

---

# Sirius Mods Release Notes Version 6.0

**December, 2000**

---



Sirius Software, Inc.  
875 Massachusetts Avenue, Suite 21  
Cambridge, MA 02139

Telephone: (617) 876-6677

FAX: (617) 234-1200

E-mail: [support@sirius-software.com](mailto:support@sirius-software.com)

World Wide Web: <http://sirius-software.com>

August 5, 2010

© 2010 Sirius Software, Inc.

---

## *Proprietary Notices*

The following products:

- *Fast/Unload User Language Interface*
- *Janus Network Security*
- *Janus Sockets*
- *Janus Specialty Data Store*
- *Janus TCP/IP Base*
- *Janus Web Server*
- *Sirius Functions*
- *Sirius Performance Enhancements V2*
- *Sirius Mods*
- *SirFact*
- *SirMon*
- *SirPro*
- *SirSafe*
- *SirScan*

are proprietary products of Sirius Software, Inc.:

**Sirius Software, Inc.**  
**875 Massachusetts Avenue, Suite 21**  
**Cambridge, Massachusetts 02139**  
**USA**

**Model 204®** is a proprietary product of Computer Corporation of America, a wholly-owned subsidiary of Rocket Software, Inc., which owns the trademark:

**Rocket Software Corporate Office**  
**M204 Division**  
**275 Grove Street**  
**Suite 3-410**  
**Newton, Massachusetts 02466-2272**  
**USA**

**SoftSpy™** is a proprietary product of Information Technology Systems:

**Information Technology Systems**  
**95 Wells Avenue**  
**Newton, Massachusetts 02459-3216**  
**USA**

---

# *Contents*

<b>Proprietary Notices</b> . . . . .	<b>ii</b>
<b>Contents</b> . . . . .	<b>iii</b>
<b>Chapter 1: Introduction</b> . . . . .	<b>1</b>
<b>Chapter 2: Maintenance and Support</b> . . . . .	<b>3</b>
Store protect most of the Model 204 load module . . . . .	3
Support custom zap to disable expiration warnings . . . . .	3
<b>Chapter 3: Janus Sockets - New Product</b> . . . . .	<b>5</b>
<b>Chapter 4: All products</b> . . . . .	<b>7</b>
ERCNT - user error count . . . . .	7
SIRPRMPT - job name in Model 204 command prompt . . . . .	7
HTML/TEXT block . . . . .	7
<b>Chapter 5: Janus TCP/IP Base</b> . . . . .	<b>9</b>
JANUS LIMITS subcommand . . . . .	9
JANUS LOADXT and DISPXT commands . . . . .	9
JANUS STAT . . . . .	9
JANUS TSTAT[US] subcommand . . . . .	9
[NO]AUDTERM . . . . .	10
Multiple commands on JANUS DEFINE . . . . .	10
Multiple files/groups on JANUS DEFINE . . . . .	10
NEWSESCMD, NEWSESOPEN, NEWSESNOCLEARG and NEWSESGUESTOK parameters . . . . .	10
PRELOGINUSER parameter . . . . .	11
Threads visible before login . . . . .	11
TRACE . . . . .	11
TRACE bits . . . . .	12
XTAB parameter . . . . .	12
<b>Chapter 6: Janus Network Security</b> . . . . .	<b>13</b>
JANUS ADDCA, DELCA, DIS[PLAY]CA and STAT[US]CA subcommands . . . . .	13
JANUS SSLSTAT[US] subcommand . . . . .	13
RSA offload . . . . .	13
SSL client support . . . . .	13

SSL V3 support	14
SSLCLCERT and SSLCLCERTR parameters on JANUS DEFINE	14
SSLIBSIZE and SSLOBSIZE parameters on JANUS DEFINE	14
SSLMAXAGE parameter on JANUS DEFINE	14
SSLMAXCERTL parameter on JANUS DEFINE	15
SSLOPT parameter on JANUS DEFINE	15
SSLPROT parameter on JANUS DEFINE	15
<b>Chapter 7: Janus Web Server</b>	<b>17</b>
HTML and TEXT blocks	17
Janus Web Server legacy support improvements	18
JAN[US]DEB[UG] command	18
New exception handlers	19
New MSGCTL options	19
New port definition parameters	20
CLOSEIMMED and CLOSEREAD	20
FORMPOSTWAIT	20
FORMREDIRWAIT	20
MSGSEND and NOMSGSEND	20
NOSCREENREDIR	20
NOTRACEFIELD	21
WEBLOGCOOKIE	21
SQUAREB parameter	21
SESCookie parameter	21
SESCOOKIENSEC parameter	21
SESTIMEOUT parameter	22
SLOWCLOSE parameter	22
SSLSES parameter	22
VARIPADDR parameter	22
New JANUS WEB DEBUG rules	23
PORTOPEN, DEBPORTOPEN, NOPORTOPEN and NODEBPORTOPEN	23
PORTCMD, DEBPORTCMD, NOPORTCMD and NODEBPORTCMD	23
New JANUS WEB ON rule parameters	23
CLOSEIMMED, NOCLOSEIMMED, CLOSEREAD and NOCLOSEREAD	23
FORMPOSTWAIT	24
FORMREDIRWAIT	24
MSGSEND and NOMSGSEND	24
XTAB	24
New JANUS WEB SCREEN rule parameters	24
LEGJS and NOLEGJS	24
SCREENREDIR and NOSCREENREDIR	24
New \$functions	24
\$WEB function	24
\$WEB_CERT_INFO and \$WEB_CERT_LEVELS	25
\$WEB_END_SES	25

\$WEB_FORM_ACTION . . . . .	25
\$WEB_FORM_ACTION_LEN . . . . .	25
\$WEB_FORM_DONE . . . . .	25
\$WEB_PROTOCOL function . . . . .	26
\$WEB_SAVE_LIST and \$WEB_REST_LIST functions . . . . .	26
STOP FILE enhancement . . . . .	26
<b>Chapter 8:    Sirius Performance Enhancements V2 . . . . .</b>	<b>27</b>
Macro language . . . . .	27
MAXINCL parameter . . . . .	27
<b>Chapter 9:    SirFact . . . . .</b>	<b>29</b>
Comment-initialized globals . . . . .	29
Open records dumped in SirFact dumps . . . . .	29
Found sets and LIST's dumped in SirFact dumps . . . . .	29
SIRFACT DUMP command enhancements . . . . .	29
SIRFACT SNAP command . . . . .	30
<b>Chapter 10:   SirMon . . . . .</b>	<b>31</b>
New External Call Facility stats . . . . .	31
New MQ Series stats . . . . .	31
<b>Chapter 11:   SirPro . . . . .</b>	<b>33</b>
Model 204 editor improvements (SIREEDIT parameter) . . . . .	33
<b>Chapter 12:   SirSafe . . . . .</b>	<b>35</b>
Support for read-only files under MVS . . . . .	35
Reduction in "noise" logging . . . . .	36
<b>Chapter 13:   SirScan . . . . .</b>	<b>37</b>
Combined selection criteria . . . . .	37
SCANTIME system parameter . . . . .	37
Improved userid support . . . . .	37
New selection criteria . . . . .	38
<b>Chapter 14:   Sirius Functions . . . . .</b>	<b>39</b>
\$BIND_LIST . . . . .	39
\$C2D . . . . .	39
\$D2C . . . . .	39
\$FIELD_LIST . . . . .	39
\$IMGINF . . . . .	39

\$IMGOVL	40
\$LISTADJ	40
\$LIST_CAPTURE	40
\$LISTCHK	40
\$LISTFINDI	40
\$LISTFINDI_SUB	40
\$LISTIMG_COPY	41
\$LISTOVLI	41
\$LISTREP	41
\$LISTREPI	41
\$RESETN	41
\$PROCOPN of temp request	41
<b>Chapter 15: Compatibility/Bug fixes</b>	<b>43</b>
Sirius Functions	43
Image association saved for \$list derivation \$functions	43
Janus TCP/IP Base compatibility issues	43
DEB[UG] keyword replaced by TRACE	43
[NO]AUDTERM	44
Janus Network Security compatibility issues	44
SSL buffer sizes	44
Janus Web Server compatibility issues	44
Legacy support now uses JavaScript	44
Legacy support now uses REDIRECT's	45
\$WEB_SEL family now returns null	45
Fixes in <b>Sirius Mods</b> 6.0 but not in 5.5 or 5.6	45
Version co-requisites	46

---

**CHAPTER 1** ***Introduction***

This document lists the enhancements and other changes contained in the newest release of *Sirius Mods*: version 6.0. The previous generally released version of *Sirius Mods*, 5.5, was released in October, 1999. An intermediate version, 5.6, was released to some customers in May, 2000.

Because 5.6 was an intermediate version, all customers running that version are expected to upgrade to *Sirius Mods* 6.0 as soon as possible. Also, because *Sirius Mods* 5.6 was not a general release, these release notes describe all enhancements to the *Sirius Mods* including those that were available in 5.6.



---

**CHAPTER 2** ***Maintenance and Support*****2.1 Store protect most of the Model 204 load module**

Most of the CSECT's in the *Model 204* load module are now store protected to prevent accidental overlays of the *Model 204* nucleus and so improve *Model 204*'s reliability. If this should cause a problem, this storage protection can be turned off by clearing the X'01' bit of the new MODPROT user zero parameter. This storage protection does not, by default, protect FUNU. If FUNU is reentrant, it can also be protected by setting the X'02' bit of the MODPROT user zero parameter.

**2.2 Support custom zap to disable expiration warnings**

If you are using a Sirius product which expires within a month, normally you receive operator MSIR.0427 expiration warnings each time a *Model 204* online (or BATCH204) job starts. You can use a custom zap to suppress the “current round” of these messages. This capability was previously introduced by ZAP54E0.



*Janus Sockets*, a new member of the Janus family of products, is available in version 6.0 of the *Sirius Mods*. This provides the basic connection over the Internet between *Model 204* and a very wide range of other Internet applications.

The JANUS DEFINE command is enhanced to provide two new port types:

**CLSOCK** This allows you to define a name that is used by User Language applications to create an “active” connection to a service running on some Internet host.

**SRVSOCK** This allows you to associate a port number with a processing command (CMD parameter). When an application running on some Internet host requests a connection to that port number on the *Model 204* Internet host, that command is invoked, with the connection established.

Either type of connection, CLSOCK or SRVSOCK, allows your User Language program to use the connection for sending and receiving data to the application running on the other Internet host. The interface is easy to use, with a flexible and convenient approach for establishing, managing, and ending the connection, and for dealing with errors which may occur.

A system manager provides the JANUS commands that describe the *Janus Sockets* connections that can be used; the connections are established and used in User Language programs with a set of \$functions. Some noteworthy features are:

#### **Translation**

Translation between the EBCDIC character set and the character set expected by the other Internet host is handled without explicit manipulation by the User Language programmer. You can specify either translated (CHAR) and non-translated (BINARY) operations, and when CHAR is in effect, you can use the new Janus XTAB capabilities to specify the translation tables, if you need anything other than standard US ASCII.

#### **Length- or separator-based receives**

There are two receive \$functions. One specifies the specific number of bytes to receive, and the other specifies a set of separator strings that end the next string to receive. This makes it easy to operate with various protocols that are well defined for Internet services, or to create your own protocols.

### **Sending strings or lines**

You can optionally follow a transmitted string with a LINEND string (specified by its ASCII hex code), again making it easy to operate with various protocols.

### **Captured print**

The flexibility and convenience of various print output in *Model 204*, including the new Sirius HTML statement, can be used to send lines over a connection (see “HTML and TEXT blocks” on page 17).

### **Certificate support**

You can use digital certificates to confirm the identity of the remote host on a *Janus Sockets* connection, and you can provide a digital certificate to a remote host, validating the identity of the *Model 204* online. Certificate support uses the features of *Janus Network Security*.

### **Controlled access**

With *Janus Sockets*, you can specify which local users and/or applications can access which remote Internet hosts and/or port numbers.

You can control which remote hosts (and/or users, which can be authenticated by a remote host providing a client certificate) use the local SRVSOCK services you provide to the Internet. You can also protect resources made available on a SRVSOCK port by the NEWSOCSMD facility, which can perform a *Model 204* LOGIN operation.

### **SSL encryption**

You can use encrypted communication with *Janus Sockets*, using the features of *Janus Network Security*.

### **Error handling**

You can establish a label in your User Language program to receive control when an error occurs, or you can test for error return codes from each \$function invocation, or you can allow *Janus Sockets* to simply cancel the User Language request when a socket error condition occurs.

### **Socket manipulation**

You can change a wide number of settings that apply to a connection, and you can obtain the current values of the settings.

### **Sirius-supplied applications**

Since many of the Internet protocols are well-defined, Sirius Software will provide User Language source that you can use in your system. For example, a User Language subroutine is available which you can call to send an e-mail message directly to the Internet.

---

**CHAPTER 4** *All products***4.1 ERCNT - user error count**

ERCNT is a new resetable user parameter. It is the user's current number of counting errors.

**4.2 SIRPRMPT - job name in Model 204 command prompt**

SIRPRMPT is a new user zero parameter. If the X'01' bit is set, the job name is displayed on the *Model 204* command prompt. This parameter was implemented in previous releases as ZAP5560, ZAP54C9 and ZAP5297.

**4.3 HTML/TEXT block**

The HTML (or TEXT) block in User Language is a general purpose alternative to a series of PRINT statements for producing a fair amount of literal data. It was introduced primarily for use with *Janus Web Server* (hence the name "HTML") but is available with many Sirius products; see ["HTML and TEXT blocks" on page 17](#). It is most useful when PRINT output is captured or directed by some means.



## 5.1 JANUS LIMITS subcommand

The JANUS LIMITS subcommand provides information about overall Janus thread usage. This information can be useful in trying to determine whether there is a danger of exceeding the licensed thread limit or the number of defined SDAEMON threads.

## 5.2 JANUS LOADXT and DISPXT commands

The JANUS LOADXT subcommand loads a pair of EBCDIC to ASCII and ASCII to EBCDIC translation tables that can be used for *Janus Web Server*, *Janus Sockets Janus Open Server* and *Janus Specialty Data Store* connections. The JANUS LOADXT command can load its translation tables from *Model 204* procedures or from members of an OS PDS. The format of the JANUS LOADXT command is

```
JANUS LOADXT name [[FILE|GROUP] name proc] | [DDNAME name]
```

The JANUS DISPXT displays either a list of all loaded translate tables or the contents of a specific translate table.

## 5.3 JANUS STAT

The JANUS STAT subcommand is now a synonym for JANUS STATUS.

## 5.4 JANUS TSTAT[US] subcommand

The JANUS TSTAT subcommand provides information about thread usage by port. This information can be useful in trying to determine what is causing thread usage.

## 5.5 [NO]AUDTERM

The NOAUDTERM and AUDTERM parameters of JANUS DEFINE now apply to the SDS, OPENSERV, and SRVSOCK port types, in addition to WEB ports (as previously). Also, the default has been changed from AUDTERM to NOAUDTERM, and even with NOAUDTERM in effect, compiler messages are sent to the audit trail. See [“Janus TCP/IP Base compatibility issues”](#) on page 43.

## 5.6 Multiple commands on JANUS DEFINE

The CMD parameter on the JANUS DEFINE command now allows multiple commands to be specified. The commands should be separated by the AND keyword as in

```
JANUS DEFINE XX 80 WEBSERV 5 CMD -  
    'R MCPU 100' AND -  
    'UTABLE LGTBL 1500'
```

## 5.7 Multiple files/groups on JANUS DEFINE

The OPEN parameter on the JANUS DEFINE command now allows multiple files or groups to be specified and each file or group can have the file open privileges specified. The files or groups should be separated by the AND keyword and privileges should be specified as hexadecimal strings as in

```
JANUS DEFINE XX 80 WEBSERV 5 OPEN -  
    FILE WEBPROC -  
    AND GROUP WEBCACHE X'BFFF' -  
    AND WEBINFO X'0761'
```

## 5.8 NEWSESCMD, NEWSESOPEN, NEWSESNOCLARG and NEWSESGUESTOK parameters

The indicated parameters are now available on the JANUS DEFINE command for WEBSERV and SRVSOCK ports. The NEWSESCMD indicates the command or commands (separated by AND) that should be issued when a new session is established.

With SRVSOCK requests, every new connection is a new session but *Janus Web Server* has the ability to maintain logical sessions either using SSL or session cookies so that not every connection will cause the NEWSESCMD commands to be issued. The NEWSESCMD might be able to issue a \$SIR\_LOGIN to cause a login for a particular

user based on information the NEWSESCMD deems appropriate such as an SSL client certificate or a userid and password sent on a form or a cookie or whatever.

NEWSEOPEN indicates the files and/or groups to be opened before issuing the NEWSESCMD.

NEWSESNOCLEARG indicates that a CLEARG is not to be done after the NEWSESCMD commands are done. This can be useful if there is a strong need to communicate information from the NEWSESCMD commands to the rest of the *Janus Web Server* or *Janus Sockets* applications, but should be used carefully in light of the slight risk this entails: careless coding of a NEWSESCMD program could leave information in GTBL which should not “escape” NEWSESCMD, for example, passwords in GTBL which might even somehow be accidentally sent to the client.

NEWSESGUESTOK indicates that if the NEWSESCMD command issues a \$SIR\_LOGIN for a guest login, that is a login for a userid not defined to the Online or the external authorizer, it will be allowed albeit with low privileges.

## **5.9 PRELOGINUSER parameter**

This JANUS DEFINE parameter indicates the userid to be used for SDAEMON threads allocated for use by *Janus Web Server*, *Janus Sockets*, *Janus Specialty Data Store* and *Janus Open Server* applications but not yet logged in. This parameter defaults to “NO USERID”.

## **5.10 Threads visible before login**

For certain types of Janus requests, there is quite a bit of activity on a thread before a user is logged on. While this is happening, such a thread could be consuming resources or, perhaps worse, tying up a thread while waiting for a client thread. With *Sirius Mods 6.0* these threads will now be visible to a MONITOR command or *SirMon* and can be BUMP'ed. By default, the userid used for a non-logged-in but visible thread will be “NO USERID” but can be overridden by specifying the PRELOGINUSER parameter on the JANUS DEFINE command for a port.

## **5.11 TRACE**

The TRACE keyword has been introduced, replacing the previous uses of the DEBUG keyword. See [“Janus TCP/IP Base compatibility issues” on page 43](#).

## **5.12 TRACE bits**

Several new bits have been added to the trace flag specified with a JANUS TRACE command. Also, the meaning of the X'08' bit on non-SSL ports has been changed so that setting the X'08' bit will cause a trace of the unencrypted data being sent over a connection. The new TRACE bits are:

- X'20'** Trace web response headers.
- X'40'** Trace web response data; generally HTML or plain text though could be anything.
- X'80'** Traces the actual data being sent over an SSL connection, that is the SSL handshake data and the encrypted data that follows.

## **5.13 XTAB parameter**

This JANUS DEFINE parameter indicates the EBCDIC to ASCII, ASCII to EBCDIC and character entity translation tables to be used for the port. The translate table must have been previously loaded with the JANUS LOADXT command.

## **6.1 JANUS ADDCA, DELCA, DIS[PLAY]CA and STAT[US]CA subcommands**

The JANUS ADDCA subcommand adds a certifying authority's certificate to a port's list of trusted certifying authorities. A Janus client using SSL or a Janus server requesting a client certificate (SSLCLCERT or SSLCLCERTR) will only accept certificates signed by a trusted certifying authority. Certifying authority certificates for *Janus Network Security* must be stored in base64 encoded form in a *Model 204* procedure. JANUS DELCA deletes a certifying authority's certificate from the list of trusted authorities for a port. JANUS DISPLAYCA and JANUS STATCA display the contents and status of certifying authority certificates, respectively.

## **6.2 JANUS SSLSTAT[US] subcommand**

The JANUS SSLSTAT subcommand provides information about the versions of SSL that are being used to access a Janus server and the efficacy of logical session caching.

## **6.3 RSA offload**

RSA private-key decryption is now performed in a subtask, either an MP subtask for MP/204 customers or in a special RSA decryption subtask if the new ACTRDS and MAXRDS parameters are greater than 0. This is useful for taking advantage of idle processors to perform the CPU intensive RSA private-key decryption operation.

## **6.4 SSL client support**

*Janus Network Security* now provides support for client SSL connections. This can be used for *Janus Open Client* and *Janus Sockets* client applications. Because SSL clients do not necessarily have to present a certificate to the server, SSL ports that will be serving clients can be started without a certificate or private key. This is indicated by specifying 0 after the SSL keyword on the JANUS DEFINE. command.

## **6.5 SSL V3 support**

*Janus Network Security* now supports SSL V3. The important benefits of SSL V3 over SSL V2 are its greater security and its longer logical session times (24 hours vs. 2 minutes) which can dramatically decrease *Janus Network Security* overhead because an expensive private-key decryption must be performed by *Janus Network Security* whenever a new logical session is started.

## **6.6 SSLCLCERT and SSLCLCERTR parameters on JANUS DEFINE**

The SSLCLCERT parameter on the JANUS DEFINE command for SSL server ports indicates that a client certificate should be requested. The client might or might not present a certificate.

The SSLCLCERTR parameter indicates that a client certificate is required, that is one should be requested and if not presented by the client the connection should be closed unless running a *Janus Web Server* thread with a SSLNOCERTERR exception handler in which case that exception handler would be run.

## **6.7 SSLIBSIZE and SSLOBSIZE parameters on JANUS DEFINE**

The SSLIBSIZE and SSLOBSIZE parameters for a JANUS DEFINE of an SSL port is the size of the input and output buffers respectively. In previous releases only the SSLBSIZE parameter was available and this set only the input buffer size, the output buffer size always being a paltry 256 bytes. With *Sirius Mods* 6.0 SSLBSIZE sets both the input and output buffer sizes, the default output buffer size is 4096 bytes and the input and output buffer sizes can be set independently with the SSLIBSIZE and SSLOBSIZE parameters.

## **6.8 SSLMAXAGE parameter on JANUS DEFINE**

The SSLMAXAGE parameter for a JANUS DEFINE of an SSL port is the maximum number of minutes an SSL logical session is to be held. When a session exceeds SSLMAXAGE, a new key-exchange must be performed. The default SSLMAXAGE is 1440 minutes which is the standard maximum session length for SSL. SSLMAXAGE can be set to a smaller value and perhaps gain a marginal improvement in security.

## **6.9 SSLMAXCERTL parameter on JANUS DEFINE**

The SSLMAXCERTL parameter for a JANUS DEFINE of an SSL port is the amount of space to be allocated for each possible SSL session to hold a certificate received from a client for a Janus server or from a server for a Janus client. The default SSLMAXCERTL of 1024 should be sufficient for most purposes. However, if extremely long certificates are being presented to Janus a larger SSLMAXCERTL can be specified or, if virtual storage usage is a severe constraint a smaller value can be specified though this is not really recommended.

## **6.10 SSLOPT parameter on JANUS DEFINE**

Only available on CLSOCK ports this parameter indicates that the *Janus Sockets* client application might or might not use SSL, namely that SSL is optional. The use of SSL on such a port would be indicated by the SSL parameter on the \$SOCK\_CONN call.

## **6.11 SSLPROT parameter on JANUS DEFINE**

The SSLPROT parameter for a JANUS DEFINE of an SSL port is a bitmask that indicates which protocol is to be allowed for SSL connections. The two bits currently supported are 1 for SSL V2 and 2 for SSL V3. The most likely use of this parameter is to limit SSL connections to SSL V3.



---

**CHAPTER 7** *Janus Web Server*

## 7.1 HTML and TEXT blocks

HTML (and plain text) to be sent to a browser no longer needs to be placed inside quotes in a PRINT statement or sent via \$WEB\_PROCSSEND. Instead large blocks of mostly static HTML can be placed inside an HTML/END HTML or TEXT/END TEXT block without quotes. Any *Model 204* expression including simply %variable values or \$function invocations can be placed inside an HTML block as long as it is enclosed inside the expression start and end characters which default to curly brackets (“{” and “}”). The following is an example of the use of the HTML statement:

```
HTML
<form method="POST" action="processform">
<br>
Last name: <input type="text"
           name="lname" value="{%LNAME}"/>
</form>
END HTML
```

The HTML (or TEXT) block can be used in contexts other than the production of a Web page; it is a general purpose alternative to a series of PRINT statements for producing a fair amount of literal data. In addition to producing a Web page, this can be used:

- to store lines in a temporary procedure or sequential dataset, using the USE command
- to populate a \$list, using the \$LIST\_CAPTURE function
- to send lines over a *Janus Sockets* connection, using the \$SOCK\_CAPTURE function
- to prepare output processed by some other program, when the output is created by something like a BATCH2 or RCL connection to *Model 204*

The HTML block is available to any customer who owns any of the following products:

- *Fast/Unload User Language Interface*
- *Janus Sockets*
- *Janus Web Server*
- *Sirius Functions*

## 7.2 Janus Web Server legacy support improvements

*Janus Web Server* legacy support (the automatic conversion of 3270 screens to HTML) has been enhanced in several ways.

It now uses REDIRECT's by default so that all legacy screens are the results of GET's to the same URL which means that browsers will only have a single history entry for a legacy session. This behavior can be overridden with the NOSCREENREDIR parameter on the JANUS DEFINE command, JANUS WEB SCREEN rules or with \$WEB\_SCREEN.

*Janus Web Server* legacy support by default now sends JavaScript that provides initial cursor positioning upon output and subsequent cursor sensing upon input (it senses the last input field the cursor was on) and allows the **Enter key** to cause form submittal no matter which browser is being used and no matter how many input fields are present on the form. This JavaScript can be suppressed with the NOLEGJS parameter on the JANUS DEFINE command, JANUS WEB SCREEN rules or with \$WEB\_SCREEN.

## 7.3 JAN[US]DEB[UG] command

The JANUSDEBUG command can be used to run web requests on a non-SDAEMON thread, most probably a 3270 thread doing interactive debugging. JANUSDEBUG can be abbreviated to JANUSDEB, JANDEBUG, or JANDEB. A thread running JANUSDEBUG can run TEST SUBSYSTEM, TEST DEBUG, SSTEEST, SoftSpy or any other testing subsystem. JANUSDEBUG will only allow debugging of requests running under the same userid as the debugging user or public (no login required) requests that are being debugged as a result of JANUS WEB DEBUG rules that had the PUBLIC keyword. The JANUSDEBUG command takes a few optional arguments:

- IPADDR ipaddr** Indicates the IP address or addresses for which requests are to be debugged. The IP address can be a single IP address or can be an IP address followed by a subnet mask or a number of subnet bits.
- PORT portname** Indicates the ports for which requests are to be debugged. Can contain wildcard characters.
- URL url** Indicates the URL's to be debugged. Can contain wildcard characters.

## 7.4 New exception handlers

The following new exception handling conditions can be specified in JANUS WEB ON, TYPE and REDIRECT rules:

<b>NOLOGCOOKIE</b>	Invoked when a port has the WEBLOGCOOKIE parameter set but a browser does not provide the login cookie for a public URL.
<b>NOUSERID</b>	Invoked when a user has requested a URL that requires a login but the user has not specified a userid yet. This condition was implemented as a ZAP in <i>Sirius Mods 5.5</i> in ZAP5544.
<b>OPENERR</b>	Invoked when there is an error opening a file or group in the JANUS DEFINE command for the port or the JANUS WEB ON rule for the requested URL.
<b>SSLNOCERTERR</b>	Invoked when a user tries to connect to an SSL port defined with the SSLCERTR parameter but does not present a client certificate. This can be useful in presenting a user with the information that a client certificate is required and how such a certificate might be obtained.
<b>SSLPROTOCOLERR</b>	Invoked when a user tries to connect using an SSL protocol version that is not being allowed. Generally this would be used to respond to attempts to connect to <i>Janus Web Server</i> using SSL V2 when only SSL V3 is being allowed.
<b>UNAUTHORIZED</b>	Invoked when a user has logged in with a valid userid and password but is not authorized to access the requested URL. This condition was implemented as a ZAP in <i>Sirius Mods 5.5</i> in ZAP5544.

## 7.5 New MSGCTL options

The following new options are available on the MSGCTL command when the 1 bit of the WEBAUDIT system parameter is set.

<b>AUDITW</b>	Only audit message for web threads.
<b>NOTERMW</b>	Only send message to terminal for non-web threads.
<b>TERMW</b>	Only send message to terminal for web threads.

These options were actually made available in *Sirius Mods 5.5* as ZAP55A3.

## **7.6 New port definition parameters**

### **7.6.1 CLOSEIMMED and CLOSEREAD**

CLOSEREAD indicates that web threads should “put up” a TCP/IP read after receiving the entire request but before processing it. This read can then detect if a connection has been lost. If CLOSEREAD is not specified a lost connection will probably only be detected when the web thread actually tries to write data to the connection.

CLOSEIMMED indicates that if a lost connection is noticed by the Janus PST, the PST should BUMP the *Model 204* thread associated with the connection immediately. If CLOSEIMMED is not specified a lost connection will only cause request termination when the web thread tries to send data to the connection or invokes a \$WEB function.

CLOSEREAD and CLOSEIMMED are useful for ensuring that CPU and I/O intensive web requests will be terminated immediately should the user become impatient and cancel the request.

### **7.6.2 FORMPOSTWAIT**

Indicates the default number of seconds to wait for the POST'ing of a form while using \$WEB\_FORM\_DONE. This effectively sets a session timeout for persistent *Janus Web Server* applications using \$WEB\_FORM\_DONE.

### **7.6.3 FORMREDIRWAIT**

Indicates the default number of seconds to wait between a REDIRECT and receipt of the GET request associated with the REDIRECT while using \$WEB\_FORM\_DONE or the *Janus Web Server* legacy support.

### **7.6.4 MSGSEND and NOMSGSEND**

Indicates whether or not *Model 204* messages (as opposed to the output from PRINT or HTML statements) are to be sent to the browser. For backward compatibility, the default for this parameter is MSGSEND though NOMSGSEND is probably more useful for most applications.

### **7.6.5 NOSCREENREDIR**

Indicates that *Janus Web Server* legacy support (automatic mapping of 3270 applications to HTML) is not to use the REDIRECT's to map all screens for a legacy session to the same URL. This means that all legacy screens will be maintained as a separate browser cache entry that is a result of a POST to a generated URL. Use of this parameter is discouraged but is provided if for some reason full backward compatibility with previous versions of *Janus Web Server* legacy support is desired.

### **7.6.6 NOTRACEFIELD**

Indicates the form field or fields whose values are not to be traced when the X'04' bit (causes tracing of form field values for web requests) of JANUS TRACE (or DEBUG) is in effect. This is useful for preventing passwords in form fields from being logged to the audit trail. Multiple form fields names can be specified after NOTRACEFIELD by separating them with an "AND" as in

```
JANUS DEFINE MYWEB 80 WEBSERV 28 -  
        NOTRACEFIELD PASSWORD AND NEWPASSWORD
```

When form field values are being traced because of the X'04' TRACE bit, the values for fields specified in the NOTRACEFIELD clause always appear as asterisk in the trace.

### **7.6.7 WEBLOGCOOKIE**

Indicates the name of a cookie that contains a userid to be used for public (non-login protected) URL's. This can be useful for tracking activity for public URL's.

### **7.6.8 SQUAREB parameter**

The SQUAREB parameter on the JANUS DEFINE command can be used to override the default EBCDIC characters to be used for square brackets in the *Janus Web Server* log file. This can be useful if the *Janus Web Server* defaults (X'AD/X'BD' under CMS and X'BA/X'BB' under MVS) are not appropriate for the file transfer program being used to transfer the log to an ASCII based system.

### **7.6.9 SESCOOKIE parameter**

Indicates the name of a cookie that will be used to maintain logical sessions that will be used to perform trusted logins for users once they have performed a standard login either through standard HTTP challenge-response mechanisms or through a \$SIR\_LOGIN issued during NEWSSESCMD processing. This feature reduces the overhead of NEWSSESCMD processing by limiting it to logical session establishment time and eliminates some of the idiosyncrasies of HTTP challenge-response authentication associated with password changes whether end-user or system initiated. Note that SESCOOKIE's are cryptographically designed to be virtually impossible to "fake".

### **7.6.10 SESCOOKIEINSEC parameter**

Indicates that the cookie specified by SESCOOKIE will not be sent to the browser as a "secure" cookie. This would make it possible with some browsers in certain situations to use the SESCOOKIE cookie to have a logical session operate over multiple *Janus Web Server* ports.

### **7.6.11 SESTIMEOUT parameter**

Indicates the number of minutes of inactivity that will cause a logical login session being held either via SESCOOKIE or SSLSES to be terminated. When such a session is terminated, *Janus Web Server* will go through normal login validation which might or might not force the end-user to re-enter a password depending on the validation technique and possibly the browser.

### **7.6.12 SLOWCLOSE parameter**

A bitmask that indicates the conditions under which *Janus Web Server* is to perform a “slow” close of a connection with a browser. This parameter is there to deal with a bug in some browsers that sends an extra junk carriage-return and line-feed character after the contents of an HTTP POST. If these two extra characters are sent by the browser and *Janus Web Server* does not read them from TCP/IP (most of the time it will) then under certain conditions TCP/IP will discard outgoing data from *Janus Web Server* when a normal close is done on the socket. To prevent losing this data in this case *Janus Web Server* must perform a more complex and so “slower” and, not incidentally, more expensive close for the connection. This parameter defaults to X'01' which means that *Janus Web Server* will only perform “slow” close processing for a POST and only when it has not already received an extra carriage-return and line-feed. While this default covers the known browser bug the parameter can be set to other values to request “slow” closes for GET's (X'02') or PUT's (X'04') or POST's, GET's or PUT's (X'10', X'20' and X'40', respectively) even when an extra carriage-return and line-feed have been received. This should only be necessary should browsers that send junk characters under other conditions be discovered.

### **7.6.13 SSLSES parameter**

Indicates that SSL logical sessions will be used to perform trusted logins for users once they have performed a standard login either through standard HTTP challenge-response mechanisms or through a \$SIR\_LOGIN issued during NEWSESCMD processing. This feature reduces the overhead of NEWSESCMD processing by limiting it to logical session establishment time and eliminates some of the idiosyncrasies of HTTP challenge-response authentication associated with password changes whether end-user or system initiated.

### **7.6.14 VARIPADDR parameter**

Indicates that a single browser might access *Janus Web Server* using different IP addresses (probably because it's accessing *Janus Web Server* through a “proxy farm”) so that *Janus Web Server* should not use IP address to decide whether a set of requests all came from the same browser or not. Setting VARIPADDR reduces *Janus Web Server's* ability to distinguish requests from multiple browsers potentially causing problems with legacy sessions, persistent sessions or slightly reducing the efficacy of

session cookies. Nevertheless, VARIPADDR is more or less unavoidable if users are accessing *Janus Web Server* through “proxy farms”.

## **7.7 New JANUS WEB DEBUG rules**

New JANUS WEB DEBUG rules indicate which URL's are to be run on non-SDAEMON threads, most likely for interactive debugging. JANUS WEB DEBUG rules, in addition to a URL which can contain wildcards can also have the IPADDR or IPGROUP and the USER or USGROUP keywords to indicate which IP addresses and userids are to have their requests run interactively. In addition, a PUBLIC keyword indicates that a URL that does not require user login will also be run interactively. There are also JANUS WEB NODEBUG rules to turn off debugging for specific URL's. A request for a URL that matches a JANUS WEB DEBUG rule will cause the request for the URL to wait for a JANSDEBUG command to be issued on another thread for the URL. The issuer of the JANUSDEBUG command would then have the request run on her thread allowing that user to, among other things, run an interactive debugger.

Other keywords on JANUS WEB DEBUG rules are:

### **7.7.1 PORTOPEN, DEBPORTOPEN, NOPORTOPEN and NODEBPORTOPEN**

Override the explicitly specified or default DEBPORTOPEN parameter on the JANUS DEFINE for the port. This indicates whether or not the files or groups specified on the OPEN clause of the JANUS DEFINE command are to be opened on a debugging thread.

### **7.7.2 PORTCMD, DEBPORTCMD, NOPORTCMD and NODEBPORTCMD**

Override the explicitly specified or default DEBPORTCMD parameter on the JANUS DEFINE for the port. This indicates whether or not the commands specified on the CMD clause of the JANUS DEFINE command are to be issued on a debugging thread.

## **7.8 New JANUS WEB ON rule parameters**

### **7.8.1 CLOSEIMMED, NOCLOSEIMMED, CLOSEREAD and NOCLOSEREAD**

Override the default or explicitly specified CLOSEIMMED and CLOSEREAD parameters on the port definition. These can also be overridden in a \$WEB\_SET call.

### **7.8.2 FORMPOSTWAIT**

Override the default or explicitly specified FORMPOSTWAIT parameters on the port definition.

### **7.8.3 FORMREDIRWAIT**

Override the default or explicitly specified FORMREDIRWAIT parameter on the port definition.

### **7.8.4 MSGSEND and NOMSGSEND**

Override the default or explicitly specified MSGSEND parameter on the port definition.

### **7.8.5 XTAB**

Override the default or explicitly specified XTAB parameter on the port definition.

## **7.9 New JANUS WEB SCREEN rule parameters**

### **7.9.1 LEGJS and NOLEGJS**

Override the default or explicitly specified LEGJS parameter on the port definition. These can also be overridden in a \$WEB\_SCREEN call.

### **7.9.2 SCREENREDIR and NOSCREENREDIR**

Override the default or explicitly specified SCREENREDIR parameter on the port definition. These can also be overridden in a \$WEB\_SCREEN call.

## **7.10 New \$functions**

The following \$functions are now available:

### **7.10.1 \$WEB function**

Returns a 1 if running a *Janus Web Server* request and a 0 otherwise. This functions provides a simple, convenient test that can be placed inside code that might run in web and non-web requests. It is better than a test of \$WEB\_PORT because it will work

correctly even after the connection has been lost say as the result of a \$WEB\_DONE and it is better than a \$VIEW of 'IODEV' because SDAEMON's can also run non-web requests and a test of IODEV will not work for web requests running under JANUSDEBUG. Finally, it is just semantically neater to have an "IF \$WEB" in code instead of some more complex test.

### **7.10.2 \$WEB\_CERT\_INFO and \$WEB\_CERT\_LEVELS**

These functions allow retrieval of data from a client certificate sent by the browser and can be used in NEWSESCMD processing to do client-certificate based authentication. A port must be running with the SSL and SSLCLCERT or SSLCLCERTR parameters for these functions to return anything useful.

### **7.10.3 \$WEB\_END\_SES**

Terminates the logical session created using SSLSES or SESCOOKIE and can be useful in providing a user with a "logout" button or URL and can also be useful in allowing a user to switch to a different userid.

### **7.10.4 \$WEB\_FORM\_ACTION**

Returns the exact URL requested by a browser. It can be useful in writing HTML form-based applications where it is desired that the form post to the same URL that displayed it. This is generally a good idea and is especially useful in conjunction with \$WEB\_FORM\_DONE. Generally, a \$WEB\_FORM\_ACTION would be used to generate the data to appear in the *action* parameter of a *<form>* tag. If a URL is longer than 255 bytes, \$WEB\_FORM\_ACTION can be used in conjunction with \$WEB\_FORM\_ACTION\_LEN to generate a form action piecemeal.

### **7.10.5 \$WEB\_FORM\_ACTION\_LEN**

Returns the length of the exact URL requested by a browser. \$WEB\_FORM\_ACTION\_LEN is useful when \$WEB\_FORM\_ACTION is being used and there is a chance that the URL will be longer than 255 bytes.

### **7.10.6 \$WEB\_FORM\_DONE**

Can be used to accomplish two things.

First, it can be used to send a REDIRECT to a browser after a POST of a form. This REDIRECT will then receive the output generated by the application. This can be useful in updating, HTML form-based applications when you need to return error messages with a redisplay of a form (say when a user made some input errors) or you wish to send

an “update complete” message to the end-user. By issuing the REDIRECT you prevent the browser from creating an extra history entry for the result of the POST. These extra history entries cause all kinds of problems when a user uses a browser's backpage facility.

\$WEB\_FORM\_DONE can also be used to actually wait for the result of a POST of a form, effectively creating a persistent HTTP session. While this is not a highly recommended technique in the long run, it can be very useful as an interim solution when converting a complex multi-screen 3270 application to an HTML based application.

### **7.10.7 \$WEB\_PROTOCOL function**

Returns the bitmask associated with the SSL protocol being used by the browser: 1 for SSL V2, 2 for SSL V3 and 0 for non-SSL connections.

### **7.10.8 \$WEB\_SAVE\_LIST and \$WEB\_REST\_LIST functions**

\$WEB\_SAVE\_LIST allows the contents of a \$list to be saved in a web request so that it can be retrieved in a subsequent web request with \$WEB\_REST\_LIST or a \$WEB\_RESTORE\_LIST. These functions are most likely to be useful in web applications that produce some kind of list that can be scrolled through via “back” and “forward” links or buttons and the processing to produce the lists is expensive enough to be worth going to some effort to avoid where possible.

## **7.11 STOP FILE enhancement**

The STOP FILE command now causes saved found sets or LIST's (saved with \$WEB\_SAVE\_RECSET) to be deleted and prevents future saved found sets or LIST's from being saved. This means that a STOP FILE now can be used to prevent the *Janus Web Server* saved record set feature from preventing a file from being FREE'd from an Online or from urgent file maintenance being performed on a file such as an INCREASE command.

## *Sirius Performance Enhancements V2*

### 8.1 Macro language

A compile time macro language that makes it possible to conditionally compile blocks of code and to have macro variables that are evaluated at compile time to generate code. The conditional compilation has many uses. It is especially useful in preventing multiple definitions of the same subroutines when the procedure in which the subroutines reside is INCLUDE'd more than once, say once by an outer proc and once by an inner. For example:

```
PROCEDURE SUBR_COMMON
!DUPEXIT
SUBROUTINE COMMON(%INPUT IS FLOAT)
. . . . .
END SUBROUTINE
END PROCEDURE
```

prevents subroutine COMMON from being defined more than once as long as the subroutine is only defined in procedure SUBR\_COMMON. !DUPEXIT also allows you to specify a macro variable if COMMON is defined in multiple procedures. You can also use !DEF or !UNDEF to set a macro variable to *defined* or *undefined*, and then conditionally generate code using various !IFxxx and !ELSExxx macro statements.

### 8.2 MAXINCL parameter

The MAXINCL system parameter on the EXEC PARM card or the user zero parameters makes it possible to nest INCLUDE'd procedures to a depth greater than *Model 204's* limit of 5.



## 9.1 Comment-initialized globals

*SirFact* causes dummy string variables that begin with a backslash to have a default value of '\*' rather than null. That is, a dummy string that begins with a '?&\' will be treated as an asterisk if the global is not set. This makes these types of dummy strings useful for commenting out debugging code. When the debugging code is to be activated, the global could simply be set to null. The X'04' bit of the SIRFACT system parameter disables this feature if it is deemed undesirable.

## 9.2 Open records dumped in SirFact dumps

*SirFact* now dumps all "open" records at the time of a request cancellation. These are all records that are in currently active FOR EACH RECORD and FOR RECORD NUMBER loops. Field values can be displayed when scanning a *SirFact* dump by typing D F.fieldname where fieldname can be an explicit fieldname and occurrence number or can contain wildcard characters and occurrence number ranges. For example, D F.\* would display all occurrences of all fields in all open records.

## 9.3 Found sets and LIST's dumped in SirFact dumps

*SirFact* now dumps the count of records in or on and a certain number of record numbers in or on a foundset or LIST. The number of record numbers dumped can be set with the SIRFACT RECNDUMP command and defaults to two which means that the first and last record number in a found set or LIST will be dumped. To display the count of records and RECNDUMP record numbers from a found set, type D RIN.label when scanning a *SirFact* dump. D RON.listname does the same thing for LISTS.

## 9.4 SIRFACT DUMP command enhancements

*SirFact* dumps can now be further controlled based on message number or the SNAP keyword on the SIRFACT DUMP command. For example:

```
SIRFACT DUMP SUBSERR DUMP.+P.+T M204.0553
```

would send all dumps for subscript range errors to file SUBSERR and

```
SIRFACT DUMP FACTSNAP DUMP.+P.+T SNAP
```

would send all dumps resulting from SIRFACT SNAP commands to file FACTSNAP.

## **9.5 SIRFACT SNAP command**

A system manager can now take a *SirFact* dump of another user by issuing the SIRFACT SNAP command with the user number or userid of the user for which the dump is to be taken. SIRFACT snaps can even be taken for non-evaluating threads though the snap will have precious little besides GTBL.

---

**CHAPTER 10** *SirMon***10.1 New External Call Facility stats**

*SirMon* now provides all the standard statistics produced by the *Model 204* External Call Facility. These statistics are now supported by the \$SYSTAT, \$USSTATL and \$USSTAT functions. In order to access them from the SIRMEN subsystem, you must have installed *SirMon* version 5.4 or higher.

**10.2 New MQ Series stats**

*SirMon* now provides all the standard statistics produced by the *Model 204* MQ Series interface. These statistics are now supported by the \$SYSTAT, \$USSTATL and \$USSTAT functions. In order to access them from the SIRMEN subsystem, you must have installed *SirMon* version 5.4 or higher.



---

 CHAPTER 11 *SirPro*

## 11.1 Model 204 editor improvements (SIREEDIT parameter)

Several enhancements to the *Model 204* editor are available to customers authorized for the *SirPro* product. These changes are designed to improve the usefulness of the *Model 204* editor, especially for mixed-case programming such as that found in procedures that generate or contain HTML. Wide screen Mod 5 3270 sessions (with the `RESET MODEL 5 Model 204` command) are fully supported in non-DBCS environments.

The per-user, resettable SIREEDIT parameter contains the following bits:

- X'01'** Switch to \*LOWER mode upon entry to the editor, revert to prior setting upon exit.
- X'02'** Change the LINEND character for the current procedure to carriage return (X'0D'). This allows procedures to include semi-colons as data and also makes it easier to edit a procedure that has been uploaded from an HTML generator.
- X'04'** Always force the current 3270 Mod 2 behavior (24 lines by eighty characters), even if the editor is invoked by a thread using a mod 5 session. This bit takes precedence over the following bit.
- X'08'** Allow mod 5 sessions in DBCS environments. Ordinarily the widescreen (mod 5) support is disabled under DBCS environments. When this bit is set, even DBCS sessions are eligible for widescreen support.
- X'10'** Provide case-insensitive locate in non-DBCS environments. Thus, the editor command

```
/this is
```

will locate both of the following lines, regardless of the setting of \*LOWER/\*UPPER or the SIREEDIT X'01' bit:

```
This is sometimes hard to find due to mixed
case, while "this is" is all lower case.
```

- X'20'** Provide case-insensitive search strings for the replace command, while still retaining the case as entered on the replacement string, assuming that either \*LOWER is in effect or the X'01' bit of SIREEDIT is set. Under these conditions and using the text from our previous example, the command:

```
r/this is/This Is/* all
```

will change occurrences on both lines, resulting in:

This Is sometimes hard to find due to mixed case, while "This Is" is all lower case.

The default value for the SIREEDIT parameter is 19 (X'13'), which provides case-insensitive locate and carriage-return for linend.

---

**CHAPTER 12** *SirSafe*

## 12.1 Support for read-only files under MVS

*SirSafe* can now be configured to provide support for read-only files under OS/390™ and MVS™ environments. By default this support is deactivated. In order to take advantage of read-only files, the system manager must explicitly activate MVSRO mode with the AUTHCTL command.

When *SirSafe* is active in MVSRO mode, additional checks are performed whenever a *Model 204* database file is *physically* opened. For each dataset comprising the *Model 204* database file, the external authorizer (RACF or ACF2) will be invoked to determine if the *Model 204 job* is running under a profile that allows write access. If so, the dataset will be opened for output, else the dataset will be opened for input.

If any of the datasets for a *Model 204* database file are opened just for input, then the *Model 204* database file will be forced into read-only mode. Whatever privileges would have been granted to the opening user will be logically and'ed with X'8763' and the *Model 204* message M204.0620 will be produced. If the first (or only) dataset for a *Model 204* database file is opened just for input, then the *Model 204* message M204.0590 will be produced and shared DASD enqueueing will be deactivated.

In order to activate read-only file support, the System Manager must use the AUTHCTL command. If *SirSafe* is already active, use the AUTHCTL LIST command to display the current *SirSafe* configuration:

```
authctl list sirsafe
AUTHCTL A SIRSAFE REQUIRED MVSRW -
        RACF=M204*
```

The keyword MVSRW indicates that read-only support is inactive. Because *SirSafe* is running in REQUIRED mode, visible password entries may exist in CCASTAT. In our example, read-only processing could be enabled with the following command:

```
AUTHCTL C SIRSAFE REQUIRED MVSRO RACF=M204*
```

The keyword MVSRO indicates that read-only support is active.

For most jobs the overhead of read-only support should be insignificant. That is because most commonly used *Model 204* database files tend to remain physically open for the life of a run. However, certain kinds of unusual jobs could experience degradation. An example could be an IFAM host language job that performs many IFOPEN and IFCLOSE calls.

If AUTHCTL TEST ON has been activated, two new messages track the *SirSafe* dataset access checking. Message MSIR.0597 indicates the dataset being checked and the user ID and group for the access. MSIR.0598 indicates failure or success of the check. If *Model 204* attempts to open a database file without *SirSafe* MVSRO active, and the job has only read access to one or more of the datasets comprising the file, an IEC150I message will be produced, a 913 ABEND will be intercepted and the open will be rejected with an M204.054 error message. Note that it is still possible to receive an IEC150I message even when *SirSafe* MVSRO. is active. This is because *SirSafe* MVSRO processing just checks for *update* access to each dataset of a *Model 204* database file. If a *Model 204* job has **no** access to a dataset, an open in read-only mode will still be attempted.

## 12.2 Reduction in “noise” logging

A *Model 204* run with *SirSafe* active for file/group password access validation can log a large number of RACF or ACF2 access violations. In many cases this is a normal result of exploiting a strength of *SirSafe*, namely the ability for a single file or group password to confer different privileges to different users.

This is accomplished by having more than one password entry for a file or group with the same password value. When an end-user provides a password, CCASTAT is searched in the collating sequence of the *index character* for the appropriate file/group entries. If the password value in the entry matches the value entered by the user, then the external authorizer (RACF or ACF2) is used to verify that the end user has access to the CCASTAT entry. If access is denied, then the next CCASTAT entry with a matching password is checked.

Starting with release 6.0 of the *Sirius Mods*, the checks for access to CCASTAT entries will be performed with a NOLOG option if *Model 204* is run with APF authorization. Thus, no RACF or ACF2 violations will be logged as a result of an attempted *Model 204* file or group open. However, if a user enters a password that is not present in CCASTAT or if the user does not have access to a CCASTAT entry with the password, a standard *Model 204* error message will be issued. If the *SirSafe* test facility is activated by an AUTHCTL TEST ON command, the positive and negative results of access checks for CCASTAT entries will be logged to the journal and in the VIEW ERRORS table by MSIR.0552, MSIR.0553, MSIR.0554 and MSIR.0557 messages.

---

**CHAPTER 13** *SirScan*

### **13.1 Combined selection criteria**

*SirScan* now allows selection criteria to be combined with the “&” separator. For example, a selection criterion of “IODEV15&SCRATCHY IODEV7&ITCHY” would return journal records for all IODEV15's logged on as user “SCRATCHY” and IODEV7's logged on as user “ITCHY”.

### **13.2 SCANTIME system parameter**

The SCANTIME system parameter indicates the maximum number of seconds between *SirScan* “heartbeat” messages to the audit trail. These messages indicate the userid and, where appropriate, IP address and Janus port name associated with the thread and are only issued when a thread is about to send a message to the audit trail and it has not issued a *SirScan* heartbeat message in SCANTIME seconds. *Model 204* journal records are not stamped with the associated userid or any other useful identifying information other than the user number so the *SirScan* “heartbeat” messages are useful for obtaining such information about journal messages without ever having to go back more than SCANTIME seconds in the audit trail. While this can be useful even for examining journal entries formatted by AUDIT204, it is especially useful when looking at data using the SIRSCAN subsystem which automatically uses these “heartbeat” messages to determine whether a message is to be included in its output. By optionally starting SCANTIME seconds before the requested time range, *SirScan* can determine the userid, IP address and Janus port number associated with any and all messages in the journal.

### **13.3 Improved userid support**

*SirScan* will now make every effort to associate messages with a userid even if the user is no longer logged on. This includes things like login messages or user stat blocks or *SirScan* “heartbeat” messages produced because of a non-zero SCANTIME setting. This makes userid based selection criteria in *SirScan* much more usable, even in a *Janus Web Server* environment where logons tend to be fairly brief. In fact, by using a non-zero SCANTIME journal records can be deterministically assigned to a specific userid no matter when the user logged on or off.

## **13.4 New selection criteria**

*SirScan* now allows displayed journal records to be limited by IP address and Janus port name.

---

**CHAPTER 14** *Sirius Functions*

The following \$functions are new in version 6.0:

**14.1 \$BIND\_LIST**

Returns, into a \$list, data about semaphores bound using \$BIND.

**14.2 \$C2D**

Returns a number whose binary representation is given by a byte string argument.

**14.3 \$D2C**

Returns a byte string which is the binary representation of a numeric argument.

**14.4 \$FIELD\_LIST**

Returns field names and values into a \$list. This is an alternative to PAI INTO that places field data into a \$list so that one can take advantage of all the power of \$lists to do things like sorting and searching while avoiding the headache of trying to allocate an appropriate size array or running the PAI INTO statement in a loop if the number of array entries is insufficient to hold all the field values.

**14.5 \$IMGINF**

Retrieves the value of an image item. Because the image item name can be in a %variable this allows retrieval of image item values where the image item name is dynamically generated at evaluation time.

## **14.6 \$IMGOVL**

Updates the value of an image item. Because the image item name can be in a %variable this allows updating of image item values where the image item name is dynamically generated at evaluation time.

## **14.7 \$LISTADJ**

Changes the length of a \$list item.

## **14.8 \$LIST\_CAPTURE**

Captures *Model 204* output to a \$list. This output would most likely be the result of PRINT statements though output from the new HTML and TEXT statements will also be captured and \$LIST\_CAPTURE has an option to capture evaluation-time error messages.

## **14.9 \$LISTCHK**

Validates a \$list identifier.

## **14.10 \$LISTFINDI**

Locates a \$list item that contains a particular value at the same position and in the same format as an indicated image item.

## **14.11 \$LISTFINDI\_SUB**

Creates a subset of a \$list of items that contain a particular value at the same position and in the same format as an indicated image item.

## **14.12 \$LISTIMG\_COPY**

Copies the \$LISTIMG image association of a source \$list to a target \$list.

## **14.13 \$LISTOVLI**

Updates part of a \$list item at the same position and in the same format as an indicated image item.

## **14.14 \$LISTREP**

Replaces the contents of a \$list item with a string.

## **14.15 \$LISTREPI**

Replaces the contents of a \$list item with the contents of an image.

## **14.16 \$RESETN**

Changes the value of certain *Model 204* parameters; of note is MSGCTL, so you can suppress error messages during a request, with more granularity than the APSY error message control, or for a non-APSY application.

## **14.17 \$PROCOPN of temp request**

\$PROCOPN now allows a previous request number (that is, 0, -1, -2, ...) as the procedure to be opened, thus allowing \$PROC DAT, \$PROC GET, and \$PROC LOC to operate on a previous request.



---

**CHAPTER 15** *Compatibility/Bug fixes*

This section lists any compatibility issues with prior versions of the *Sirius Mods* and any bugs which have been fixed in this version of the *Sirius Mods* but had not, as of the date of this release, been fixed in the immediately prior version (5.5).

It also lists any differences in processing that result from execution with *Sirius Mods* version 6.0, as compared with the same inputs to *Sirius Mods* version 5.5.

## 15.1 Sirius Functions

### 15.1.1 Image association saved for \$list derivation \$functions

\$LISTCPY, \$LISTSORT, \$LISTSRT and \$LISTSUB now propagate the image association of their input \$lists to their output \$lists. Before *Sirius Mods* version 6.0, these functions would clear the image association of their output \$lists. This should not cause any backward compatibility problems with User Language that uses these \$functions because the only code that would pick up the new image associations would have failed before *Sirius Mods* version 6.0.

## 15.2 Janus TCP/IP Base compatibility issues

### 15.2.1 DEB[UG] keyword replaced by TRACE

The syntax of the former JANUS DEBUG, JANUS DEFINE, and JANUS DISPLAYDEBUG commands has been changed by replacing the strings DEBUG or DEB with TRACE. This is to clarify the distinction with the JANUSDEBUG command, which is new in version 6.0.

The old commands and parameters are still allowed as aliases for TRACE, but an informational message is issued to alert you to the change. Also, in some future version, Sirius Software may disable these old uses of DEBUG.

Also, JANUS DISPLAYTRACE and JANUS DISPLAYDEBUG now show the current settings as a hexadecimal string as in

```
JANUS TRACE WEBPORT X'03'
```

This should make it easier to determine which bits are on.

## **15.2.2 [NO]AUDTERM**

The NOAUDTERM and AUDTERM parameters of JANUS DEFINE now apply to the SDS, OPENSERV, and SRVSOCK port types, in addition to WEB ports (as previously). Also, the default has been changed from AUDTERM to NOAUDTERM, and even with NOAUDTERM in effect, compiler messages are sent to the audit trail.

This introduces a small incompatibility. Starting with version 6.0, compared to earlier versions, any WEB port connection without an explicit AUDTERM or NOAUDTERM will probably generate fewer audit trail lines, as will any SDS or OPENSERV port. This should be a benefit, since most of this output is either uninteresting or already logged to the audit trail as ER, AD or MS lines. Logging these messages as RK lines as well is just a waste of journal space and I/O and makes application diagnosis and debugging from the audit trail more difficult because of the extra noise data. For WEB, OPENSERV, or SRVSOCK applications that wish to explicitly audit information, the User Language AUDIT statement should be used, not the PRINT statement.

## **15.3 Janus Network Security compatibility issues**

### **15.3.1 SSL buffer sizes**

SSLBSIZE on the JANUS DEFINE command now sets both the input and output buffer sizes. Previously it only set the input buffer size and the output buffer size was always 256 bytes. In addition, the default output buffer size if SSLBSIZE is not specified on a JANUS DEFINE command has been changed from 256 to 4096. This means that the memory requirements of each SSL thread will go up by either 4096-256 or 3840 bytes if SSLBSIZE was not specified or N-256 bytes where N is the value for SSLBSIZE if it was specified. This increase should not be significant unless there are a huge number of SSL threads and even then output buffer sizes can now be independently and explicitly set with the SSLOBSIZE parameter.

## **15.4 Janus Web Server compatibility issues**

### **15.4.1 Legacy support now uses JavaScript**

*Janus Web Server* legacy applications now, by default, use JavaScript to improve the HTML presentation of User Language READ SCREEN statements. If JavaScript is prohibited on web pages at a site or is deemed undesirable, it can be suppressed with NOLEGJS parameter on the JANUS DEFINE for a port or with the like-named parameter in JANUS WEB SCREEN rules for selective URL's or even on an application level by setting NOLEGJS using \$WEB\_SCREEN.

### 15.4.2 Legacy support now uses REDIRECT's

*Janus Web Server* legacy applications now, by default, use a REDIRECT scheme that causes all of the HTML-mapped screen interactions to occupy just one entry in the browser history cache. This provides much cleaner browser behavior for these sessions but does produce different behavior from previous versions. If for some reason the old behavior is deemed desirable, it can be requested with the NOSCREENREDIR parameter on the JANUS DEFINE command for the port or with the like-named parameter in JANUS WEB SCREEN rules for selective URL's or even on an application level by setting NOSCREENREDIR using \$WEB\_SCREEN.

The most likely reason one might prefer the old behavior is if a legacy 3270 application is being accessed from a common public URL through a proxy server or with VARIPADDR set so that IP addresses cannot be used to identify users and without WEBCOOKIEID so that all users/browsers look the same from *Janus Web Server's* perspective. In such a scenario, legacy pages between users will be confused because all users will be requesting the same URL. Even then, however, it would be easy to take advantage of the new *Janus Web Server* legacy support behavior simply by specifying WEBCOOKIEID on the port definition or by adding some unique ISINDEX data to the URL that invokes the legacy session. A time stamp with a user number is probably sufficient to uniquely identify any legacy session.

### 15.4.3 \$WEB\_SEL family now returns null

\$WEB\_SEL, \$WEB\_SELs, \$WEB\_SELp, \$WEB\_SELSP and \$WEB\_SELPS used to return a 0 but now return a null. This was done to make it more convenient to place \$WEB\_SEL function calls inside an HTML/END HTML block without having to deal with the spurious 0's these function calls would produce. This change was actually distributed previously as ZAP5582.

## 15.5 Fixes in *Sirius Mods* 6.0 but not in 5.5 or 5.6

This section lists other fixes to functionality existing in the *Sirius Mods* version 5.5 but which, in absence of customer problems, have not, as of the date of the release, been fixed in that version.

**There are no problems without fixes.**

## **15.6 Version co-requisites**

This section lists any restrictions on usage of various products (including *Sirius Mods* itself) which will be imposed by use of version 6.0 of *Sirius Mods*.

**There are no version co-requisites.**