Rocket Model 204 SirScan

User's Guide

Notices

Edition

Publication date: July 2013

Book number: SCN-0704-UG-01

Product version: Rocket Model 204 SirScan

Copyright

Trademarks

Examples

License agreement

Contact information

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

Contents

_____
CHAPTER 1   *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**, 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)

CHAPTER 2 *Overview and Setup for SirScan*

2.1   SirScan system parameters are essential

For versions of Model 204 before 7.5, the installation of the *Sirius Mods* is required and sufficient to install and run the SIRSCAN subsystem. However, out of the wrapper, SIRSCAN has some limitations in selecting and formatting journal entries. These are caused by the fact that journal records contain only a user number and do not contain other potentially useful identifying information, such as userid, terminal ID, or IP address. So when *SirScan* scan specifications in the SIRSCAN subsystem are based on things like userid, IP address, or port name, there is often no way to determine whether a single journal entry is associated with the requested attributes or not.

In fact, before *Sirius Mods* version 6.0, selection criteria based on userid would only work for currently logged on threads, and then only for journal entries from the current session. This is because, for logged in users, *SirScan* can determine user number and login time. From these two pieces of information, it can be sure that all journal entries for the thread associated with a requested userid, and with a date/time stamp after the login time of the user, are associated with that user.

Outside of being very limiting — it was impossible to select journal info based on userid for non-logged in users — *SirScan* could also produce anomalous behavior in auto-refresh mode ("Auto-refresh mode" on page 20). Because *SirScan* only collects data from the previous end of the interval when in auto-refresh mode, if a userid present and selected in a previous interval logged off before the data is refreshed for the new interval, none of the journal records for that user would appear between the end of the previous interval and the end of the current. Outside of the annoyance of not seeing the expected data, this could cause confusion among *SirScan* end-users, because a user's journal entries might seem to suddenly stop without a logoff record or any other hint as to why.

In *Sirius Mods* 6.0 and later, *SirScan* makes more of an effort to use whatever information is available in the journal to associate journal records with userids or other attributes not actually on the journal records. For example, if *SirScan* sees a M204.0352 message for a login, it knows that from that time on, all journal records for that thread number are associated with the userid on the M204.0352 message, until the end of the interval or a M204.0352 logout message is reached for that thread.

In a similar way, *SirScan* can use the MSIR.0114 message, which indicates a Janus port and client IP address for any Janus server request, to associate a thread number with an IP address and port name. Unfortunately, while this approach makes it possible to use non-thread-number and non-thread-type based selection criteria for sessions that are no longer active, it also makes *SirScan*'s behavior even more anomalous and confusing, because whether or not a journal record is selected might depend on whether or not *SirScan* happened to see an identifying message in the journal for that thread.

For example, if user MOE was logged on from 8:37 to 11:43, and a request for MOE's records is made for the interval 8:38 to 9:30, none of MOE's records would be seen. But if the selected interval was 8:37 to 9:30, they would all be seen.

The behaviour of *SirScan* in auto-refresh mode would be even more confusing: whether a thread's information is collected would depend on whether identifying messages were found in the new part of an interval on a refresh, so users' records would seem to inexplicably stop in midstream, only to mysteriously reappear after a user exits and re-enters the scan screen for essentially the same interval.

So to fix this anomalous behavior, *Sirius Mods* version 6.0 provided *SirScan* "heartbeat" messages. These heartbeat messages are RK type audit messages, and they contain identifying information about the thread such as userid, terminal ID, and IP address and port number for Janus server threads. The heartbeat messages are controlled by the SCANTIME system parameter.

2.1.1   Using the SCANTIME system parameter

SirScan heartbeat messages are controlled by the SCANTIME system parameter, which must be set with the User 0 parameters in the CCAIN stream. This parameter indicates the maximum number of seconds between journal messages for a thread before a heartbeat message is issued. If a message is about to be sent to the journal for a thread, and it has been more than SCANTIME seconds since the last heartbeat message, then the heartbeat message is first sent to the journal. This means that if a journal message is found for a thread, the userid and other identifying information can **always** be found by looking backwards in the journal no more than SCANTIME seconds.

The default value for SCANTIME is 0, which means that heartbeat messages are not logged. If SCANTIME is set to a positive value up to its maximum value of 3600, the *SirScan* heartbeat messages will be logged to the journal.

If SCANTIME is set to a non-zero value, the SIRSCAN subsystem detects this and, on the scan specification screen, gives users the option of reading an extra SCANTIME seconds on every interval collected from the journal. The extra SCANTIME seconds read are before the start of the requested interval. While the data from the extra SCANTIME seconds worth of journal entries is not formatted, it is scanned for the heartbeat messages, and any information in those messages is saved for each thread. In this way, all messages for all threads after the start of the requested interval can be deterministically associated with a userid, terminal ID, and IP address and Janus port for Janus server requests.

The cost of getting this deterministic behavior is the overhead of scanning an extra SCANTIME seconds worth of journal. If scanning is being done in auto-refresh mode, each refresh (attempt to move past the current bottom) will cause an extra SCANTIME seconds to be scanned. This means that the setting of SCANTIME involves a trade-off between the number of heartbeat messages logged to the journal and the overhead of scanning an extra SCANTIME seconds of journal on every *SirScan* browse request.

The bigger the value of SCANTIME, the fewer heartbeat messages will be logged, but the more journal records would need to be scanned on each *SirScan* browse request. A small value of SCANTIME will, of course, have the opposite effect.

Since journal messages tend to be bunched (when a thread gets one audit message logged it is likely to get several around the same time), there is probably not a significantly higher cost to a relatively small SCANTIME, like 10 or even 5, than to a much larger one, like 60. On the other hand, the amount of extra work performed by *SirScan* associated with the extra SCANTIME second scan for each browse request will be roughly proportional to the size of SCANTIME. All this would suggest using a fairly small SCANTIME: a value of 10 might be a good starting point.

If you are setting SCANTIME for the first time in an Online, it is prudent to anticipate a 3% increase in journal usage, though the actual increase will almost undoubtedly be much less than that (probably less than even 1%).

Note: The heartbeat message can even be useful outside of *SirScan*. Since the heartbeat messages are regular RK type messages, they can be viewed with AUDIT204 as well as *SirScan*.

This means that if one is using AUDIT204 for a previous run's journal, and one wishes to find entries for a particular userid in a particular time interval, one can simply request from AUDIT204 all records for all users in the desired interval plus SCANTIME seconds before. By doing so, one can be certain that any messages formatted by AUDIT204 will be preceded by a message identifying the user associated with that message.

2.1.2   Using the SCANPARM system parameter

The second *SirScan* related User 0 parameter is SCANPARM. This parameter is a collection of bits that indicate certain *SirScan* behavior.

The SCANPARM bits are:

X'01'   Allow non-system-manager users to browse journal entries for public requests on *Janus Web Server* threads. Public requests are those that are not protected from access by userid.

X'02'   Issue a redundant "MSIR.0361 Processing request *method URL*" message after login. Normally, MSIR.0361 is issued on web threads before login, since the method and URL are known before the userid and password, if a login is even required for the request.

Unless the X'02' SCANPARM bit is set, users browsing journal entries for *Janus Web Server* threads based on userid do not see the MSIR.0361 message (since it occurs before the login), so they have a difficult time ascertaining the method and URL being processed in a particular request.

The default for the SCANPARM parameter is X'00'.

2.2   SIRSCAN subsystem definition

SirScan is a full-screen utility that allows users in a *Model 204* online to browse the contents of the journal. *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 displayed data may be filtered by date, time, IODEV, User (thread) number and UserID. In addition, any combination of specific journal record types may be requested.

SirScan provides the ability to view an ordinary journal dataset or any member of a ring journal, including the offload datasets if they are not on tape. Even the current unflushed contents of the journal buffer(s) are included in the display, without the user having to know what journal configuration is being used.

The SIRSCAN application subsystem (APSY) is distributed with a setup screen that allows the subsystem manager to limit the amount of disk I/O generated by users of *SirScan* from each of the 6 SCLASSes. This setup screen is part of SIRADMIN, and can only be altered by privileged users.

```
  --------------- * * * SirScan I/O and Record Caps * * * --------------------*
  ===>


          SCLASS               Max I/O              Max Records

      1. ADMIN_HI               1000                  110000

      2. ADMIN_MED              100                   10000

      3. ADMIN_LO               10                    1000

      4. USER_HI                1000                  100000

      5. USER_MED               100                   10000

      6. USER_LO                10                    1000

  --------------- * * * SirScan I/O and Record Caps * * * --------------------*
```

Updating the *SirScan* SCLASS maximums in SIRADMIN

Each of the 6 SCLASSes is assigned a limit to the physical I/O permitted for each journal scan, and a maximum number of records *SirScan* will format for the user. The I/O settings prevent excessive amounts of disk activity during journal scans, and the record maximum prevents excessive use of CCATEMP. If the subsystem is semi-public, the DEFAULT SCLASS is treated the same as USER_LO for these limits.

A user's *Model 204* privileges are set while in *SirScan* based on the user's SCLASS. If they include System Administrator privileges, the user can view any journal entries; otherwise the user can view entries only for his or her own logon session.

You can change the SCLASS privileges with the subsystem management facility. The default SCLASS definitions allow users in USER_LO, USER_MED, and USER_HI to view information only for their own logon session, and allow users in ADMIN_LO, ADMIN_MED, and ADMIN_HI to view journal entries for any user.

Scan Specification

Installation of *SirScan* is covered in the **UL/SPF Installation and Maintenance Guide**. *SirScan* should be installed as a private APSY subsystem in order to use the I/O limits assigned to the SCLASSes. To access the system, the user types "SIRSCAN" (or the name of the subsystem you have installed) on the *Model 204* command line. A scan specification screen is presented:

```
-------------------Journal Scan Criteria for: ULSPF400 -----01/08/05 16:10:02
==>                                                    SCLASS: ADMIN_HI

Start time (HH:MM:SS) ==> 10:23:00     Start date (YY/MM/DD) ==>
Interval (Minutes)    ==> 5          (Refresh to current time if blank).

User        ==> IODEV15,BOBBYJO,BILLYJO,BETTYJO,32,66
Line Width ==> 79                      Read extra SCANTIME seconds ==> Y

Maximum I/O's ==> 1000    Maximum records ==> 110000

Display - entry dates? (Y/N) ==> N        server numbers? (Y/N) ==> N
          entry times? (Y/N) ==> Y        user numbers? (Y/N)   ==> Y
          entry types? (Y/N) ==> Y

Format Entry types:   ST ==> Y   AA ==> Y   SirScan RK    ==> N

AD ==> N   CI ==> N   CP ==> N   CS ==> N   ER ==> N   LI ==> N   LP == > N
LS ==> N   MS ==> N   OI ==> N   OO ==> N   RK ==> N   US ==> N

-----------------------------------------------------------------------------
 1/Help                     3/Quit
```

The Scan Specification Screen.

This screen allows the user to specify the journal data to be retrieved and the format in which it should be displayed. Input fields are:

Start Time

Formatted **HH:MM:SS**.or **-MMMMMM**. The earliest audit trail entry to be formatted. The second syntax identifies the number of minutes to go back from the current time to begin formatting the journal. If Start Time is not specified, data is formatted from the start of the run or the oldest ring journal (if using ring journals) if the requesting user is a system manager (or is in one of the ADMIN SCLASSes). Otherwise data is formatted from the logon time of the requesting user.

Start Date

Formatted **YY/MM/DD**. The date of the earliest audit trail entry to be formatted. If this is not specified, it is determined based on the start time. If the start time is less than the current time the current date is used, otherwise yesterday's date is used.

Interval Valid formats are **MM:SS, HH:MM:SS** or **MMMMMMM**, where "H" is hours, "M" is minutes and "S" is seconds. If an interval is not specified, data is formatted up to the current time (or until I/O limits are hit). In addition, by leaving this time blank, *SirScan* runs in auto-refresh mode ("Auto-refresh mode" on page 20) so that the data being scanned is constantly refreshed to reflect any new audit trail data that was generated after the initial data was collected.

User Users to be included in the formatted output. This input field indicates which thread's/user's audit entries will be viewed. The selection criteria can be a set of blank or comma delimited "phrases," each made up of one or more "clauses" separated by the & symbol. Each clause can contain one of the following criteria:

IODEV*n* A number *n* indicating a specific IODEV type, as in IODEV15, IODEV7, or IODEV11.

PST Entries for all *Model 204* Psuedo-SubTasks.

n1.n2.n3.n4 An IP address for a Janus thread, as in 198.242.244.97 or 150.209.8.51. The IP address can also be followed by a slash (/) and a subnet mask, or by a hyphen (-) and a number of bits in a subnet mask, as in 198.242.244.0/255.255.255.0 or 198.242.244.0-24. These two subnetted IP addresses encompass the same set of IP addresses.

JAN:*sss* The name of a Janus port, possibly containing wildcards, as in JAN:WEBPORT, JAN:WEB*, or JAN:???PORT.

xxx A specific user number, as in 0, 233, or 1024.

xxx-yyy A range of user numbers, as in 0-20 or 111-1000.

ssss A string, possibly containing wildcards, that indicates a specific userid, as in RASPUTIN, RAS*, ???PUTIN. For users in the ADMIN_xxx SCLASSes, a userid of just an asterisk (*) is special-cased to mean not only all logged on users, but all threads, whether logged on or not.

Criteria can be mixed and matched using the & separator, which indicates an "AND" operation, or using blanks or commas, which indicate an "OR" operation. For example

```
IODEV15&LENIN 11-20
```

requests information for all IODEV 15 threads logged on as userid LENIN, and requests all the information for user numbers 11 through 20.

```
TROT*&198.242.244.33 JAN:SOCIALIST&MARX PST
```

requests information for all of the following:
- All connections from IP address 198.242.244.33 that log on a userid that begins with TROT
- All connections to Janus port SOCIALIST that log on to userid MARX
- All PSTs

Portnames and userids can contain special wildcard characters. These characters and their meanings are:

*   Matches any number of characters. For example, BRE* matches BREAD, BREEZY, and BREZHNEV.

?   Matches a single character. For example, ?RUSHCHEV matches TRUSHCHEV, BRUSHCHEV, and KRUSHCHEV.

"   Means the next character is to be treated literally, even if it is wildcard character. Using the double-quotation character is necessary if a wildcard character appears in the name to be matched. For example, E"*BARTER matches E*BARTER.

For users in USER_HI, USER_MED or USER_LO SCLASSes, no matter what selection criteria are specified, they will only be able to view audit entries associated with their own userid or, if the system SCANPARM 1 bit is set, entries for public logins on Janus web threads. So if a user in the USER_MED SCLASS specifies

```
IODEV15
```

for a selection criterion, the user will be able to see only IODEV 15 activity for her own userid or perhaps for public logins to a Janus Web thread (if the SCANPARM 1 bit is set).

Because the specified time interval may not include the journal entries that would allow *SirScan* to associate a thread's activity with a particular userid, IP address or port number is is possible that entries associated with a particular userid, IP address or port number will not be formatted. It is also possible that many entries in a time interval for a requested userid, IP address or port number will not be formatted but that those after an audit entry that allows determination of all these entities (a since-last statistic or a *SirScan* RK line), will be.

SirScan makes every effort to use all available information (current logged on userids and log times, M204.0352 messages, M204.0118 messages,

since-last stat entries, etc..) to ascertain this information about each audit trail entry but these attempts will necessarily be hit and miss. While most of the time, *SirScan* will pick up the desired information it is possible that information will also seem to be inexplicably missing. Often this information can still be retrieved by playing with the date/time interval.

Alternatively, if the SCANTIME system parameter is set, and the "Read extra SCANTIME seconds" switch is set to Y, *SirScan*s behavior could be made completely consistent and predictable at a (hopefully slight) cost.

If no criteria are specified for USER, only audit entries for the requesting user are displayed.

Note that one additional selection criterion is applied to any USER phrase if *SirScan* has not been purchased but instead is automatically authorized by *Limited Janus Web Server*. *Limited Janus Web Server* is a free, restricted version of *Janus Web Server*; they are both documented in the M204wiki (see
http://m204wiki.rocketsoftware.com/index.php/Janus_Web_Server). To assist in web development, sites that don't already have *SirScan* can use it for free if they are using *Limited Janus Web Server*. Sites using *SirScan* for free are automatically limited to viewing the journal activity of Janus Web threads and the TCP/IP subtask; any USER selection will thus be automatically restricted.

You may, of course, upgrade to a full *SirScan* at any time.

Output Line Width

The audit trail data can be formatted for any line width from one less than the screen width to 255. The minimum output line width is 131 for Model 5 terminals and 79 for all other terminal types.

Read extra SCANTIME seconds

This field only appears if the SCANTIME ("Using the SCANTIME system parameter" on page 6) system parameter is set to a non-zero value. The value of this field must be either "Y" or "N". Setting this field to "N" allows anomalous and confusing behavior on *SirScan*'s part so should be avoided unless the SCANTIME parameter is set, against recommendations, to a very high value, the user is very concerned about security and the user understands the anomalous behavior that is likely to result and either feels it doesn't apply to her selection criteria or doesn't care.

Maximum I/O's

The maximum number of full-track journal reads *SirScan* will perform when scanning the journal. The maximum value allowed for this field is determined by the user's subsystem SCLASS. This value should generally

be kept to a relatively small number such as 100 or 1000 in the case of very busy systems to avoid accidentally scanning too much of the journal when an inappropriate time interval is specified.

Maximum records

The maximum number of journal records to be formatted by *SirScan*. The maximum value allowed for this field is determined by the user's subsystem SCLASS. This value should generally be kept to a reasonably small number such as 10000 to avoid accidentally building an unmanageably large list of formatted records.

Display User Numbers

This indicates whether user numbers are to appear in the fomatted audit trail data.

Display Server Numbers

This indicates whether server numbers are to appear in the fomatted audit trail data.

Diplay Entry Dates

This indicates whether the date of each entry is to appear in the formatted audit trail data. The dates are displayed in YYMMDD format.

Display Entry Times

This indicates whether the time of each entry is to appear in the formatted audit trail data. The times are displayed in HHMMSSTH format.

Display Entry Types

This indicates each type of entry which is to be included in the formatted audit trail data. The dates are displayed in YYMMDD format.

ST   Responding "Y" to this prompt causes ST records (all types of statistics records) to be included in the formatted output. Specifying "N" excludes these records.

AA   Responding "Y" to this prompt causes all audit type records to be included in the formatted output. Specifying "N" causes *SirScan* to pay attention to the specific Y/N settings for each record type in the bottom two rows on the screen. The various record types are described in the next section, and also in the *Rocket Model 204 System Manager's Guide*.

SirScan RK

Responding "Y" to this prompt causes SirScan heartbeat RK messages for the SCANTIME system parameter ("Using the SCANTIME system parameter" on page 6) to be formatted. These messages largely exist to help *SirScan* identify threads by userid or other selection criteria and are not particularly interesting so are ordinarily suppressed regardless of the AA or RK switch settings. If this prompt is set to "Y" the *SirScan* heartbeat messages will be displayed.

Commands:

The following commands and PF keys are valid on the scan specification screen:

=x.y.z   Commands prefixed by "=" invoke fastpath navigation of the UL/SPF menu system. "=M" sends the user to the UL/SPF main menu if UL/SPF is active. "=X" exits to command level. "=M.4.5.2" sends the user to the "Active Subsystems" display in SIRMON (if SIRMON is active).

X   Exits to command level.

PF keys:

PF1   Accesses online help.

PF3   Quit (return to command level).

Browsing the journal

Output from the journal scan is placed in a special data structure, and the user is transferred to a browse session.

If *SirScan* is in auto refresh mode — that is, the user left **Interval** blank on the specification screen — *SirScan* places newly formatted journal entries into the data structure every time the user requests a forward scroll past the end of the current data.

```
    ------------ Scan ULSPF400 98/08/05 16:07:19 - *------------98/08/05 16:28:51
 ===>                                         Line: 256     Cols: 1   To 79
              VTBL=3 PDL=412 CPU=2 PCPU=284 RQTM=7
 16284350 20 ST USERID='ALAN' ACCOUNT='ALAN' LAST='EVAL' SUBSYSTEM='SUBSYSMGMT'
              PROC-FILE='M204PROC' PROC='SUMN.LOGIN' NTBL=1 GTBL=472 QTBL=5
              STBL=16 VTBL=3 PDL=412 RQTM=1
 16284351 20 MS M204.1168: IN FILE M204PROC INCLUDE SHRS.XFERLOG
 16284378 20 MS M204.1168: IN FILE M204PROC INCLUDE DICS.RELEASE
 16284386 20 MS M204.1168: IN FILE M204PROC INCLUDE SHRS.CHKSYSFILES
 16284413 20 ST USERID='ALAN' ACCOUNT='ALAN' LAST='CMPL' SUBSYSTEM='SUBSYSMGMT'
              PROC-FILE='M204PROC' PROC='SUMN.LOGIN' NTBL=83 QTBL=516
              STBL=2002 VTBL=105 PDL=1140 CPU=267 PCPU=428 RQTM=621 DKPR=110
 16284419 20 ST USERID='ALAN' ACCOUNT='ALAN' LAST='EVAL' SUBSYSTEM='SUBSYSMGMT'
              PROC-FILE='M204PROC' PROC='SUMN.LOGIN' NTBL=83 GTBL=563 QTBL=516
              STBL=2087 TTBL=8 VTBL=105 PDL=1140 CPU=33 FINDS=2 PCPU=550
              RQTM=60 DKPR=27
 16284421 20 MS M204.1168: IN FILE M204PROC INCLUDE SUMP.MAIN0000
 16284422 20 ST USERID='ALAN' ACCOUNT='ALAN' LAST='LOAD' SUBSYSTEM='SUBSYSMGMT'
              PROC-FILE='M204PROC' PROC='SUMP.MAIN0000' NTBL=294 QTBL=2611
              STBL=12679 VTBL=583 FSCB=3664 CPU=6 PCPU=1200 RQTM=5 DKPR=13
 -----------------------------------------------------***  Bottom of data ***
  1/Help      2/Scale on   3/Quit      4/PFkey off
  7/Up        8/Down       9/Repeat    10/Left      11/Right
```

Browsing the formatted journal output.

The body of this screen presents the journal information returned from the user's *SirScan* request. The following commands and PF keys are valid.

Commands

ALL searchstring

Limits the display to lines containing the string *searchstring*. Typing ALL with no searchstring returns the user to viewing the entire selection of journal entries as originally specified. To limit based on a string ending or beginning with blanks, use slashes to delimit the string, for example:

> ALL / *searchstring* /.

USER unum The USER command limits the display to only those lines associated with the indicated user number. This is very similar to specifying `ALL / unum /`, except that, with the USER command, lines not associated with the the user number that happen to have the number in them for some other reason are suppressed.

Also, continuation lines for entries associated with the indicated user are likely to be suppressed for the ALL command, but will not be for USER.

An effect similar to the USER command can be achieved by moving the cursor down to a line associated with the required user and pressing the Enter key. This is different from the USER command in that entries are limited to the login session associated with the entry over which the cursor was positioned.

This facility is dependent on M204.0352 messages appearing in the journal. If these are MSGCTL'ed to NOAUDIT, cursor-position based user selection will behave the same as USER command based selection.

PFKEYS ON | OFF PFKEYS allows the user to turn on or off the display of the messaage line and the two lines with PF key descriptions. If Pu3EYS OFF is specified, the 3 extra lines are used to display formatted journal data.

SCALE ON | OFF Turns on or off a two-line character scale display. If the scale is turned off the extra two lines are used to display formatted journal data.

RIGHT x Scrolls the display **x** characters to the right.

LEFT x Scrolls the display **x** characters to the left.

F xxxx or /xxxx Finds the string **xxxx** in the formatted audit trail data beginning at the current line.

-F or -/ Same as above, but performs the search backward form the current line.

M (with PF7, 8, 10 or 11)   Scrolls the display the maximum number of lines up or down, or the maximum number of characters left or right.

L xxx   Moves the current line to the requested line number xxx.

PF Keys:

`PF1`   Accesses full-screen help.

`PF2`   ON/OFF toggle for the horizontal scale.

`PF3`   Quit to *SirScan* specification screen.

`PF4`   ON/OFF toggle for the PF key display.

`PF5`   Scroll up an Interval (only active if an Interval was specified on the previous screen).

`PF6`   Scroll down an Interval (only active if an Interval was specified on the previous screen).

`PF7`   Scroll up.

`PF8`   Scroll down.

`PF9`   Repeat the last command.

`PF10`   Scroll left.

`PF11`   Scroll right.

When entries are displayed with dates, the dates are in YYMMDD format. When entries are displayed with times, the times are displayed in HHMMSS format.

For journal entry type selection, the following types of entries might be displayed (each type may be included or excluded from the display by switches on the previous screen):

Entry   Description

AD   General *Model 204* status messages.

CI   Physical input line from user terminal or IFAM argument.

CP   Physical input line from a procedure.

CS   A physical line from full screen input.

ER   A *Model 204* error message.

LI	A logical input line from user terminal or Host Language Interface argument.
LP	A logical input line from a procedure.
LS	One input value from a SCREEN or a menu selection number of a MENU.
MS	A *Model 204* information message.
OI	An input line from the operator's console.
OO	An output line to the operator's console.
RK	Information about system initialization or to record a call to a Host Language Interface function.
ST	Miscellaneous statistics.
US	Information audited on a User Language AUDIT statement.

4.1   Auto-refresh mode

When no time interval was specified on the scan specification screen *SirScan* browsing occurs in auto-refresh mode. This means that every time an attempt is made to go beyond the end of the current formatted records, *SirScan* formats a new interval of records from the last time it formatted records to the current time. Going beyond the end of current formatted records could occur because of an attempt to scroll beyond the current set of records, an attempt to position beyond the current set of records, or a search for a string that is not found in the current set of records. In addition, if any display is about to contain some empty lines, an auto-refresh is done to see if the lines can be filled in.

Auto-refresh mode is a convenient way of monitoring things as they happen. A typical thing to do with *SirScan* is to specify a start time of -Ø, which means start "now" with no time-interval size. The browse screen is then entered and the user could then initiate the action being monitored and, because auto-refresh automatically populates the screen with newly written journal records, see what's being logged to the journal without leaving the browse screen.

Ordinarily, auto-refresh mode is fairly efficient; however it should be kept in mind that if the system SCANTIME parameter is set and the "read extra SCANTIME seconds" field on the scan selection screen was set to "Y", then an extra SCANTIME seconds of journal will be read on every auto-refresh.

This is not to say that users should set the indicated parameter to "N" when in auto-refresh mode. Quite the contrary. Without that parameter set to "Y", the behavior of auto-refresh could be very confusing indeed. It should just be kept in mind that the use

of the SCANTIME data will perhaps add a little extra delay and overhead to auto-refresh processing, especially on very busy systems with heavy journal activity. Auto-refresh mode on such systems will tend to be somewhat sluggish anyway, so the addition of the SCANTIME data scan would probably still not produce dramatic delays, unless SCANTIME is set to an absurdly high value.

_____
APPENDIX A *Date Processing*

SirScan 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 *SirScan* is under a rental or trial agreement.

- To display the current date, as returned by the TIME SVC (if the *Sir2000 User Language Tools* is active, the date is obtained by temporarily switching the clock using the APPDATE INTERNAL command) as page headers in various end-user displays.

- To select and display *Model 204* timestamps from the journal.

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.

Index

Index