



Rocket M204 SirPro

User's Guide

September 2013
PRO-0704-UG-01

Notices

Edition

Publication date: September 2013

Book number: PRO-0704-UG-01

Product version: Rocket M204 SirPro

Copyright

© Rocket Software, Inc. or its affiliates 1989-2013. 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.

Contact information

Website: www.rocketsoftware.com

Rocket Software, Inc. Headquarters

77 Fourth Avenue

Waltham, MA 02451-1468

USA

Tel: +1 781 577 4321

Fax: +1 617 630 7100

Contacting Global Technical Support

If you have current support and maintenance agreements with Rocket Software and CCA, contact Global 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	ii
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: SirPro Main menu	5
Chapter 3: File/Procedure Specification	7
Chapter 4: Procedure List	11
Move/Copy	14
Deleting Procedures	15
Printing Procedures	16
Rename Procedures	18
SmartBrowse -- Browse, Variable and (#) Expand.	19
Chapter 5: Model 204 Command processing	23
Chapter 6: Print All Information	25
Chapter 7: Field Definition Scan	27
Chapter 8: CCASTAT maintenance (User, File and Group password privileges)	29
Chapter 9: CCAGRP maintenance (Group definitions)	33
Appendix A: Change management tools and functions	37
Sequencing: Creating working copies of procedures	38
XCOMPARE: Creating an update procedure	42
New: Create a new procedure via update deck	45
K (Klobber): Managed Update procedure erasure	47
ReSequence: Refreshing internal sequence numbers.	48

Contents

Appendix B: Change Management Overview 51

Appendix C: Date Processing 53

Appendix D: Terminal MODEL 6 Support 55

Index 57

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>)

SirPro (Option 1 from the *UL/SPF* Main Menu) provides tools for programmers, database administrators and application managers. An interface is provided to any user-specified procedure file, to a list of commands, to field definitions in *Model 204* files, and to passwords, privileges and group definitions held in *CCASTAT* and *CCAGRP*.

```
----- SirPro V5.00 ----- ULSPF500/4.1.1E/CMS ----- 99-02-03 09:56:48 -----  
=>  
  
1. Procedure List  
2. Command Processor  
3. Print All Information  
4. Field Definition Display  
5. User Password Table  
6. File Password Table  
7. Group Password Table  
8. View/Maintain Group Definitions  
  
X. Quit  
  
-----  
1/Help          3/Quit
```

SirPro Main Menu

Each *SirPro* suboption is described in the chapters that follow.

File/Procedure Specification

Procedure-level commands are accessed through Option 1 on the SirPro Main Menu. *SirPro* displays a screen that allows selection of a procedure file and criteria for specifying the procedure(s) to be displayed.

```

----- SirPro ----- ULSPF500/4.1.1E/CMS ----- 99-02-01 21:50:58 -----
==>

          Context ==> FILE
    File/Group Name ==> SIRULSPF
          Password ==>

    Procedure Name ==>
    Last Updater ID ==>

    Search Strings:
    _____ ==>
    or ==>
    or ==>
    or ==>
    or ==>

    Display:
    # of lines of code ==> 0

-----
1/Help          3/Quit

                                     11/Switch to *UPPER

```

The Procedure File selection screen

Supply values in the fields shown, as described below:

Context

Valid input values are FILE, GROUP, TEMP GROUP, PERM GROUP, and blank. *SirPro* attempts to open only in the specified context. If a blank is input, *SirPro* searches for an entity (using the value specified in **File/Group Name**) in the default *Model 204* search order, that is, it looks for a TEMP GROUP, then a PERM GROUP, and then a FILE.

File/Group Name

The name of a procedure file or file group. If a file, the file must be allocated to the Online *Model 204* region.

To display a list of all available files, enter a question mark (?), or leave this field blank. When you press **ENTER**, *UL/SPF* shows a list of all available files. You select a file from this list by typing **S** in the selection column to the left of the desired file. The **Password** field, below, lets you open a file from this screen.

If your site's METADATA is not populated, this screen is not displayed.

Password The password for the file or group, if needed.

Procedure Name The name of the procedure to be edited. To display a list of all procedures for a given file, enter a question mark (?) in this field, or leave it blank. To display a partial list of procedures, use the asterisk (*) wild card character, for example: *XYZ, XYZ*, or *XYZ*.

If a single procedure name is specified, a *Model 204* edit session is immediately invoked on the procedure. If the procedure name field is left blank, or if a wild card selection was entered, a list of procedures is presented (described in next section).

Last Updater ID Optional entry that will narrow the set of procedures from a selected file to only those that were last updated by the specified account (user ID). The asterisk (*) wild card character is also supported in this field.

Search Strings As many as five optional search strings may be entered in these fields. The procedures that match the other selection criteria (file, procedure wild card, and account) are scanned, and only those that contain any of the requested string(s) appear on the list of procedure names. If there are more than 100 procedures to be scanned, you are warned about the possibility of a long search. To override this warning, press **PF12** when prompted by the warning message.

The string search is done in mixed case if you are currently in *LOWER mode, and it is done in uppercase if you are currently in *UPPER mode. Case mode can be determined by the setting of the "Switch" PF key at the bottom of the screen. (If the user is in *LOWER mode, *Switch to *UPPER* is shown as the PF key prompt).

The Search String fields **do not** strip blanks before searching. *SirPro* will distinguish between all of the following searches:

```
'%X=$GETG('SNOPES')'  
'%X = $GETG('SNOPES')'  
'%X=$GETG('SNOPES')
```

Judicious use of the Erase-EOF key should prevent confusion when trailing blanks might be present.

PF keys for this panel are:

PF1 for help

PF3 to quit to the SirPro Main Menu

PF11 to toggle between *LOWER (mixed case edit) and *UPPER case mode

When **enter** is pressed *SirPro* either invokes an Edit session on the selected procedure, or passes the user to a list of procedures for further selection.

The command line at the top of the screen allows fastpath transfer to other *UL/SPF* options, and allows the following commands.

CLOSE xxxxxxxx Closes the specified file or group.

CHAPTER 4 *Procedure List*

```

----- FILE: JANUS ----- ULSPF500/4.1.1E/CMS ----- 99-02-01 21:52:01 -----
==>                                     Total Procs = 293
Sel   Procedure Name                    Account   Bytes   Date     Time
  1   /JCL/BATCH2.TXT                    ALAN      252   98/04/22 12:59:20
  2   /JCL/PIPE.TXT                      ALAN      317   98/04/22 12:59:00
  3   /PICS/EG1.GIF                      ALAN    33774  98/04/22 12:58:48
  4   /PICS/EG2.GIF                      ALAN    35331  98/04/22 12:59:22
  5   /PICS/EG3.GIF                      ALAN    30336  98/04/22 12:59:13
  6   /PICS/EG4.GIF                      ALAN    35520  98/04/22 12:58:55
  7   /PICS/EG5.GIF                      ALAN    41330  98/04/22 12:59:16
  8   /POP/A.CLASS                       ALAN    10723  98/04/22 12:59:02
  9   /POP/B.CLASS                       ALAN     6438  98/04/22 12:59:19
 10   /POP/BRGRAPH.HTM                   ALAN      674  98/04/22 12:58:47
 11   /POP/C.CLASS                       ALAN     1979  98/04/22 12:59:22
 12   /POP/D.CLASS                       ALAN     2759  98/04/22 12:59:04
 13   /POP/E.CLASS                       ALAN     6687  98/04/22 12:58:58
 14   /POP/F.CLASS                       ALAN     4640  98/04/22 12:59:04
 15   /POP/G.CLASS                       ALAN    16651  98/04/22 12:59:11
 16   /POP/H.CLASS                       ALAN     5031  98/04/22 12:58:53
 17   /POP/I.CLASS                       ALAN     3790  98/04/22 12:58:54
----- ULSPF.120: Updates not allowed.
1/Help  2/Sort-Name  3/Quit  4/Sort-User  5/Sort-Date  6/Sort-Size
7/Up    8/Down    9/Repeat 10/Refresh 11/*UPPER 12/FULLNAME

```

The Procedure List screen

The procedure list shows the procedures that fit the criteria specified on the selection panel. A prefix area to the left of each procedure name allows entry of a variety of single-character procedure commands.

Supply a value in the select field (SEL), as described below:

- B** Browse
- C** Copy
- D** Delete
- E or S** Edit
- I** Include
- M** Move
- P** Print
- R** Rename
- V** Variable analysis
- #** Expanded Browse

In addition, the following *managed update commands* are provided for users of the *SirLib* change management system.

- X** Xcompare
- Q** seQuence

- N** New
- K** Klobber
- Z** resequence (or ReSequence or resequenZ)

The managed update commands are described in [“Change management tools and functions” on page 37](#).

Users may enter any number and combination of prefix commands, none of which will execute until **ENTER** is pressed. **Deletes** and **Prints** are done for all procedures at once. Processing for other commands is performed one procedure at a time, but with all same-type commands processed sequentially (that is, all Edits are performed, then all Browses, then all Prints, etc.) in the order in which commands are encountered from the top of the procedure list.

The display area of the screen, described below, can be toggled between a set of views. The DLIST view, similar to the view of procedures most commonly seen by User Language developers at the command level, contains the following information:

PROCEDURE NAME The name of a procedure available for editing. Procedure names longer than 25 characters are truncated. An asterisk is added as the last character of the truncated procedure name to indicate that the full name is not displayed, and the procedure name is highlighted. These procedures can still be selected from the list, and the full procedure name is used in all functions.

ACCOUNT The account (user id) of the last user to update the procedure.

BYTES Number of bytes stored in Table D for the procedure.

DATE/TIME The date and time the procedure was last updated.

PF Keys:

The following PF keys invoke actions as described:

- PF1** Full screen help.
- PF2** Sort the displayed list by procedure name.
- PF3** Quit (return to previous screen).
- PF4** Sort the displayed list by last updating user.
- PF5** Sort the displayed list by last date updated.
- PF6** Sort the displayed list by size of procedure.
- PF7** Scroll backward (up).

- PF8** Scroll forward (down).
- PF9** Repeat last command-line command.
- PF10** Refresh the list. Use this PF key to clear any prefix selections and to recheck the file for procedure information that may have changed (for instance, if another user is deleting procedures in the same file, your list may be out-of-date).
- PF11** Toggles between upper and mixed case **E**dit when the editor is invoked. The PF key label indicates the case to which you will switch when the PF key is pressed.
- PF12** Toggle the display between a “D LIST” format and a “FULLNAME” format. In DLIST format, the display shows the traditional *Model 204* procedure list information with long procedure names truncated at 25 characters (for display only). In FULLNAME format, long procedure names overlay their corresponding procedure information.

The following commands are valid on the command-line of this screen:

- L n** Locate line number n.
- n or -n** Scroll n or -n lines from the top line of the display.
- F string or /string** Find a specific “string”, starting the search from the top line of the display. This string can be any part of the procedure name. That is, it is not a prefix search.
- F string or -/string** Find a specific “string”, searching backward from the top line of the display.
- TOP** Moves the display to the top of the procedure list.
- BOTTOM** Moves the display to the bottom of the procedure list.
- M** When using **PF7** or **PF8**, scroll up or down to the top or bottom of the list of procedures.
- CLOSE xxxxxxxx** Closes the specified file or group. *SirPro* APSY files, and the file(s) from which the current procedure list was taken, cannot be closed.
- END or QUIT** Same as **PF3**, returns the user to previous screen.

In addition, any of the prefix commands may be typed on the command line, followed by either a procedure name, or the number of that procedure in the displayed list. If the procedures selected in this manner is the first procedure selected, processing is initiated

immediately. If other prefix commands have already been entered, the procedure is added to the selected list.

Include and **Edit** (**E** or **S**) are not documented further, as they merely invoke a *Model 204* edit or include. The other prefix commands invoke special screens and processing, which are documented in the sections that follow.

4.1 Move/Copy

The destination screen is displayed once for each procedure selected for **Copy** or **Move**. The screen requires entry of destination file and procedure name, and allows the procedure to be edited on the way to the new location. The source information is protected.

```
----- COPY PROCS ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 11:59:17 -----  
==>  
  
Copy Procedure From:  
  
  Proc File: JANUS  
  Procedure: JANCATN_STORE_COLUMNS  
  
Copy Procedure to:  
  
  Context:  
  Proc File: JANUS                               Password:  
  Procedure: JANCATN_STORE_COLUMNS  
  
  Replace Existing Procedure? (Y/N): N  
  Edit New Procedure? (Y/N): N  
  
-----  
1/Help           3/Quit
```

The Move/Copy Procedure screen

The copy screen requires input in the following fields:

FROM FILE

The name of the file from which you are moving/copying. This field is protected.

FROM PROC

The name of the procedure from which you are moving/copying. This field is protected.

DESTINATION FILE

The name of the file to which you want to move/copy the file.

DESTINATION FILE PASSWORD

The file password, if needed.

DESTINATION PROCEDURE NAME

The procedure can be renamed during Copy/Move operation. If the file is being copied or moved to the same file, it must be renamed.

REPLACE Y/N

Type Y to replace if the procedure already exists. The default is N.

EDIT Y/N Type Y to edit the new procedure before *UL/SPF* performs the move/copy. The default is N.

If you type Y, editing takes place before *UL/SPF* stores the procedures in the "TO" procedure and leaves the source procedure unchanged. The procedure status line (the top line of the editor display screen) shows that you are editing a temporary procedure into the destination procedure name. When you finish editing the procedure, type END in the command window. *UL/SPF* stores the new procedure in the destination procedure name. To terminate the edit process and abandon edit changes and the Copy, type QUIT.

PFKEYS - The following PF keys invoke actions as described:

PF1 Full screen help.

PF3 Cancel the request to copy or move this procedure (if this is one of a list of selected procedures, the next procedure is presented; if this is the only procedure, or if it is the last procedure in a list, you are returned to the procedure list display).

4.2 Deleting Procedures

Procedures selected for deletion are presented in a single scrollable list, from which the user presses **PF3** to cancel (and return to the list where the selections can be changed), or **PF12** to confirm the deletions.

```
----- FILE: ALANPROC ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 11:59:40 -----
==>                                     Total Procs = 10
Sel   Procedure Name                    Account   Bytes   Date    Time
D 1   ABRULES                           ALAN     399    97/04/28  9:03:43
D 2   ADHOC_ADD_A_COLUMN                 ALAN    6715   96/09/04 13:44:04
D 3   ADHOC_APSY_USERS                   ALAN    4337   95/11/29 23:32:05
D 4   ADHOC_APSYS                        ALAN    3450   95/11/29 20:58:29
D 5   ADHOC_CCASYS                       ALAN    134    98/09/23 10:07:38
D 6   ADHOC_RPC_DRIVER                   ALAN    1937   98/10/13 15:14:57
D 7   AGENTS_DEFINE                      ALAN    262    95/03/02 14:58:51
D 8   AGENTS_DETAIL                      ALAN    2692   94/05/22 17:54:51
D 9   AGENTS_LIST                        ALAN    3751   98/04/01 13:03:57
D 10  ALANWEB_DEFINE                     ALAN    255    97/06/24 15:38:01

----- ULSPF.032: Use PF12 to confirm or PF3 to cancel DELETE.
1/Help  2/Sort-Name  3/Quit  4/Sort-User  5/Sort-Date  6/Sort-Size
7/Up    8/Down        9/Repeat 11/*UPPER  12/Confirm
```

Confirming procedure deletion

PF keys are the same as on the procedure selection list, except that **PF10** is not active (the user must return to the previous screen to refresh the procedure list), and **PF12** is used as a confirmation key for the delete.

On returning to the original list, procedure information lines are overwritten with a success or failure message for the delete operation.

4.3 Printing Procedures

A variety of options are provided for printing procedures. Procedures may be routed to an output file, or to a printer via a **USE printer.id** command or a **USE \$PRINT** command.

No confirmation is performed for print operations. The user is routed directly to the Print Destination screen for routing.

The Print Destination screen controls output routing of the procedures, and various characteristics of the print job.

```

---- FILE: ALANPROC ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 11:59:50 -----
==>

      Output Specification for Selected Procedures.

      Destination ==>   (Select from options 1 through 3 below)

      1. Dataset DDNAME   ==>
      2. Printer ID       ==>
      3. $PRINT Class     ==>

              with      ==> COPIES=1
                        ==>

      Lines Per Page      ==> 60 (UDDLPP)
      Characters per Line ==> 133 (UDDCCC)
      Record Format        ==> 12 (UDDRFM)
      Header Control      ==> 3 (HDRCTL)

      Output for procedure reload (Y/N)   ==> N
      Use SIRPRO Formatting (Y/N)        ==> N
-----
      1/Help                               3/Quit

```

Print routing screen

Enter the following information:

OUTPUT DESTINATION The destination of the procedure. Select 1 to print to a the printer named in the PRINTER field, 2 to output the procedures to the file named in the FILE field, and 3 to invoke a USE \$PRINT, to the specified CLASS and PRINTER.

PRINTER A valid predefined printer name. If this field is blank or invalid when a 1 is specified in the OUTPUT DESTINATION field, the print comes to the user's terminal.

OUTFILE A valid preallocated sequential file.

CLASS The job class to use for USE \$PRINT requests. This job class should be a valid class at your site. USE \$PRINT is an obsolete method of printing, and its use is discouraged. If the PRINTER is blank or invalid when a USE \$PRINT is invoked, the print is routed to PRINTER=LOCAL by *Model 204*

WITH criteria Enter in these two fields any print routing criteria normally placed in the "WITH" clause of a USE statement (such as COPIES=xx, TAG or ID statements).

UDDLPP The number of lines to be printed or displayed per page. The default is 60 lines per page.

- UDDCCC** The number of characters to be printed or displayed per line. The default is 133 characters per line.
- HDRCTL** Header control settings allow the user to suppress various levels of automatic print headers.
- UDDRFM** The number of characters to be printed or displayed per line. The default is 133 characters per line.
- Use Sirius Formatting?** Specifying “N” to this prompt prints procedures as they are in the procedure file. Specifying “Y” forces a page break at the beginning of each procedure, numbers the lines of the procedure and produces a header at the top of each page, indicating the procedure file, procedure name, date, and time of the print.
- Format for file reorg?** Specifying “Y” to this prompt prints procedures beginning in column 1 and placing PROCEDURE <procname> and END PROCEDURE <procname> statements around each output procedure. Procedures output to a file using this formatting feature can be input directly to a procedure file reorganization job. Specifying “Y” to this prompt overrides a “Y” to the previous prompt (Use Sirius Formatting?), because file reorganization format precludes the type of headers and line numbers used in Sirius Formatting.

PF keys are the same as on the procedure selection list, except that **PF10** is not active (the user must return to the previous screen to refresh their selected list), and **PF12** is used as a confirmation key for the print.

If either Option 1 or Option 2 is selected, *SirPro* executes a simple USE command. If Option 3 is selected, a command of the format:

```
USE $PRINT PRINTER=printer.id CLASS=print.class
```

is executed. If the specified printer is invalid, PRINTER=LOCAL is substituted.

All user entries on the print screen are saved in the user profile except the destination.

4.4 Rename Procedures

Procedures selected for **R**enaming are presented one at a time.


```

----- Rename Procedure -----
==>

Rename Procedure From:

  Proc File:  ALANPROC

  Procedure:  AGENTS_DETAIL

Rename Procedure To:

  Procedure:  AGENTS_DETAIL

-----

1/Help          3/Quit
  
```

Renaming procedures screen

On returning to the original list, procedure information lines are overwritten with a success message for renamed procedures, and they are protected from update until the user's procedure list is refreshed.

PFKEYS - The following PF keys invoke actions as described:

- PF1 Full screen help for this screen.
- PF3 Cancel the request to rename this procedure.

4.5 SmartBrowse -- Browse, Variable and (#) Expand.

Browse, **V**ariable and **#** (Expand) prefix commands all invoke SmartBrowse sessions. **B**rowse displays the procedure in a scrollable list (which does not invoke a edit session). **E**xpand (**#**) brings in-line all INCLUDED code, and presents the fully expanded procedure in a scrollable list (INCLUDEs of temporary procedures are ignored). **V**ariable scans the fully expanded procedure and presents a list of undeclared %variables.

```

----- Browse: COUNTER.WORDS (Lines=69) -----
==>
                                Line:   37   Cols: 1   To 77
%S = $PROCGET
IF %S EQ '' THEN
  IF %M GT 500 THEN
    PRINT 'PROC: ' WITH %P WITH ': ' TAB %M
  END IF
  LOOP END
END IF
%S = $UNBLANK(%S)
%U = %S
JUMP TO (LOOPY) $INDEX(%S, '*')
%J = 0
REPEAT WHILE $LEN(%U)
  %U = $PARSEX(%U, ' ,=)(+*/><')
  %J = %J + 1
END REPEAT
%H = %H + %J
IF $SUBSTR(%S, $LEN(%S), 1) NE '-' THEN
  IF %H GT %M THEN
-----
1/Help          3/Quit
7/Up            8/Down          9/Repeat
  
```

SmartBrowse: Browsing a procedure

```

----- Undeclared variables for: COUNTER.WORDS (Lines=74) -----
==>
                                Line:    1   Cols: 1   To 77
%ABERCROMBIE
%FITCH
%ROW.COUNTER
%COL.COUNTER
%MAX

----- *** Top of list ***
1/Help          3/Quit
7/Up            8/Down          9/Repeat
  
```

SmartBrowse: Displaying undeclared variables

All three of these commands invoke the same procedure, so the user may switch their browse session from one view to another with the following commands:

- BROWSE** Convert the display to a simple browse session.
- UNDECL** Display undeclared variable names only.
- VARIABLE** Display undeclared variable names only (same as UNDECL).
- EXPAND** Expand all include statements.

The following command-line commands are also valid:

- | | |
|------------------------------|--|
| L n | Locate line number n. |
| n or -n | Scroll n or -n lines from the top line of the display. |
| F string or /string | Find a specific “string”, starting the search from the top line of the display. This string can be any part of the procedure name. That is, it is not a prefix search. |
| -F string or -/string | Find a specific “string”, searching backward from the top line of the display. |
| TOP | Moves the display to the top of the procedure list. |
| BOTTOM | Moves the display to the bottom of the procedure list. |
| M | When using PF7 or PF8 , scroll up or down to the top or bottom of the list of procedures. |

PF Keys:

The following PF keys invoke actions as described:

- | | |
|------------|--|
| PF1 | Full screen help for this screen. |
| PF3 | Exit from SmartBrowse of this procedure. |
| PF7 | Scroll up. |
| PF8 | Scroll down. |
| PF9 | Repeat last command-line command. |

CHAPTER 5 *Model 204 Command processing*

To execute *Model 204* commands from a *UL/SPF* input panel, select Option 2 (Command processor) from the *SirPro* Main Menu. *SirPro* displays the *Model 204* Command Input screen.

```

----- M204 Commands ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 12:02:53 -----
==>

Cmd: ALLOCATE ALANDMP WITH OLD SEQUENTIAL DSNAME=ULSPF.V500.ALANDMP

                                Command stack:
1. ALLOCATE ALANDMP WITH OLD SEQUENTIAL DSNAME=ULSPF.V500.ALANDMP
2. LOGWHO
3. V SYSTEM
4.
5.
6.
7.
8.
9.

-----
1/Help                3/Quit
                      9/Retrieve

```

Model 204 Commands input screen

Note: You cannot execute subsystem commands from this input panel (a *Model 204* restriction). Because *UL/SPF* operates in the *Model 204* Application Subsystem (APSY) framework, you cannot use certain commands. Other commands, particularly file-related table commands, must be prefaced with the “IN filename” clause.

On this screen, three lines — each of which holds 70 characters — and a fourth line of 45 characters are available for entering standard *Model 204* commands. If you are entering a command that is longer than 70 characters, use continuation marks at the end of that line. Commands up to 255 characters long may be entered.

The last 9 commands executed are stored in global variables, and the first 72 characters of each is displayed in the “stack” on the lower half of the screen. These commands are stored in the user profile from session to session, and may be recalled on this screen via **PF9**, which cycles through the stack, or by the number shown to the left of each command. Pressing **ENTER** executes the command.

```
----- Command output ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 12:02:24 -----  
====> Line: 1 Cols: 1 To 77  
VERSION 4.1.1E RELEASE OF MODEL 204  
CMSVERSN 3.2.1A VERSION OF THE CMS INTERFACE  
SYSOPT X'BB' SYSTEM OPTIONS  
OPSYS X'2014' OPERATING SYSTEM  
SYSID CMS SMF SYSTEM IDENTIFICATION  
JOBNM ULSPF500 JOB NAME  
STEPNM 046864 STEP NAME  
JESID JES JOB ID  
LOCATION D204 LOCATION  
LSERVPD 8176 LENGTH OF SCHEDULER PDL  
NUSERS 35 NUMBER OF USERS  
NSERVS 3 NUMBER OF SERVERS  
NJBUFF 14 NUMBER OF JOURNAL BUFFERS  
CACHE X'00' CACHE FAST WRITE  
LAUDPROC 21 LENGTH OF AUDIT PROCEDURE NAMES  
SVIOMAXT 10 SERVER TRACKS PER USER  
SNAPID 0 NUMBER OF SNAPS SO FAR  
SNAPLIM 1 0 OR MAX NUMBER OF SNAPS ALLOWED  
SNAPCTL X'28' SNAP/DUMP OPTIONS  
SNAPPDLX 0 SNAP PDL EXTENSION OPTION  
----- *** Top of list ***
```

Model 204 Command output display

The output shown above is the captured output of the command executed in the previous example screen. Command output of any type, length and width can be captured, scrolled up and down (with **PF7** and **PF8**), and left and right (with **PF10** and **PF11**).

CHAPTER 6 *Print All Information*

The Programmer's PAI Facility helps User Language developers by running an extract of records in order to:

- View a sample of the record structure stored in the *Model 204* file.
- Extract a sample of the data for reloading into a test data base.

To use the Programmer's PAI Facility, select Option 3 from the SirPro Main Menu. *UL/SPF* displays the Print All Information Facility screen.

```

----- PAI ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 12:04:09 -----
==>
Use Parms:
      (Enter: Output dataset DDNAME, or Printer specifications)

File Name: CCASYS           Password:

      Field Name           Operator           Value
APSUNM                    =                SIRMON
APSURTYP                   =NOT             UDEF

-----
1/Help           3/Quit
11/CHECKOFF 12/Clear
  
```

Programmers' PAI facility

Supply values in the fields shown, as described below:

USE PARMS The default directs the output to the terminal. To direct the output elsewhere, specify the output destination DDNAME or the printer ID. Use the following syntax:

- To local printer** \$PRINT CLASS=X
- To VTAM printer** PRINT QUEUE XXXX
(the printer must have the VTAM printer support option)
- To a data set** OUTXXXX

FILE NAME The name of the file (up to 8 characters) to be searched.

PASSWORD The password for the file, if needed.

FIELD The field name to search for.

OPERATOR A standard *Model 204* operator command; for example, =, IS GT, IS LT.

VALUE The value of the field to search for.

Press **PF11** to toggle between CHECKON and CHECKOFF. CHECKON checks whether the field names are defined to the file before execution by looking in the *Model 204* dictionary for the file/field combination.

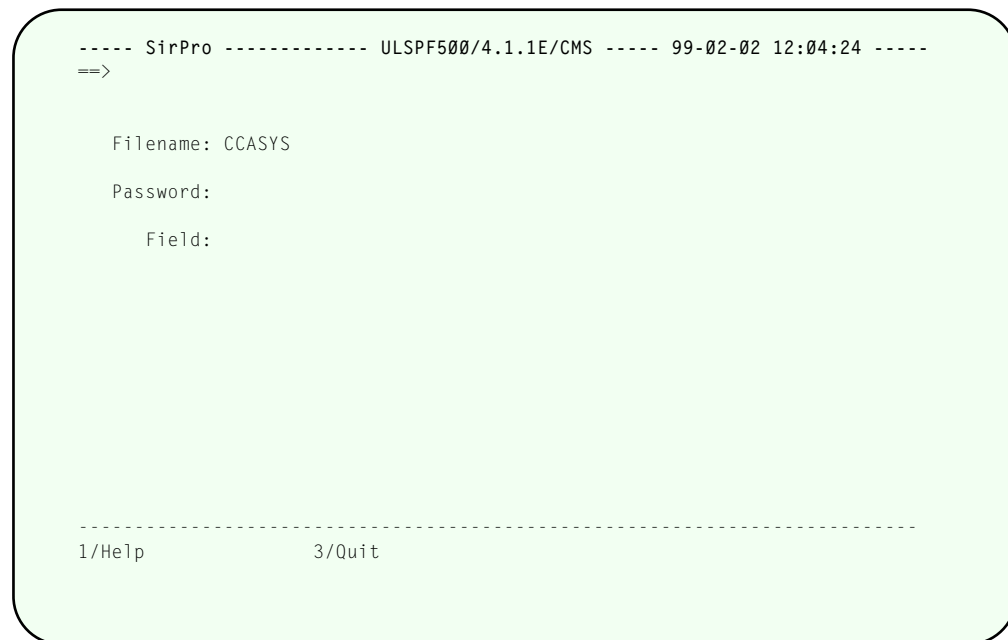
CHECKOFF executes without first checking the field names against the Dictionary. Use CHECKOFF when editing is not required or the file data are not available from the *Model 204* dictionary.

As long as you remain in *UL/SPF*, the PAI input panel will retain your latest input. If you leave *UL/SPF*, you will lose the contents of the screen.

CHAPTER 7 *Field Definition Scan*

Option 4 on the SirPro Main Menu lets a user scan the field definitions of a selected file. The scan is taken directly from Table A of the selected file. As with other *UL/SPF* panels, you may request a list of available file names or explicitly enter one. If you are only interested in a select set of field names that start with, contain, or end with a particular character string, you may use wild card search strings to narrow your selected set of field names.

You must provide a valid password that grants authority to view the field definitions. If you do not open the file correctly, an error occurs, and you will be unable to view the field definitions. This input screen is displayed in the figure that follows.

A screenshot of a terminal window with a light green background. At the top, it shows the SirPro logo and system information: "----- SirPro ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 12:04:24 -----". Below this is a command prompt "==" followed by a right-pointing arrow. The screen displays three prompts: "Filename: CCASYS", "Password:", and "Field:". At the bottom, there is a dashed line and two options: "1/Help" and "3/Quit".**Display Field definitions from target file**

The display screen for field definitions shows the field name (up to 70 characters) with its associated attributes displayed on the line below. You may search for a selected string by entering a search command in the command window. If you search for a string that is both an attribute and part of a field name (for example, KEY), the search process will find each instance of the string value whether it is contained in a field name or attribute line.

```
----- SirPro ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 12:04:29 -----  
==> Field Count: 43  
Field: APSALL  
DEF NFRV KEY NCOD STR NNR VIS UP NORD NUNIQ REPT  
  
Field: APSAPRIV  
NFRV NKEY NCOD STR NNR VIS UP NORD NUNIQ REPT  
  
Field: APSCACCT  
NFRV NKEY NCOD STR NNR VIS UP NORD NUNIQ REPT  
  
Field: APSCCLSS  
DEF NFRV KEY NCOD STR NNR VIS UP NORD NUNIQ REPT  
  
Field: APSCPRV  
NFRV NKEY NCOD BIN NNR VIS UP NORD NUNIQ REPT  
  
Field: APSCRCTY  
NFRV NKEY NCOD STR NNR VIS UP NORD NUNIQ REPT  
  
-----  
1/Help          3/Quit  
7/Back   8/Forward  9/Repeat
```

Display Field definitions (output)

Note: Line movement in this panel is by field name. Field names and attributes are always paired in the screen and never split across screen boundaries.

CCASTAT maintenance (User, File and Group password privileges)

Options 5, 6 and 7 from the SirPro Main Menu provide a full screen interface to the *Model 204* password table (CCASTAT). Option 5 displays existing User passwords, 6 displays File passwords and 7 displays passwords for Groups. These displays are only available to users with System Manager privileges, and access to CCASTAT is always in update mode. Other users do not see Options 5, 6 or 7 on the *SirPro* Main Menu.

```

----- User privileges ---- ULSPF500/4.1.1E/CMS ----- 99-02-02 10:25:17 -----
==> -
Sel  User      Password Privs  Priority  Terminals
-----
      bethesda  00      NONE     ALL
      ALAN      FF      HIGH     ALL
      ALAN2     FC      NONE     ALL
      ALEX      FF      NONE     ALL
      ALEX2     FF      NONE     ALL
      BART      10      NONE     ALL
      CHARLIE   FF      NONE     ALL
      DAN       00      NONE     ALL
      DARREN    FF      HIGH     ALL
      DME       FF      NONE     ALL
      DMETEST   44      NONE     ALL
      DME2      FF      NONE     ALL
      DME3      FF      NONE     ALL
      DME4      FF      NONE     ALL
      DME5      FF      NONE     ALL
      GARY      FF      NONE     ALL
      HOMER     00      NONE     ALL
-----
1/Help  2/Sort-User  3/Quit  4/Sort-Priv  5/Sort-Prio  6/Add-Users
7/Up    8/Down      9/Repeat 10/Refresh  11/View     12/End
  
```

User entries in CCASTAT (user password table)

The **User password** display shows the User Id, a password field which can be toggled between visible and invisible, the user privilege bits (UPRIV), priority and the list of terminals the user is permitted to use. For more information about these display fields, see the M204wiki LOGCTL and LOGLST command descriptions (for example, http://m204wiki.rocketsoftware.com/index.php/LOGLST_command).

A one-character prefix input field allows users to specify C or D, for Change or Delete. If any information is changed on a particular line, it is the same as if a C was entered. New and changed information is committed immediately to CCASTAT when **ENTER** or any PF key is pressed. New users are added to CCASTAT when they are added to the empty lines at the bottom of the list of existing users. Note that user ID cannot be overwritten.

The standard PF keys are supported on this screen; **PF1** accesses help text, **PF3** quits without saving the current updates, **PF7** scrolls up through the displayed list, **PF8** scrolls down, **PF10** repeats the last command-line command, and **PF12** refreshes the display with current CCASTAT information and exits the CCASTAT display session, saving any changed information.

The following PF keys access screen-specific actions:

- PF2** Sort the displayed users by their user id.
- PF4** Sort the displayed users by their privilege bits.
- PF5** Sort the displayed users by their priority and user id.
- PF6** Add new users. This PF key scrolls the list of existing CCASTAT user entries to the bottom, and positions the cursor on the first blank line of the screen. This action is the same as manually scrolling the list to the bottom and typing new entries into the blank lines that follow. Information added in these lines is always considered to be a request to add users.
- PF11** Toggle the password display to **VISIBLE**. This PF key allows users to see the new and changed passwords they are typing in. It also displays the LOGCTL commands as they execute (a “diagnostic” display will scroll by on the screen). Note that passwords existing in CCASTAT are never displayed: *SirPro* is not aware of the current value of password entries in CCASTAT, and cannot backward encrypt the entries. This function cannot be used to view existing values of passwords.

The File password display (Option 6 on the SirPro Main Menu) is similar to the User password display. All File entries existing in CCASTAT are shown.

```

----- File privileges ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 10:25:57 -----
==> _
                                     |----Priv Levels--|
Sel FileName Idx Password Privs  N  S  R  U  A  Terminals
ALANPROC                                     0CCC 0  0  0  0  0  ALL
ALANPROC 1                                     BFFF 0  0  0  0  0  ALL
ALANPROC 2                                     0201 0  0  0  0  0  ALL
ASDF                                          BFFF 0  0  0  0  0  ALL
BACKUP                                       8761 0  0  0  0  0  ALL
BACKUP 1                                     BFFF 0  0  0  0  0  ALL
BACKUP01                                    8761 0  0  0  0  0  ALL
BACKUP01 1                                   BFFF 0  0  0  0  0  ALL
BATPROC 1                                     BFFF 0  0  0  0  0  ALL
BATPROC 2                                     0221 0  0  0  0  0  ALL
BGPROC 1                                     BFFF 0  0  0  0  0  ALL
BGPROC 2                                     0221 0  0  0  0  0  ALL
CALPOL                                       BFFF 0  0  0  0  0  ALL
CCASYS 0                                     BFFF 0  0  0  0  0  ALL
CIE                                          BFFF 0  0  0  0  0  ALL
CLIENTS                                     BFFF 0  0  0  0  0  ALL
-----
1/Help    2/Sort-File  3/Quit      4/Sort-Priv   6/Add
7/Up      8/Down      9/Repeat   10/Refresh   11/View     12/End

```

File entries in CCASTAT (file password table)

The displayed fields are a one-character input field (Sel), the Filename, Index (the index entry for multiple passwords on the same file), Password, Privilege bits, Privilege levels and terminals. For more information about these fields, see the M204wiki LOGCTL and LOGFILE command descriptions (for example,

http://m204wiki.rocketsoftware.com/index.php/LOGFILE_command).

The same common PF keys as for User Passwords is supported on this screen.

In addition, the following PF keys perform special functions.

- PF2 Sort the displayed files by name.
- PF4 Sort the displayed files by their privilege bits.
- PF6 Add new files. This PF key scrolls the list of existing CCASTAT file entries to the bottom, and positions the cursor on the first blank line of the screen. This action is the same as manually scrolling the list to the bottom and typing new entries into the blank lines that follow. Information added in these lines is always considered to be a request to add new files.
- PF11 Toggle the password display to **VISIBLE** and show a “diagnostic” list of LOGCTL commands as they execute. The same caveats apply as for PF11 on the User Password screen.

The Group password display is identical to the File password display. All CCASTAT entries for Groups are displayed.

```
----- Group privileges --- ULSPF500/4.1.1E/CMS ----- 99-02-02 10:26:09 -----
==> _
                                     |----Priv Levels--|
Sel FileName Idx Password Privs  N  S  R  U  A  Terminals
  ALANGRP   1          BFFF 0   0   0   0   0  ALL
A
A
A
A
A
A
A
A
A
A
A
A
A
A
A
A
-----
1/Help    2/Sort-File  3/Quit      4/Sort-Priv  6/Add
7/Up      8/Down      9/Repeat    10/Refresh   11/View     12/End
```

Group entries in CCASTAT (group password table)

The displayed fields have slightly different meanings for groups than they do for files. These differences can be found in the *Model 204 Command Reference Manual* under the LOGCTL and LOGGRP commands.

CCAGRP maintenance (Group definitions)

Option 8 on the SirPro Main Menu provides a full-screen facility for managing group definitions in CCAGRP. This option is only available to system managers.

```

----- CCAGRP entries ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 10:26:21 -----
==> _
Total Groups: 44
|-----Priv Levels-----|
Sel  Group  Access  Priv  Bldgft  Procfile  Updtfile  N  S  R  U  A
_   BENDER  PUBLIC  BFFF  Y  *
OLPROC  BGPROC  BATPROC
_   BIGDDMY  SEMIPUB  BFFF  Y  *  ALANPROC  =20 45 0 0 1
SIRULSPF SIRPRO  SIRFILE  SIRXREF  SIRSCAN  SIRMON  SIRFIXES  DEVULSPF ...
_   BIGJUNK  SEMIPUB  BFFF  Y  *  ALANPROC  =20 45 0 0 1
SIRULSPF SIRPRO  SIRFILE  SIRXREF  SIRSCAN  SIRMON  SIRFIXES  DEVULSPF ...
_   BLAH    PUBLIC  BFFF  Y  *  ALANPROC
ALANPROC DEVMON
_   BYPASS  PUBLIC  BFFF  Y  SIRULSPF
SIRULSPF
_   DEVAUDT PUBLIC  BFFF  Y  *
DEVAUDT  SIRAUDT
_   DEVELOP PUBLIC  BFFF  Y  *
DEVULSPF DEVPRO  DEVLIB  DEVMON  DEVSCAN  DEVFILE  DEVXREF  DEVJAN ...
_   DEVFILE PUBLIC  BFFF  Y  *
DEVFILE  SIRFILE
-----
1/Help      3/Quit
7/Up        8/Down    9/Repeat  10/Left   11/Right

```

CCAGRP interface - Permanent group definitions

For each group defined in CCAGRP, *SirPro* displays two lines of information. The first line contains the group name and parameters, and the second line contains the files defined to the group. The parameters shown for each group are:

- Access** PUBLIC, SEMIPUB or PRIVATE
- Privileges** Hexadecimal switches associating various privileges with the default opening of PUBLIC and SEMIPUB groups.
- BLDGFT** “Y” to build global find table, “N” to not build this table when the group is opened. “Y” is the default.
- PROCFILE** Filename of the group member that will contain procedures. If all members may contain procedures, an asterisk (*) is used.
- UPDTFILE** File that should be the target of record adds when no specific member name context is given.
- Priv Level N** Procedure class level.

Priv Level S Class level required for SELECT

Priv Level R Class level required for READ

Priv Level U Class level required for UPDATE

Priv Level A Class level required for ADD

The *Model 204 Command Reference Manual* provides more detail on the meaning of each of these parameters.

The *SirPro* CCAGRP interface imposes a limit of 24 files per group. If you have groups defined with more than 24 files, Option 8 on the SirPro Main Menu may be used to view them, but should not be used to update them, as the 25th and later files will be deleted from the definition. *SirPro* will prevent Update, Copy, or Rename commands against groups with more than 24 members.

Prefixing each group name is a single-character command field. Valid commands in these fields are:

- D** Deletes the group definition.
- R** Rename the group.
- C** Copy the group definition to a group definition of a new name.
- U** Update: Change the parameter settings for the files defined to the group definition.

Delete and **Rename** are executed directly from the CCAGRP screen. **Copy** and **Update** send the user to a group definition update screen. In addition, the command-line command **ADD groupname** sends the user to the same screen. The update screen is described below.

PF keys on the Group Definition Display are:

- PF1** Help.
- PF3** Quit (return to SirPro Main Menu).
- PF7** Scroll up.
- PF8** Scroll down.
- PF9** Repeat last command-line command.
- PF10** Scroll file lines to the left.

PF11 Scroll the file lines to the right. If there are more than 8 files in a group definition, an ellipsis (. . .) will appear to the right of the eighth file. This PF key shifts the view to the right to display the remaining files.

Copy, Update and ADD commands invoke the following display, which allows an entire GROUP definition to be built.

```

----- CCAGRP entries ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 10:26:31 -----
==> _

      Group  Access  Priv  Bldgft  Procfile  Updfile  |-----Priv Levels-----|
BIGJUNK_ SEMIPUB_ BFFF   Y   *_____ ALANPROC =20 45_ 0_  0_  1_

                * * * Files * * *
SIRULSPF SIRPRO_  SIRFILE_ SIRXREF_ SIRSCAN_ SIRMON_  SIRFIXES DEVULSPF
DEVPRO_  DEVFILE_ DEVXREF_ DEVSCAN_ DEVMON_  ALANPROC ALEXPROC _____

-----
----- ULSPF.189: Enter new group information then save with PF12.
1/Help          3/Quit
                9/Repeat          12/Save-End

```

CCAGRP interface - Updating a group definition

Access, Privileges and Bldgft are set to their *Model 204* default values. The user should enter values for the remaining fields and press **PF12** to build the group.

PF keys valid for this display are:

- PF1** Help.
- PF3** Quit (return to SirPro Main Menu).
- PF9** Repeat last command-line command.
- PF12** Commit the definition to CCAGRP. This key invokes error checking before the group definition commands are executed.

Change management tools and functions

Managed update commands are available if the change management system, *SirLib*, is active in the Online. Managed update commands allow the manipulation and promotion of update procedures, “deltas”, instead of entire procedures or entire procedure files. The output from all managed update commands that produce update procedures always contain a single-line comment describing the creator, date, time and source file for the change. The managed update commands are entered as prefix commands on the Procedure List screen (see “[Procedure List](#)” on page 11). These prefix commands are:

X *Xcompare.*

Compares two versions of a procedure and creates a third procedure containing the differences between the two. This output is an “update procedure”, the physical unit managed by the change management system.

Q *Sequence.*

Creates both a working and sequenced version of a procedure, which are the later input to Xcompare.

N *New.*

Creates a “differences” update procedure similar to Xcompare, but which is the source for a new procedure.

K *Erase.*

Creates an update procedure similar to Xcompare output, but telling *SirLib* to perform a logical delete on the procedure.

Z *ReSequence.*

Creates a special update procedure to reorder the internal sequence numbers of a procedure.

The Procedure List screen is the main working environment for the programmer in the *SirPro/SirLib*-controlled environment. This screen, and others like it, provide access to all procedure activity in *Model 204*. Details on the use of *SirLib*, including the special commands for managed updates, are contained in the ***SirLib User's Guide***. Only the special Procedure List screen prefix commands (Q, X, K, N and Z) for creating and manipulating managed updates are explained here.

The Q, X, N, K and Z prefix commands are used to generate 'working', base and sequenced copies of procedures and to generate updates from those copies. The programmer types one of these managed update commands to the left of the selected procedure and presses **ENTER**. *SirPro* then presents one of the managed update screens to get further details on the managed update. Those screens are shown in the following detailed sections.

A.1 Sequencing: Creating working copies of procedures

“Sequencing” is the managed update method by which programmers make copies of procedures to work on. In other systems, programmers make backup copies of procedures and then work on the original. In *SirPro* programmers never alter the original program code, but execute “Q” commands to get working and sequenced copies from which update procedures can be generated.

```
--- Sequence Procedure ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 13:39:41 ---
==>

Input procedure
Proc file          ==> SIRSCAN
Procedure          ==> SCNP-START

Output procedures
Proc file          ==> DEVSCAN           Password ==>
Unsequenced procedure ==> SCNP-START
Sequenced procedure ==> SEQ.SCNP-START

The FixFile must contain control procedure CONTROL.SIRSCAN
and all updates for file SIRSCAN

FixFile            ==> SIRFIXES           Password ==>

Replace existing procedures? (Y/N) ==> N
-----
1/Help             3/Quit
```

Sequencing a procedure with the "Q" command

Prefixing a procedure with the Q command on the Procedure List screen tells *SirLib* to generate both a copy of the selected procedure (the copy that the programmer will work on) and a “sequenced” copy of the procedure (a copy containing sequence or line numbers), which will later be used as comparison input for the Xcompare command. Building these copies is always the first step in generating a managed update. The Q prefix command presents the **Sequence Procedure** screen, which has the following input fields:

Output Procedure File

Enter a procedure file where *SirLib* should create a working and SEQ. version of the selected procedure.

Password for Output Procedure File

An update password for this file is required unless the file is already open with update privileges.

UnSequenced Copy

Procedure name for the copy of the selected procedure that the programmer will be making updates to. This defaults to the input procedure name.

Sequenced Copy

Procedure name for a copy of the procedure with each line prefixed by a sequence number, indicating line sequence and amount of update activity that has been applied in generating the working version.

The default procedure name for sequenced procedures is:

`SEQ.<procname>`

This default prefix may be overridden if it conflicts with local APSY naming standards.

FixFile name

File containing Control and Update procedures for this file. Default file is SIRLIBP.

FixFile password

A password for this FixFile is required unless the file is public or is already open with read privileges. The privileges for this password must allow the contents of procedures in the control file to be read.

Control Procedure Name

is displayed in the block of text above the FixFile input field for informational purposes only. This procedure, formatted as follows:

`CONTROL.<filename>`

where <filename> is the name of the file from which the selected procedure is being copied must exist in the specified FixFile. This CONTROL procedure contains the project names active for this file. The update procedures that are pointed to by the project identifiers in the control procedure must be in the same file.

Replace Existing Procedure

This field defaults to “N”, indicating that if a same-named procedure already exists in the destination procedure file, the attempt to copy over it will be aborted. If a user wishes to overwrite existing procedures this indicator should be set to “Y”.

When the “Q” prefix command is executed from the Procedure List screen, *SirLib* looks in the source file for a procedure named ***BASE.<procname>***, where ***<procname>*** is the name of the selected procedure. If no such procedure is found then *SirLib* does a copy of the selected procedure into the specified target procedure name in the specified Output file. In addition, *SirLib* generates a sequenced copy of the procedure, named whatever the programmer specified in the Sequenced Copy field on this screen. The sequenced copy of the procedure is identical to the unsequenced (or working) copy, except that each line is prefixed with a sequence number.

If *SirLib* does find a *BASE.<procname>* procedure then it looks for the *CONTROL.<filename>* procedure in the specified FixFile. A control procedure of the specified name must exist or the Q prefix command fails. Noting the active projects in the control procedure, *SirLib* finds all update procedure names that match the pattern:

<file>.<project>.<procname>

Where *<file>* is the name of the input file for this operation, *<procname>* is the name of the selected input procedure, and *<project>* is one of the identifiers in columns 1 to 8 of the specified control procedure. All update procedure names that conform to this pattern are sorted by their project and then by the sequence line numbers they effect. This aggregate update is then applied to *BASE.<procname>* to produce the sequenced and unsequenced output procedures. In other words, the programmer doesn't just get a copy of the procedure they request, they get a generated procedure that is the BASE of what they want, with all active updates applied.

If a project name is commented out in the control procedure, it is not included in the resulting sequenced and unsequenced output procedures. A Project name is commented out by placing an asterisk (“*”) in front of it in the control procedure. (This is how working versions of production procedures can be generated in development regions if the development region has active projects that do not exist yet in production).

The sequenced version of a procedure should never be changed, as it is the “before” image for the XCOMPARE that will eventually generate the update procedure. Sequence numbers should not concern programmers for the most part, though there are a few times when it is worth knowing something about how they work.

The “Q” prefix command automatically begins sequencing at 10000 and increments each line in the *BASE.<procname>* procedure by 10000. As changes are generated, the XCOMPARE function generates new line numbers for lines of code being inserted and replaced. The XCOMPARE algorithm attempts to number the first new line of code beginning with a sequence number 1 greater than its starting point in the existing sequence (this applies to Inserts and Replaces: it is irrelevant for Deletes). Subsequent new lines which are part of the same update are incremented by a power of 10 less than the last sequence of numbers for the section of code. Later changes that apply to the same section again begin numbering at their starting sequence number plus 1, and continue to increment at the next lowest available power of 10.

Fortunately, the programmer doesn't have to understand any of this in order for it to work. The lower-order digit(s) of the sequence numbers in a SEQ procedure however will indicate the number of times a section of code has been changed. The sequence numbers themselves will indicate whether the XCOMPARE will be able to “fit” changes into the same section of code again. For example, a section of code with sequence numbers that look like this:

```
003451001
003452001
003453001
```

indicates that changes will still “fit” in the hundreds and tens columns. If an update procedure attempts to insert 3 lines of code after line 003452001, the resulting code will be numbered like this:

```
003451001
003452001
003452102
003452202
003452302
003453001
003454001
```

Replace (a “/R” command in an update procedure) operates slightly differently than Insert, but the outcome looks very similar. When changes will no longer “fit”, XCOMPARE can still successfully generate an update procedure that effects this section of code, but the update procedure will become unnecessarily long and *SirLib* might lose its ability to detect update collisions. In this case, a Resequence is required. Resequencing is described later in this chapter.

Once an initial update has been coded and the update procedure placed in the FixFile, all subsequent “Q” prefix commands against the same procedure will generate copies containing the update, and sequence numbers in output sequenced versions of the procedure will reflect the insertion of the update code.

A.2 XCOMPARE: Creating an update procedure

The XCOMPARE operation is executed by placing an “X” next to the changed procedure on the Procedure List screen. This invokes a compare operation of that procedure against its sequenced version. The output from this operation is an update procedure.

```
--- Build Update Deck ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 13:45:17 ---
==>

Unsequenced input procedure
Proc file      ==> DEVSCAN
Procedure      ==> SCPR-SCAN2

Sequenced input procedure
Proc file      ==> DEVSCAN                Password ==>
Procedure      ==> SEQ.SCPR-SCAN2

Output update procedure
FixFile        ==> SIRFIXES                Password ==>
Project Name   ==>
Target File    ==>

Replace existing update? (Y/N) ==> N
Enter editor for update? (Y/N) ==> N

Synchronization count          ==> 2
Unlock this procedure? (Y/N)   ==> Y
-----
1/Help                3/Quit
```

Xcompare screen - creating a "differences" deck

Prefixing a procedure by an X in the Procedure List screen tells *SirLib* to compare the specified procedure against a matching “sequenced” procedure, and to generate another procedure containing the differences between the two. The X prefix command presents the **Xcompare screen**, which has the following input fields:

UnSequenced Procedure and File

Protected fields displaying the source file and procedure.

File Name for Input Sequenced Copy

Source file of sequenced procedure. This defaults to the same file as for the input unsequenced procedure.

Password

Read access is required for the above file. No password need be input if the file is already open or has high enough default privileges to read procedures.

Proc Name for Sequenced Copy

Procedure name of sequenced procedure. “SEQ.” is the prefix generated by the Q command, so Xcompare defaults to this.

FixFile filename

Destination for update procedure output from this XCOMPARE.

FixFile Password

Write access to this file is required. If the file is already open, or if default privileges for the file are high enough to allow the creation of a new procedure, no password is required.

Project name

This is the 8-character project name to which this update should be linked.

The *SirLib* file administrator may have required that updates for this file only be linked to existing projects, in which case the entered project name is checked to make sure it exists in the control procedure. If this setting is not on, updates may be created, and the project entered in the control procedure later.

Target File

This is the production file containing the base procedure against which this update will apply.

Replace Existing Update

Entering “Y” in this field allows the user to overwrite a same-named update procedure in the destination file with the current update procedure. Programmers are always permitted to overwrite update procedures created under their userid when this switch is set to “Y”.

The *SirLib* file administrator may have specified in the Administration options that programmers may not overwrite each other's update procedures, either in this file or within the entire system. If this option is set, and an update procedure exists of the same name as the one being created in the target file, and the userid who created the update procedure (or last updated it) is not the same as the current user's ID, then the generation of the update will not be permitted.

Enter Editor for update

Entering “Y” at this prompt places the generating user into an edit session on the new update procedure. The update procedure will have already been stored before the edit session is invoked, so QUIT may be used to exit the edit session without losing the update procedure.

Synchronization Count

This field allows programmers to set the number of lines that will be compared before the sequenced and unsequenced versions of the procedures are considered to be back in sync, and further lines are no longer to be generated to the output update procedure (at least until the next mis-match).

The default sync count value is 2, and this will work well for the vast majority of cases. Increasing the sync count may reduce the number of difference lines found, resulting in smaller update procedures, or it may have no effect

at all beyond a minor performance penalty in the XCOMPARE operation. In a very few cases it may produce larger output procedures.

An entered value of asterisk (*) tells XCOMPARE to repeat the compare operation, increasing the sync count from 1 until it hits a value that no longer produces a smaller output procedure. As might be imagined, this is the worst-performing option for running XCOMPARE, but will almost always generate the smallest possible output procedure. Unless changes are massive and many, the size of the output procedure is fairly irrelevant, and playing with the sync count has little utility.

Unlock Procedure

Entering "Y" in this field makes the procedure available to other programmers in systems or files where procedures are locked to one updating user at a time.

Any value but "Y" allows the programmer to generate an update procedure but will retain the exclusive lock on the procedure (such as when a test integration is being run, and the programmer doesn't wish to relinquish control of the procedures being updated). If procedure locking is not turned on, the lock/unlock information is still maintained for historical reporting, but this switch has no actual effect in locking procedures. See Administration options for Procedure Locking options.

A.3 New: Create a new procedure via update deck

Prefixing a procedure by an N in the Procedure List screen tells *SirLib* to generate an update procedure, similar to that produced by X, but containing all lines of the specified procedure (because it is a new procedure to the system).

```

--- Create New Proc ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 13:46:15 ---
==>

New procedure name
  Proc file      ==> DEVSCAN
  Procedure      ==> SCPR-WEB1

Output update procedure
  FixFile       ==> SIRFIXES           Password ==>
  Project name  ==> F5TESTX
  Target file   ==> SIRSCAN

Replace existing update? (Y/N) ==> N
Enter editor for update? (Y/N) ==> N

-----
1/HELP          3/QUIT

```

Generating a "New" procedure

Prefixing a procedure by an N in the Procedure List screen presents the **Create New Proc** screen, which has the following input fields:

New Procedure and Procedure Source File

Protected fields displaying the source file and procedure name for this update procedure creation.

FixFile name

Destination for the output update procedure.

FixFile password

Write access to this file is required. If the file is already open, or if default privileges for the file are high enough to allow the creation of a new procedure, no password is needed.

Project

A valid project identifier (up to 8-characters).

The *SirLib* file administrator may have required that updates for this file only be linked to projects previously defined in the control procedure. In this case the project entered is verified to see that it exists before the update is generated. If this setting is not on, updates may be generated first, and the project names entered in the control procedure later.

Target File

The name of the production file containing the base procedure to which this update will apply.

Replace Existing Update

Entering “Y” in this field allows the user to overwrite a same name update procedure in the destination file with the current update procedure. Programmers are always permitted to overwrite update procedures created under their userid when this switch is set to “Y”.

The *SirLib* file administrator may have specified in the Administration options that programmers may not overwrite each other's update procedures either in this file or within the entire system. If this option is set, and an update procedure exists of the same name as the one being created in the target file, and the userid who created the update procedure (or last updated it) is not the same as the current user's ID, then the generation of the update procedure will not be permitted.

Enter Editor for update

Entering “Y” at this prompt places the generating user into an edit session on the new update procedure. The update procedure will have already been stored before the edit session is invoked, so or QUIT may be used to exit the edit session without the update procedure being lost.

A.4 K (Klobber): Managed Update procedure erasure

Prefixing a procedure by a K in the Procedure List screen tells *SirLib* to generate an update procedure which contains only a special command to tell *SirLib* that the “change” to this procedure is a delete. Managed update deletes are always logical deletes, allowing deleted procedures to be recovered at any time when the file is reconfigured. For this reason, an empty (zero-line) version of the erased procedure is left in the file, and should not be physically deleted.

The K prefix command presents the **Klobber Procedure** screen, which has the same input fields as for the Create New Proc screen described in [“New: Create a new procedure via update deck” on page 45](#).

```
--- Klobber Procedure ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 13:46:38 ---  
==>  
  
Procedure to erase  
Proc file      ==> SIRSCAN  
Procedure      ==> SCLB-SAVE.PROFILE  
  
Output update procedure  
FixFile        ==> SIRFIXES           Password ==>  
Project name   ==> F5TESTX  
Target file    ==> SIRSCAN  
  
Replace existing update? (Y/N) ==> N  
Enter editor for update? (Y/N) ==> N  
  
-----  
1/Help          3/Quit
```

"K" (klobber) Generating a logical procedure deletion

A.5 ReSequence: Refreshing internal sequence numbers.

The sequence numbers generated by *SirLib* are for the most part of no concern to the programmers working on procedures. However, if a large number of update procedures are generated that affect the same section of code in a procedure, further changes to the same section may result in unnecessarily large update procedures. In addition, *SirLib* will lose its ability to detect when updates are overwriting each other.

In case this rare event occurs, a manager can generate a resequence procedure By placing a Z in front of the procedure name on the Procedure List screen, the **ReSequence Proc** screen is produced:

```
--- ReSequence Proc ----- ULSPF500/4.1.1E/CMS ----- 99-02-02 13:46:48 ---
==>

Resequencing procedure
  Proc file      ==> SIRSCAN
  Procedure      ==> SCREEN.SCPR-SCAN2

Output update procedure
  FixFile       ==> SIRFIXES           Password ==>
  Project name  ==> F5TESTX
  Target file   ==> SIRSCAN

Replace existing update? (Y/N) ==> N
Enter editor for update? (Y/N) ==> N

-----
1/HELP          3/QUIT
```

"Z" Resequencing a managed procedure

The **ReSequence Proc** screen has the following input fields:

Procedure and File for Resequencing

Protected fields displaying the source file and procedure name for this update procedure creation.

FixFile name

Destination for output update procedure.

FixFile password

Write access to this file is required. If the file is already open, or if default privileges for the file are high enough to allow the creation of a new procedure, no password is needed.

Project A valid project identifier (up to 8-characters).

The *SirLib* file administrator may have required that updates for this file only be linked to projects previously defined in the control procedure. In this case the project entered is verified to see that it exists before the update is generated. If this setting is not on, updates may be generated first, and the project names entered in the control procedure later.

Target File

The name of the production file containing the base procedure to which this update will apply.

New starting number

The default is to begin renumbering with 10000.

New increment

The default to to renumber in increments of 10000.

Replace Existing Update

Entering “Y” in this field allows the user to overwrite a same name update procedure in the destination file with the current update procedure. Programmers are always permitted to overwrite update procedures created under their userid when this switch is set to “Y”.

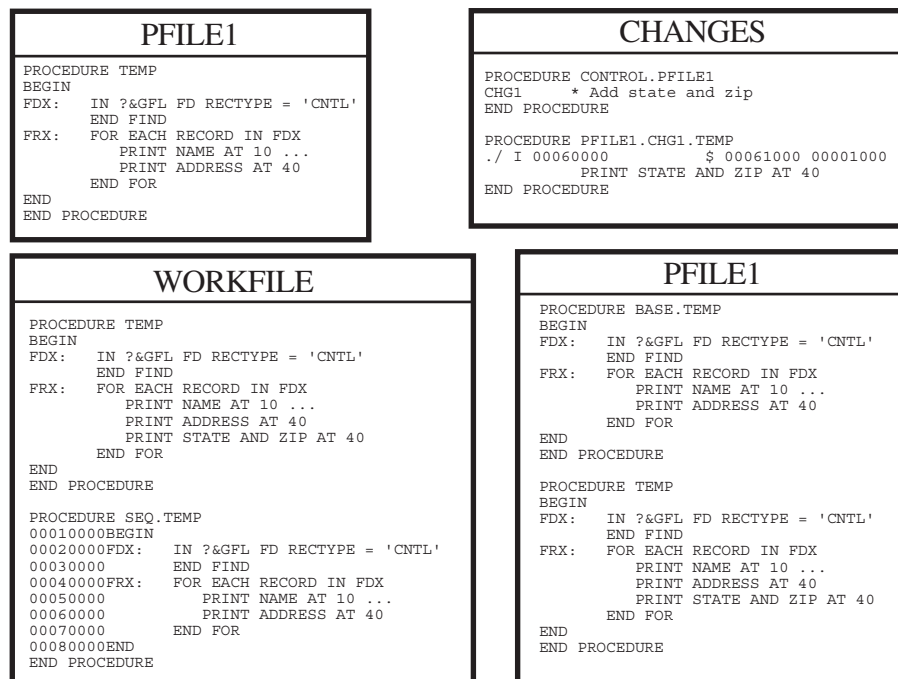
The *SirLib* file administrator may have specified in the Administration options that programmers may not overwrite each other's update procedures either in this file or within the entire system. If this option is set, and an update procedure exists of the same name as the one being created in the target file, and the userid who created the update procedure (or last updated it) is not the same as the current user's id, then the generation of the update procedure will not be permitted.

Enter Editor for update

Entering “Y” at this prompt places the generating user into an edit session on the new update procedure. The update procedure will have already been stored before the edit session is invoked, so or QUIT may be used to exit the edit session without the update procedure being lost.

APPENDIX B *Change Management Overview*

Notice that no sequenced copy of a procedure is needed for New, Erase, or Resequence operations. Q (seQuence) is the managed update equivalent of a Copy, just as N (New) is the equivalent of Create, K (Klobber, or Erase) is the equivalent of Delete. XCOMPARE has no direct equivalent in the “unmanaged” command structure, though coupled with later reconfiguration using the output update procedure it is analogous to moving finished changes to a staging area. Resequencing is of course a special purpose function with no “unmanaged” corollary.



How a change progresses through *SirLib*

In the example above, procedure TEMP exists in file PF1E1 (upper left). The programmer uses the “Q” command to copy TEMP into WORKFILE, getting both a working and a “SEQ” version of the procedure. A single line of new code is inserted in TEMP (the line “PRINT STATE AND ZIP AT 40”). Using XCOMPARE, the update is generated to file CHANGES, and named PF1E1.CHG1.TEMP. A project leader defines the project name CHG1 in CONTROL.PF1E1. Finally, *SirLib*'s ReConfiguration function, described in the next section, is used to create a “BASE” version of the procedure, so the update can be backed out if necessary, and the original procedure TEMP is deleted. The update procedure PF1E1.CHG1.TEMP is applied to BASE.TEMP to create the new executable procedure TEMP. The final state of PF1E1 is shown lower right.

APPENDIX C *Date Processing*

SirPro 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 *SirPro* 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 display, and sort by, the last-updated timestamps from the *Model 204 Procedure Dictionary*
- To keep track of the date and time at which a user issued a CHECKIN, CHECKOUT, CONFIGURE, or CUTOVER *SirLib* operation, 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, *UL/SPF* 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.

 APPENDIX D *Terminal MODEL 6 Support*

SirPro users can take advantage of the additional available screen space offered by terminal models beyond the standard Mod 2 (24 X 80), Mod 3 (32 X 80), Mod 4 (43 X 80), and Mod 5 (27 X 132). The new terminal models are supported by setting the terminal model to 6:

RESET MODEL 6

There's really no such thing as a Model 6 terminal, but setting the terminal model to 6 tells *Model 204* to issue a Write Structured Field Query to the terminal to have the terminal indicate its geometry (number of rows and columns) to *Model 204*. In this way, *Model 204* can dynamically set a terminal's geometry, whether it's one of the standard geometries (Mod 2, 3, 4, or 5) or not. Many terminal emulators allow alternate 3270 sizes to be set. This makes it possible to set the terminal geometry to match the optimal combination of font size and physical screen size for a particular workstation, rather than trying to set the emulator font size to work well with one of a limited number of screen geometries.

Unfortunately, the standard User Language screen definitions don't allow the defining of fields that extend beyond column 79. However, \$scrHide, \$scrSize, and \$scrWide make it possible for User Language screens to take advantage of columns beyond column 79. In addition, these functions make it possible to dynamically modify screen definitions to allow a single screen definition to work with an arbitrary variety of screen sizes. While these functions are a bit awkward to use and somewhat limited, they're not unreasonable for building dynamic scrolling screens — scrolling screens being particularly suited for larger screen geometries.

To facilitate User Language applications for varying screen sizes, the VIEW command for the MODEL parameter is enhanced to show the screen geometry in addition to the model number for model 6 terminals:

```
> V MODEL
MODEL      6 34*142    TERMINAL MODEL
```

So \$view issued for the above terminal returns 6 34*142, from which a User Language application could readily determine that the screen has 34 rows and 142 columns.

To enable model 6 support, the SIRTERM system parameter must be set in the CCAIN stream. This is a bitmask parameter with the following meanings for the bits:

X'01' Enable MODEL 6 support.

X'02' Always issue a Write Structured Field Query for terminals connecting to *Model 204* through VTAM. This allows *Model 204* to dynamically determine the screen geometry of any terminal connecting to it through VTAM, without having to issue a RESET MODEL 6 command.

The downside of this setting is that it could add a small amount of time to the initial connection process and a slight amount of extra network traffic.

If a terminal is using a non-standard screen geometry via model 6 support, the *Model 204* editor and command line will correctly use the available screen space.

Index

A

Administration
 Overwrite ... 43, 46, 49
 requiring identifiers ... 43, 45, 49

B

Browse
 SmartBrowse command ... 20

C

CCAGRP maintenance ... 33
 group definitions ... 33
 CCASTAT maintenance ... 29
 Command window ... 23
 Copy
 destination and options screen ... 14

D

Deleting procedures ... 15

E

Expand
 SmartBrowse command ... 20

F

File and Procedure specification
 selecting a file and subsetting procs ... 7

G

Group definitions
 Adding ... 34
 copying ... 34
 deleting ... 34
 renaming ... 34
 updating ... 34

M

Main menu ... 5
 Managed updates
 described ... 37
 MODEL parameter, VIEW command ... 55

Move
 destination and options screen ... 14

P

PAI facility
 using ... 25
 Print All Information facility ... 25
 Printing
 \$PRINT CLASS ... 17
 characters per line ... 17
 formatting for file reorganization ... 18
 header control ... 18
 lines per page ... 17
 OUTFILE ... 17
 output designation ... 17
 PRINTER ... 17
 record format ... 18
 using Sirius formatting ... 18
 WITH criteria ... 17
 Printing procedures ... 16
 Procedure List
 Browse ... 11
 Copy C ... 11
 Delete ... 11
 Edit ... 11
 entering prefix commands ... 11
 Include ... 11
 Move ... 11
 Print ... 11
 procedure multi-function panel ... 11
 Procedure list (managed update commands)
 Klobber ... 12
 New ... 11
 seQuence ... 11
 Xcompare ... 11
 Z (resequence) ... 12
 Procedures
 Delete ... 15
 Print ... 16
 Production fixes
 generating procs to match production ... 40

R

Rename procedure ... 18
Resequence ... 41

S

Sequence numbers
 how they work ... 40
SirPro
 Summary ... 5
SIRTERM system parameter ... 55
SmartBrowse ... 19
 Browse ... 19
 Expand ... 19
 displaying expanded procedures ... 19
Variable ... 19
 identifying undeclared variables ... 19

U

UL/SPF
 constituent products ... 1
 integrating with other subsystems ... 2
 introduction to ... 1
Undecl
 SmartBrowse command ... 20

V

Variable
 SmartBrowse command ... 20