

File Detail screen ... 23
 File open screen ... 21
 File Statistics ... 37
 File Tables screen ... 25
 File thresholds screen ... 17
 Files ... 6-7
 enqueueing ... 6
 files available for monitoring ... 6
 initial load ... 7

M

Main File menu ... 13
 Messages screen ... 29

O

Overview ... 5

R

Refresh ... 5, 11
 defined ... 5

T

Thresholds ... 7-9
 ARETRIES ... 7
 BFULLP ... 7
 CFULLP ... 8
 CFULLP stat collection ... 9
 CRETRIES ... 7
 DFULLP ... 8
 EOVFLADD ... 8
 EXTNADD ... 8
 Number of days advance warning ... 9
 Number of days between samples ... 9
 Number of samples to keep ... 8
 OVLFFADD ... 8
 SPILLADD ... 8

U

UL/SPF
 constituent products ... 1
 integrating with other subsystems ... 2
 introduction to ... 1

