
Using IBM HTTP Server and Rexx to view z/OS STC output via SDSF

Techdoc: TD106087

This document can be found on the web at:
www.ibm.com/support/techdocs
Search for document number **WP106087** under the category of "TechDocs"

Version 1.0

Author: Edward McCarthy
IBM
edwardmc@au1.ibm.com

Last Updated: 20 March 2013, 10:54

Table of Contents

1	WHY A BROWSER INTERFACE TO VIEW THE JES SPOOL	4
1.1	Introduction	4
1.2	Why an interface via IBM HTTP Server?	4
1.2.1	Rexx and SDSF	4
1.2.2	Does IBM HTTP Server have to run on z/OS?	5
1.2.3	What about the CPU overhead?	5
1.2.3.1	Measuring CPU	5
1.2.4	Warranty and support	5
1.3	What about security?	5
1.3.1	SDSF still controls security	6
1.4	Tested browsers	6
1.4.1	Javascript	6
1.5	JESplex considerations	6
1.6	JES3 Support	6
1.7	RACF	6
1.8	Acknowledgements	7
2	USING THE SDSF VIEWER.....	8
2.1	Initial Menu.....	8
2.1.1	Navigation	8
2.2	Setting the prefix and LPAR Scope.....	8
2.2.1	Setting STC Prefix.....	8
2.2.2	Setting LPAR Scope.....	9
2.3	Viewing z/OS Syslog	9
2.4	Viewing the Active STC's	10
2.5	Viewing the Completed STC's.....	11
2.6	Handling of very large STC output.....	13
2.7	Downloading STC output as a compressed file.....	14
2.7.1	Obtaining gzip for z/OS.....	15
3	CONFIGURATION.....	16
3.1	Files supplied with this techdoc	16
3.1.1	Our test environment	16
3.1.2	Sample httpd.conf files	16

3.2	Customise Rexx program variables	16
3.3	Setting up the IBM HTTP Server	17
3.3.1	Set up IHS on z/OS.....	17
3.3.2	No need for UID=0.....	17
3.3.3	Advice on setting up IHS.....	18
3.3.4	Sample 'real world' setup process.....	18
3.3.4.1	Define userid.....	18
3.3.4.2	Create a suitable directory for the new IHS.....	19
3.3.4.3	Create the new IHS.....	19
3.3.4.4	Create a STC to execute the IHS.....	20
3.3.4.5	Verified IHS working.....	20
3.3.4.6	Backup current httpd.conf.....	20
3.4	Modifying IBM HTTP Server to execute the sdsfViewer Rexx.....	21
3.4.1	Allowing IHS supplied icons to be used.....	21
3.4.2	Directory for the supplied Rexx.....	21
3.4.3	Copy files to IHS sub-directories.....	22
3.4.4	Restricting IHS to just the sdsfViewer Rexx.....	22
3.4.5	Start IHS and test.....	23
3.5	Configuring IHS for your security requirements	24
3.5.1	Allowing unauthenticated access.....	24
3.5.2	Allowing all authenticated user access.....	24
3.5.3	Allowing authenticated user belonging to a group access.....	26
3.5.4	Allowing authenticated user access with client credentials.....	26
3.5.4.1	Required SAF definitions.....	27
3.5.5	Use of SSL.....	28
3.5.6	Allowing access to show outstanding WTOR messages.....	28
4	DEBUGGING GUIDANCE.....	29
4.1	Comments on SDSF access	29
4.1.1	URL to display who you are.....	29
4.1.2	The isfsysname variable.....	30
4.2	Some RACF and SDSF security issues you may come across.....	30
4.3	When things go wrong.....	30

1 Why a browser interface to view the JES Spool

1.1 Introduction

This document describes how to use a supplied Rexx program in IBM HTTP Server to view output of started tasks (STC) and jobs in the JES2 Spool on z/OS systems.

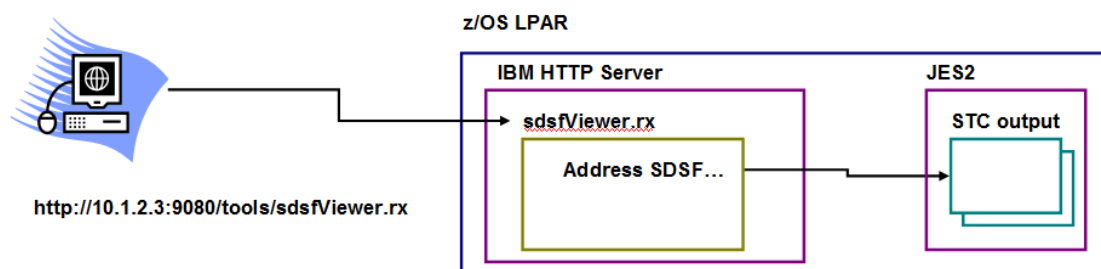
1.2 Why an interface via IBM HTTP Server?

Customers often have developers who have a requirement to view job logs or syslog stored in the z/OS JES2 Spool, but who are unfamiliar with 3270, TSO, ISPF and Spool Display and Search Facility (SDSF). Providing a web browser interface to the JES spool increases their productivity and reduces their reluctance to work on the z/OS platform.

For example, a customer may be running WebSphere Application Server (WAS) on z/OS and the applications running in WAS are writing application log messages to the default location called Stdout and Stderr. In WAS on z/OS these destinations are mapped to //SYSPRINT DD and //SYSOUT DD spool files in the WAS STC.

The Rexx program supplied with this document can be installed in an IBM HTTP Server powered by Apache on z/OS. Installation of the Rexx program into the HTTP server allows any permitted user with browser access to view these spool datasets. The end user can then navigate a simple menu to view the output they want to see.

The following diagram depicts the configuration:



The end user enters a URL which is received by the IBM HTTP Server. IHS then runs the Rexx program as a CGI type program. The Rexx program uses the Address SDSF Interface to access STC output in the JES2 Spool.

1.2.1 Rexx and SDSF

The Rexx program invoked in the IBM HTTP Server uses SDSF calls to access started tasks and job output in the JES2 Spool.

This IBM Redbook describes in detail how to use Rexx to interact with SDSF:

<http://www.redbooks.ibm.com/abstracts/sg247419.html>

This link provides the entry point in the z/OS 1.13 Infocenter to information about Rexx and working with SDSF:

http://publib.boulder.ibm.com/infocenter/zos/v1r13/topic/com.ibm.zos.r13.isfa500/rexx.htm?path=34_0_15#rexx

1.2.2 Does IBM HTTP Server have to run on z/OS?

To be able to use the Rexx program which invokes SDSF APIs that is supplied with this document, the IBM HTTP Server must be running on a z/OS LPAR.

1.2.3 What about the CPU overhead?

While no measurements have been undertaken, it is expected that the amount of CPU used by the Rexx program running in IBM HTTP Server on z/OS to view SDSF output would be similar to that used by a user doing the same access via TSO and SDSF.

The primary aim is to make it simpler for developers with no TSO and SDSF experience to be able to view what they want to view in the JES2 Spool.

1.2.3.1 *Measuring CPU*

The IBM HTTP Server runs as a started task and as such will produce SMF 30 type records which will contain information about how much CPU is used.

1.2.4 Warranty and support

This Rexx program is supplied as-is.

No warranty from IBM is supplied or implied in any way.

IBM will provide no official support for this program.

You may however contact the program author at edwardmc@au1.ibm.com if you have queries or issues who will respond as time permits.

The author would welcome any feedback and suggestions.

1.3 What about security?

There are security controls around what STC output a user can view when they use SDSF to access JES2 Spool content. Each customer site can customise the security controls to suit their requirements.

The IBM HTTP Server (IHS) runs as a started task on z/OS under an assigned userid. In the simplest setup, all client requests run under the IHS userid and will access SDSF as that userid.

It is possible to configure IHS so that the client's userid is authenticated, but the Rexx program is run under the userid of the server or the userid of the client.

We will cover this in more detail later in this document.

1.3.1 SDSF still controls security

Note that access via this Rexx running in the IBM HTTP Server to any STC output in the JES Spool is controlled by how your SDSF security configuration and by the SAF JESSPOOL class profiles. To SDSF, the calls from the Rexx program in the IBM HTTP Server are equivalent to a call from a batch job.

This link from the z/OS infocenter is a starting point of the topic about SDSF and security:

http://publib.boulder.ibm.com/infocenter/zos/v1r13/topic/com.ibm.zos.r13.isfa500/autbat.htm?path=34_0_14_3#autbat

1.4 Tested browsers

This Rexx has been tested in IE 8, Firefox and Chrome on Windows 7.

1.4.1 Javascript

Javascript is used to assist with displaying the STC output on browsers and consequently javascript support must be enabled in the browser to use this Rexx program.

1.5 JESplex considerations

If your z/OS environment has multiple JESplex's within a Sysplex, then you would need to run an IBM HTTP Server in each of these JESplex's to be able to use this Rexx to view output in each JESplex. It is not necessary to run an HTTP server instance on each member of the JESplex

1.6 JES3 Support

The supplied Rexx has not been tested with a JES3 environment. The Rexx program code may require some modifications to work successfully in a JES3 environment.

1.7 RACF

Throughout this document we reference RACF as the security product and that various RACF rules may need updating etc.

If your organisation uses an alternate security product then you would just need to make the equivalent security changes.

1.8 Acknowledgements

The author would like to acknowledge the assistance of:

Mike Cox of the IBM Washington Systems Center in producing this techdoc

Joseph Perillo of SDSF development for his assistant in developing the code

Willian Schoen of z/OS Unix development

2 Using the SDSF Viewer

This chapter describes how to use the SDSF Viewer.

2.1 Initial Menu

Use a URL like this to access the initial menu:

<http://<hostname>:<portnumber>/tools/sdsfViewer.rx>

Change the '<hostname>' to DNS name or TCPIP address of the z/OS LPAR where your HTTP server resides, and change the '<portnumber>' to the port on which the HTTP server is listening on.

This will produce an initial menu similar to this:



2.1.1 Navigation

All pages displayed by the Rexx program, apart from the above initial menu, contain a back arrow image, which when clicked will take you back to the previous page.

2.2 Setting the prefix and LPAR Scope

You can set a prefix to control what STC's are listed and also set an LPAR scope to limit the active STC display to a particular LPAR.

2.2.1 Setting STC Prefix

Before selecting the Active STC's or Completed STC's link, you can set a prefix. This prefix will be used as a mask to limit the entries displayed when you click on the Active STC's or Completed STC's link.

You may or may not need to set a prefix depending on how security is set up in the IHS.

For example if the userid the IHS server is running under only has access to view STC's that start with 'WAS', then you would probably not need to set any prefix, since the list displayed will be limited anyway.

On the other hand, if the userid the IHS is running under had access to all STC output, then you may well want to set the prefix to limit the list of STC's displayed to just those matching that mask.

To set the prefix type in the characters you want to match on and click the Set Prefix button.

Note matching of the prefix value to the STC name is done from the start of the STC name (i.e. E2* would match E2DMGR, E2XXX, etc.). There is no support for full wildcarding of STC names, E2%M%%S for example.

Depending on your SDSF setup, you may need to enter an asterisk at the end of the mask value.

2.2.2 Setting LPAR Scope

The LPAR dropdown box allows you to set an LPAR scope when displaying active STCs, LPAR scope is not applied to viewing syslog/operlog or completed STCs.

If you select the ALL value, then all STC's matching your prefix value will be display from ALL LPARs in the JESplex.

If you select a specific LPAR, then only STC's on that LPAR matching the prefix will be displayed.

The LPAR dropdown box does not appear when viewing completed STC output.

The Rexx program uses the SDSF MAS command to determine what LPARs are members of the JESplex the IHS is running in.

2.3 Viewing z/OS Syslog

You can view the z/OS syslog/operlog by clicking on the link called Syslog in the main menu.

This will display z/OS syslog/operlog in the browser, an example is shown below:

z/OS Syslog

Start date: 02/14/13 Time: 17:54:00

End date: 02/14/13 Time: 18:04:55

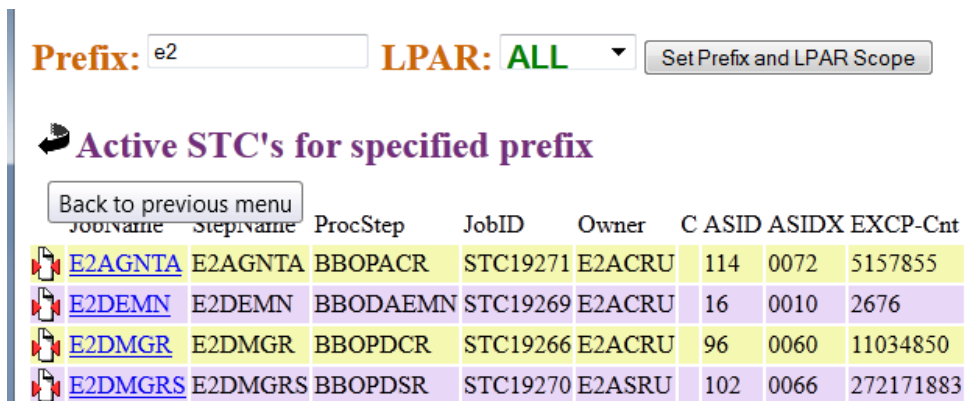
```
M 0000000 SYSC      13045 17:54:07.37      00000210 IWM034I PROCEDURE QCASRC STARTED FOR SUB
D                                     853 00000210 APPLICATION ENVIRONMENT QCSR01
E                                     853 00000210 PARAMETERS JOBNAME=QCSR01CS,ENV=QCCELL.Q
M 0000000 SYSC      13045 17:54:07.37      00000210 IWM034I PROCEDURE QCAARC STARTED FOR SUB
D                                     854 00000210 APPLICATION ENVIRONMENT QCSR01ADJUNCT
E                                     854 00000210 PARAMETERS JOBNAME=QCSR01CA,ENV=QCCELL.Q
M 0080000 SYSC      13045 17:54:07.39      00000010 IRR812I PROFILE QC*.* (G) IN THE STARTED
```

At the bottom of this display is shown outstanding WTOR messages.


2.4 Viewing the Active STC's





To view the active i.e. running STC's click the Active STC's link.

This will produce a display similar to the one shown below:



Prefix: LPAR: **ALL**

 **Active STC's for specified prefix**

Jobname	Stepname	ProcStep	JobID	Owner	C	ASID	ASIDX	EXCP-Cnt
 E2AGNTA	E2AGNTA	BBOPACR	STC19271	E2ACRU	114	0072		5157855
 E2DEMNI	E2DEMNI	BBODAEMN	STC19269	E2ACRU	16	0010		2676
 E2DMGR	E2DMGR	BBOPDCR	STC19266	E2ACRU	96	0060		11034850
 E2DMGRS	E2DMGRS	BBOPDSR	STC19270	E2ASRU	102	0066		272171883

To return to the main menu click the back arrow image. When the mouse is hovered above that image, the browser will display a pop up box saying 'Back to previous menu'.

If the variable `allowStcDownload` is set to 1 in the Rexx program, then the compressed file icon is displayed at the start of each line. Hovering over the compressed file icon will result in the browser displaying a pop up box saying 'Download as gz file'.

You can update the prefix value by typing in a new prefix and clicking the Set Prefix button. The display will be updated to list STC's that match the new prefix.

To view the various sections of an active STC, simply click the name of the STC you want to view.

This will produce a display similar to that shown below:



IBM z/OS SDSF Viewer

Viewing: E2AGNTA STC19271

	DDNAME	StepName	ProcStep	DSID	Owner	C	Dest	Rec-Cnt	Page-Cnt	Byte-Cnt	CC	I
	JESMSG LG	JES2		2	E2ACRU	S		7426		496270	1	
	JESJCL	JES2		3	E2ACRU	S		89		5761	1	
	JESYSMSG	JES2		4	E2ACRU	S		7601		346126	1	
	SYSOUT	E2AGNTA		103	E2ACRU	S		270		31272	1	
	SYSPRINT	E2AGNTA		105	E2ACRU	S		11021		872481	1	

To go back to the previous menu click the back arrow image.

If the variable `allowStcDDNameDownload` is set to 1 in the Rexx program, then the compressed file icon is displayed at the start of each line.

To view the contents of any these STC entries simple click the required entry. This will produce a display similar to that shown below:



IBM z/OS SDSF Viewer

Viewing: E2AGNTA STC19271

- JESMSG LG
- JESJCL
- JESYSMSG
- SYSOUT**
- SYSPRINT

STC19271

```

1 //E2AGNTA JOB MSGLEVEL=1
2 //STARTING EXEC E2ACRA,ENV=E2CELL.
3 XXE2ACRA PROC ENV=,PARMS=' ',REC=
4 XX SET ROOT='/wasv8config/e2cell/e
5 XX SET FOUT='properties/service/logs/applyPTF.out'
6 XX SET FOUT2='properties/service/logs/parmsRec.out'
7 XX SET WSDIR='AppServer'
  XX*****
  XX* Test that OMVS can successfully launch a shell and return *
  XX*****

```

To go back to the previous menu click the back arrow image.

There is a drop down box that lists all the DD entries of the STC. You can jump automatically by moving the mouse to the DD entry you want to view and clicking on it.

2.5 Viewing the Completed STC's

To view the completed STC's, click the Completed STC's link.

This will produce a display similar to the one shown below:

IBM z/OS SDSF Viewer

Prefix:

Completed STC's for specified prefix

JobName	JobID	Owner	PrtY	Queue	C	Pos	SAff	ASys	Status	PrtDest	SecLabel	TGNum	TGPct
B8AGNTB	STC21079	B8ACRU	1	PRINT	404					LOCAL		17	0.07
B8AGNTB	STC21086	B8ACRU	1	PRINT	468					LOCAL		7	0.03
B8CRDB2	JOB20474	SYSADM1	1	PRINT A	309					LOCAL		1	0.00
B8DBGUNT	JOB21126	SYSADM1	1	PRINT A	410					LOCAL		1	0.00

To return to the main menu click the back arrow image.

You can update the prefix value if you want, by typing in a new prefix and clicking the Set Prefix button. The display will be updated listing STC's that match the new prefix.

To view the various sections of a completed STC, simply click the name of the STC you want to view.


This will produce a display similar to that shown below:

Viewing: B8CRDB2 JOB20474

DDNAME	StepName	ProcStep	DSID	Owner	C	Dest	Rec-Cnt	Page-Cnt	Byte-Cnt	CC	Rmt	Node
JESMSGLG			2	SYSADM1	H	LOCAL	19		1352	1		175
JESJCL	JES2		3	SYSADM1	H	LOCAL	18		1008	1		175
JESYSMSG			4	SYSADM1	H	LOCAL	66		4561	1		175
SYSTSPRT	DB2EXEC		103	SYSADM1	H	LOCAL	8		175	1		175

To go back to the previous menu click the back arrow image.

To view the contents of any these STC entries simple click the required entry. This will produce a display similar to that shown below:

 **Viewing: B8CRDB2 JOB20474** JESJCL ▼

```

1 //B8CRDB2 JOB 1,'DROP DB2 TBLs',C
  // MSGCLASS=H,TIME=1440,REGION=0
  // NOTIFY=HUTCH
  /*JOBPARM S=SYSB
  //*****
  /* JOB TO DROP DATABASES, STOGROUPS, FOR B8CELL - WPS 8.0
  //*****
2 //DB2EXEC EXEC PGM=IKJEFT01,DYNAMNBR=20
3 //STEPLIB DD DISP=SHR,DSN=DSN1010.DBPO.SDSNEXIT
4 //          DD DISP=SHR,DSN=DSN1010.SDSNLOAD
5 //          DD DISP=SHR,DSN=DSN1010.SDSNLOD2
  
```

PASSWORD=,

To go back to the previous menu click the back arrow image.


2.6 Handling of very large STC output

Output for a STC, especially for WebSphere Application Server, can at times be very large, for example over one million lines.

One million lines of output as 100 characters per line is 100M of data to send from IHS on z/OS to the browser. This could take some time depending on connection speed etc.

In the sdsfViewer.rx is a variable named `maxLines` which specifies the maximum number of lines to send to a browser.

When the selected STC output contains more lines than the value of `maxLines`, the Rexx program displays a list of links to sections of the output as shown below:

 **Viewing: E2DMGRS STC17977** SYSPRINT ▼

Lines

1	Processing Trace Settings File: /wasv8config/e2cell/e2dmnode/DeploymentManager/profiles/default/config/cells/e
5001	ExtendedMessage: mmd=com.ibm.ws.webcontainer.metadata.WebModuleMetaDataImpl@65e4e7e
10001	ExtendedMessage: port --> 58005
15001	SourceId: com.ibm.ws390.orb.ClientDelegate
20001	0170 382E303B 2057696E 646F7773 204E5420>?...+.. 8.0; Windows NT
25001	SourceId: com.ibm.ws.http.channel.impl.HttpBaseMessageImpl
30001	SourceId: com.ibm.ws.classloader.CompoundClassLoader.getResource com/ibm/ws/console/core/servlet/WSCUrI

The display shows the line of output at the start of each segment of output, which may help you to select which one to view.

When you click on a segment the display will show as follows:

Viewing: E2DMGRS STC17977 **SYSPRINT**

*** Displaying ***

Lines 50001 **55001-60000** 60001

```
{JSESSIONID/ibm
 {
  _reqFromCookie
  _reqFromURL=fa
  _requestedSess
  _requestedSess
  _redirectClone
  _reqFromSSL=fa
  _allSessionIds
  _inputCloneIn
  _responseSessi
  _responseSessi
  _outputCloneIn
 }
}
Trace: 2013/01/
SourceId: com
ExtendedMessa
Trace: 2013/01/
```

ntext #

dCRFVsghg

000025C0000000309521845

dCRFVsghg

=8CBE88 c=UNK key=P8 tag= (13007004)

essionCore.SessionContext.getSessionAffinit

quest for key JSESSIONID/ibm

=8CBE88 c=UNK key=P8 tag= (13007004)

The output will show the STC output for the selected line range. There are links to the previous and next output segments. There is also a dropdown box that contains links to all the output segments. You can automatically jump to a different segment by using the mouse to select the required segment and then clicking on it.

Note this drop down display works best in Firefox. In Chrome and IE, the drop down box displays, however the font size is not the smaller size it is in Firefox.

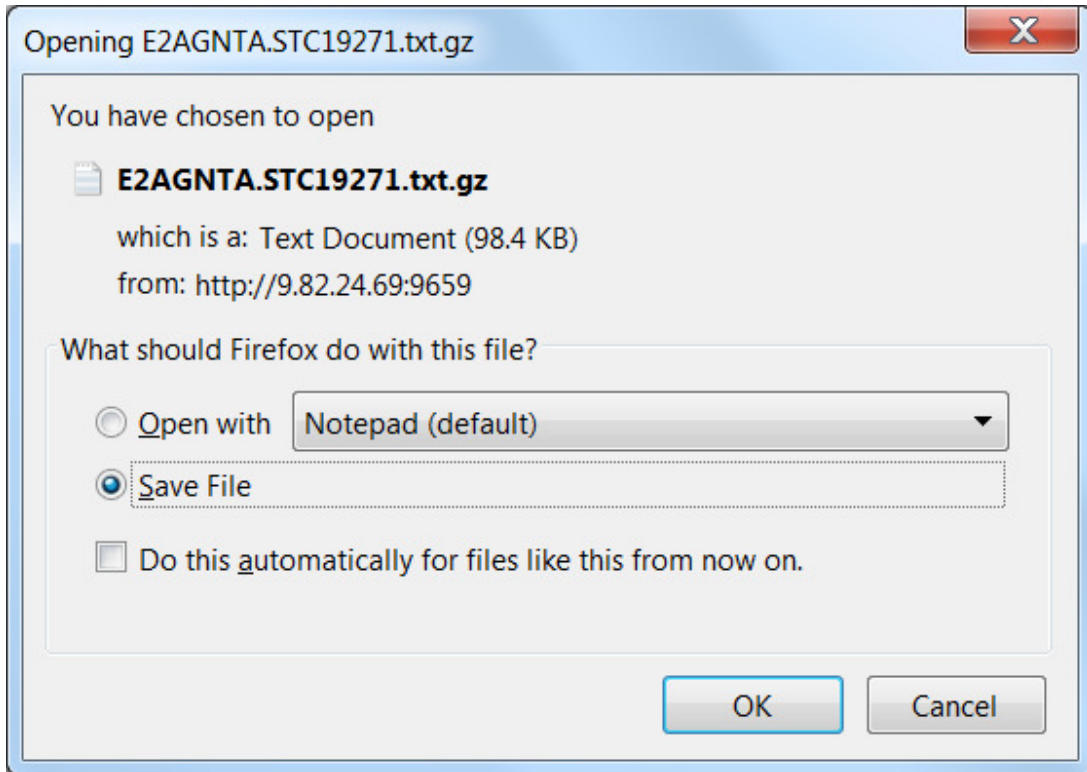
2.7 Downloading STC output as a compressed file

As mentioned above, there is a compressed file icon displayed to the left of STC entries. When you click on that icon, the Rexx program reads the output for the selected STC entry and sends the contents as a compressed file with a gz suffix to the browser.

No actual file is created on the z/OS LPAR to achieve this, the compressed output is just sent directly back to the browser as it generated.

Note in Firefox and Chrome a new tab will temporarily be opened in your browser while in IE a new Window will be temporarily be opened to handle the processing of the download. This is done so that your current browser view is preserved.

Your browser will then prompt you about what to do with the file, for example in a Firefox browser you will see:



Select the Save File option and click OK, to get a file dialog box that will allow you to specify where to save the file to.

Once the file has been saved, you can uncompress it to get a text file containing the selected STC output.

On a Windows system, you may need to have a product that knows how to uncompress and unzip Unix gz type files. There are paid and freeware products that run on Windows to do this task.

Additionally on a Windows system, you will need to use Wordpad or some other suitable product to view the output, as it will not display properly formatted in the standard Notepad program.

2.7.1 Obtaining gzip for z/OS

To be able to use the downloading compressed file option requires that the gzip utility is available on the z/OS LPAR.

The gzip utility is not shipped by default with z/OS. You can download it from this URL:

<http://www-03.ibm.com/systems/z/os/zos/features/unix/bpxa1ty1.html>

3 Configuration

This chapter describes how to set parameters in the supplied Rexx and how to configure the IBM HTTP Server.

3.1 Files supplied with this techdoc

Supplied with this techdoc is a zip file containing:

- sdsfViewer.rx – the Rexx program
- zec12.jpg - image to display on browser output
- httpd.conf - file used in our IHS for testing
- httpd-min.conf – alternate file with minimal required directives

Note the supplied httpd.conf and httpd-min.conf file contains comment lines identifying the changes we made, these can be found by searching for the string IBM-Rexx.

3.1.1 Our test environment

We developed and tested this Rexx using IBM HTTP Server V8.5 on z/OS 1.13.

This Rexx program should work in earlier versions of the IBM HTTP Server, though there may be some differences in the modules required for authentication.

3.1.2 Sample httpd.conf files

Both the supplied httpd conf files started out as the IBM supplied default httpd.conf file.

The same changes were made to both the httpd.conf and httpd-min.conf file.

All comments have been removed from the httpd-min.conf file which may help those that are less familiar with setting up the IBM HTTP Server on z/OS, since this makes it a smaller file to view and manage.

It is recommended that you use the supplied sample httpd.conf files as a guide to how to modify your own httpd.conf file to be able to run the supplied Rexx.

3.2 Customise Rexx program variables

The following table lists the variables in the sdsfViewer.rx program that can be customized to suit your environment:

Variable Name	Default Value	Description
mainImageName	../images/zec12.jpg	Identifies image to display at top of output in browser

sdsfRexxPath	/tools	Value defined in ScriptAlias directive in the httpd.conf file The ScriptAlias directive points to the actual directory containing this Rexx
maxLines	5000	Maximum number of lines to display
maxLinesRead	2000	Maximum number of lines to read from a STC at a time
allowStcDownload	1	Allow all the output for a STC to be downloaded as compressed file
allowStcDDNameDownload	1	Allow sections of an STC to be downloaded as compressed file
supportContact	your z/OS system programmer on 555-1234	Contact details who end user should contact if error message displayed
gzLoc	/local/tools/bin	Directory where the gzip program is located
timeSpan	10	How many minutes from current time of the syslog to display
maxLines	2000	Maximum number of lines of syslog to display

3.3 Setting up the IBM HTTP Server

To get the supplied Rexx program to work in the IBM HTTP Server (IHS) on the z/OS LPAR you will need to do the following.

3.3.1 Set up IHS on z/OS

Obviously you will need a working IBM HTTP Server on a z/OS LPAR.

The techdoc at this link describes in detail how to set up IHS on a z/OS LPAR:

<http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101170>

This is a link to the WebSphere Application Server V8.5 infocenter that also explains how to setup an IHS on z/OS:

<http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/topic/com.ibm.wesphere.ihs.doc/ihs/welc6miginstallihsz.html>

3.3.2 No need for UID=0

There is no requirement to set up the userid that the IHS STC runs under to have UID of zero.

It is strongly recommended that the HTTP server userid not have a UID of zero.

3.3.3 Advice on setting up IHS

If you are new to setting up an IBM HTTP Server on z/OS it is recommended that you follow the advice in the above techdoc to set up your IBM HTTP Server, get it up and running and verify you can access the home page as a first step.

We also explain in the next section how we setup our IBM HTTP Server.

The following sections then describe how to enable the IBM HTTP Server to run the supplied Rexx.

If you already have an IBM HTTP Server running on your z/OS environment then you can follow the steps in the following sections to enable your IHS to run the supplied Rexx.

The recommendation if you are starting with a new HTTP server is to use the enclosed httpd.conf file, which provides the bare minimum set of directives needed to access SDSF.

3.3.4 Sample 'real world' setup process

We will explain here the steps we used to setup an IBM HTTP Server in a manner similar to what we would recommend for use at a real customer site.

There are many ways you as a customer could set up our IBM HTTP Server to allow access to this sdsfViewer Rexx.

You may already have IBM HTTP Server setup for other purposes, in which case you can make the required modifications described.

Alternatively you may want to set up a new IBM HTTP Server just to run this supplied Rexx.

3.3.4.1 Define userid

We created a userid called WEBSRV1 which we plan to use as the userid to run the IBM HTTP Server.

Note this userid must have an OMVS segment and should also have a home directory in the USS environment. This userid does not have a UID of zero.

Doing a LU WEBSRV1 OMVS command shows how we setup this userid:

```
USER=WEBSRV1  NAME=WEBSRV1  OWNER=SYS1
CREATED=11.251
  DEFAULT-GROUP=SYS1  PASSDATE=11.252  PASS-INTERVAL=N/A
PHRASEDATE=N/A
  ATTRIBUTES=NONE
  REVOKE DATE=NONE  RESUME DATE=NONE
  LAST-ACCESS=13.072/00:46:13
  CLASS AUTHORIZATIONS=NONE
  NO-INSTALLATION-DATA
```

```

NO-MODEL-NAME
LOGON ALLOWED      (DAYS)          (TIME)
-----
ANYDAY
GROUP=SYS1        AUTH=USE          ANYTIME
CONNECT-OWNER=SYS1  CONNECT-
DATE=11.251
CONNECTS= 244  UACC=NONE          LAST-CONNECT=13.072/00:46:13
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED

OMVS INFORMATION
-----
UID= 0000079791
HOME= /u/websrv1
PROGRAM= /bin/sh
CPUTIMEMAX= NONE
ASSIZEMAX= NONE
FILEPROCMAx= NONE
PROCUSERMAX= NONE
THREADSMAX= NONE
MMApAREAMAX= NONE

```

3.3.4.2 *Create a suitable directory for the new IHS*

On our z/OS LPAR we did the following:

1. created a directory called /shared/ihs_sdsf
2. created a new zFS called OMVS.IHS.SDSF.ZFS with about 300 tracks
3. mounted this zFS at the directory
4. Changed the owner of the /shared/his_sdsf directory to be WEBSRV1 and the group to be SYS1

Note you can use any directory as the home location for your IBM HTTP Server. The aim of us using a directory like /shared/ihs_sdsf is to reinforce the suggestion that using someone's home directory like /u/bob would not be recommended.

3.3.4.3 *Create the new IHS*

As per the advice in the WP101170 techdoc we then issued these commands:

```

su - websrv1
cd /usr/lpp/IHSA/V8R5BASE/bin
./install_ihs /shared/ihs_sdsf 9659

```

This produced this output:

```

Copying install directory and creating symlinks...
Updating install paths...

```

```
cmd: /shared/IHSA/V8R5BASE/bin/postinst -i /shared/ihs_sdsf -t
install -v PORT=9659 -v SERVERNAME=wsc1.washington.ibm.com
```

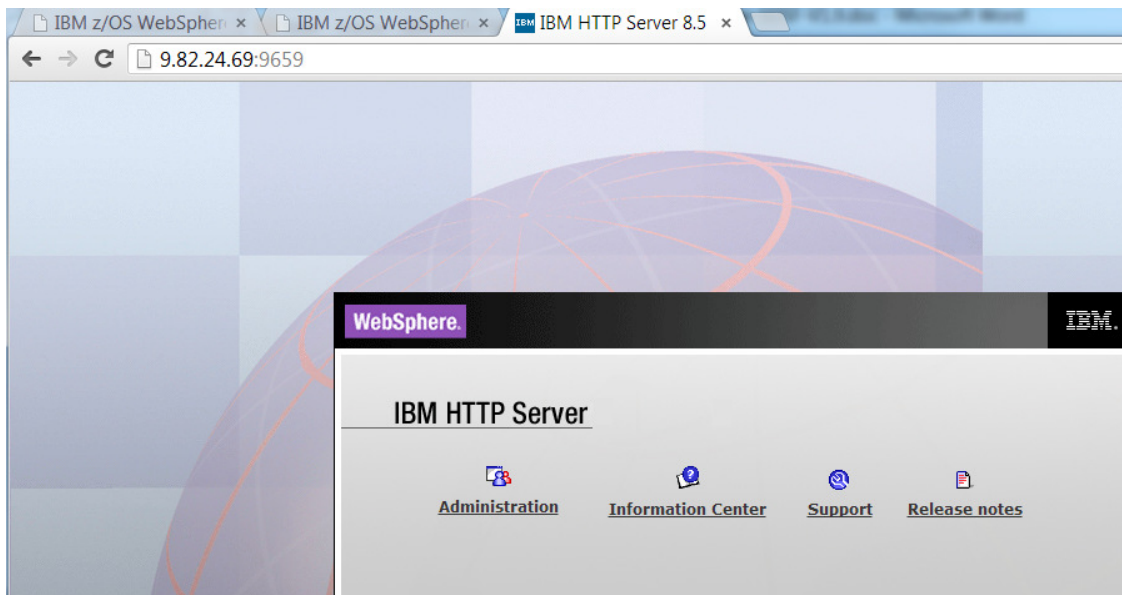
3.3.4.4 *Create a STC to execute the IHS*

We then created a STC called WEBSRV in SYS1.PROCLIB as per the example in the WP101170 techdoc. Our STC looked like this:

```
//*-----
//IHSAPACH PROC ACTION='start',
//          DIR='/shared/ihs_sdsf',
//          CONF='conf/httpd.conf'
//*-----
//IHS      EXEC PGM=BPXBATCH,
// PARM='PGM &DIR/bin/apachectl -k &ACTION -f &CONF -DNO_DETACH',
// MEMLIMIT=512M
//STDOUT   DD  PATH='&DIR/logs/proc.output',
//
//PATHOPTS=(OWRONLY, OCREAT, OTRUNC), PATHMODE=(SIRWXU, SIRWXG)
//STDERR   DD  PATH='&DIR/logs/proc.errors',
//
//PATHOPTS=(OWRONLY, OCREAT, OTRUNC), PATHMODE=(SIRWXU, SIRWXG)
//*  STDENV   DD  PATH='&DIR/conf/my_envvars'
//          PEND
```

3.3.4.5 *Verified IHS working*

We then URL <http://9.82.24.69:9659> to verify we could access the IHS default home page which produced this display:



3.3.4.6 *Backup current httpd.conf*

Before making the changes described in the following section, it is recommended to make a backup of the /shared/ihs_sdsf/conf/httpd.conf file.

3.4 Modifying IBM HTTP Server to execute the sdsfViewer Rexx

Having setup your IBM HTTP Server this section describes how to modify it so that the sdsfViewer Rexx can be invoked.

3.4.1 Allowing IHS supplied icons to be used

Two images supplied with the IHS product are used by the sdsfViewer.rx. They are located in the icons sub directory. The required icons are:

- back.gif
- compressed.gif

To allow them to be accessed by the HTTP server, the httpd.conf file your IHS sever is using requires the addition of one line.

In the conf file will be lines similar to this:

```
Alias /icons/ "/shared/ihs_sdsf/icons/"
<Directory "/shared/ihs_sdsf/icons">
    Options MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

Add the following line after the 'Options MultiViews' line:

```
Options FollowSymLinks
```

That section of the httpd.conf file would then look like this:

```
<Directory "/shared/ihs_sdsf/icons">
    Options MultiViews
    # IBM-Rexx Added next 1 line to allow icons to be used
    Options FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

The IHS server will need to be restarted to pick up this change.

3.4.2 Directory for the supplied Rexx

The cgi-bin sub-directory is where you typically store programs that the IHS executes such as this Rexx program.

It is good practice to use an alias in the URL when invoking programs from the cgi-bin directory. To achieve this we need to add a directive called ScriptAlias.

In the httpd.conf file locate these existing directives:

```
#
# ScriptAlias: This controls which directories contain server
scripts.
```

```

# ScriptAliases are essentially the same as Aliases, except that
# documents in the realname directory are treated as applications
and
# run by the server when requested rather than as documents sent
to the client.
# The same rules about trailing "/" apply to ScriptAlias
directives as to
# Alias.
#
ScriptAlias /cgi-bin/ "/shared/ihs_sdsf/cgi-bin/"

```

After the last directive above add this directive:

```
ScriptAlias /tools/ "/shared/ihs_sdsf/cgi-bin/"
```

In the sdsfViewer.rx program are the following lines:

```

/* sdsfRexxPath - Value defined in ScriptAlias directive in the
httpd.conf file
                                The ScriptAlias directive points to the actual
directory
                                containing this Rexx
*/

sdsfRexxPath = '/tools'

```

The Rexx program needs to have the variable called sdsfRexxPath updated to reflect the alias name used in the ScriptAlias directive.

You can use whatever alias name you prefer.

If you do not want to use the cgi-bin directory you can create a directory of your choosing, store the supplied Rexx in it and modify the ScriptAlias directive accordingly.

3.4.3 Copy files to IHS sub-directories

Copy using ASCII transfer, the supplied Rexx program called sdsfViewer.rx to the cgi-bin sub-directory of the IHS you have created.

Copy using binary transfer, this image file to the htdocs/images sub-directory of the IHS you created:

```
zec12.jpg
```

Note: check the permission bits of these files after they have been copied to the sub-directories, so that the userid of the IHS STC can access them.

3.4.4 Restricting IHS to just the sdsfViewer Rexx

If you would like to set up the IHS so that the only the sdsfViewer Rexx can be accessed then you achieve this by the following modifications to the httpd.conf file.

In the httpd.conf you should find a group of directives like this:

```
#LoadModule userdir_module modules/mod_userdir.so
```

```
LoadModule alias_module modules/mod_alias.so
#LoadModule rewrite_module modules/mod_rewrite.so
#LoadModule deflate_module modules/mod_deflate.so
```

Remove the # on the `LoadModule` directive for the `rewrite_module` to uncomment it so that the above now looks like this:

```
#LoadModule userdir_module modules/mod_userdir.so
LoadModule alias_module modules/mod_alias.so
LoadModule rewrite_module modules/mod_rewrite.so
#LoadModule deflate_module modules/mod_deflate.so
```

At the bottom of the `httpd.conf` file add the following:

```
<VirtualHost *:9659>
RewriteEngine On
# Only Allow requests associated with the sdsfViewer REXX
# If not expected URI, then redirect to URL to invoke the REXX
RewriteCond %{REQUEST_URI}
!(/tools/sdsfViewer.rx$|/icons/back.gif$|icons/compressed.gif$|/im
ages/zec12.jpg$)
RewriteRule .*
http://wsc1.washington.ibm.com:9659/tools/sdsfViewer.rx [R,L]
</VirtualHost>
```

The above `RewriteCond` directive is essentially an IF test, testing if the received URI matches any of the values specified.

If they do the request will be processed. If they do not then a redirect request will be sent to the browser which will result in the browser sending a request to invoke the `sdsfViewer REXX`.

Note you would need to modify the `ReWriteRule` to reflect the DNS name of your site. The `RewriteCond` directive would also require modification if you use a different value for the value of the variable `sdsfRexxPath` in the `sdsfViewer.rx` program.

The net result of adding the above is that the IBM HTTP Server can only be used to run the `sdsfViewer REXX` and nothing else.

Also if you used a different image to the supplied `zec12.jpg`, you would need to modify the `RewriteCond` rule accordingly.

3.4.5 Start IHS and test

Save your changes and then start the IHS server and test that you can invoke the `sdsfViewer REXX` and access output on the JES Spool. We entered this URL:

<http://wsc1.washington.ibm.com:9659/tools/sdsfViewer.rx>

and saw this display:



3.5 Configuring IHS for your security requirements

There are many ways you can customize security access controls around the use of this Rexx running in IHS.

We will discuss four possible ways of setting up access control.

3.5.1 Allowing unauthenticated access

After completed the IHS configuration steps described previously you will have the default level of security.

This default level will mean that no authentication is required to invoke the Rexx program. The implication of this is that anyone in your organization who knows the URL can invoke it.

It is important to note however, that while anyone could invoke the URL, they would only be able to view JES Spool output that the userid that the IBM HTTP Server is running under is authorized to view. All access to SDSF and JES spool is done under the userid of the HTTP server, which should be limited to only what is required.

3.5.2 Allowing all authenticated user access

The next level of access control is to require that only users who supply a userid and password validated via the z/OS SAF interface can access the Rexx program but access to SDSF and JES spool remains under the userid of the HTTP server. This arrangement is often referred to as 'client-as-server' access.

To achieve this, make the following modifications to the IHS httpd.conf file.

In the conf file that your IHS is using check that these two directives are present:


```
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authz_user_module modules/mod_authz_user.so
```

In the same area of the conf file add these directives:

```
LoadModule authnz_saf_module modules/mod_authnz_saf.so
LoadModule authz_default_module modules/mod_authz_default.so
```

After adding the above two lines that area of the httpd.conf file will look similar to this:

```
LoadModule authn_file_module modules/mod_authn_file.so
LoadModule authz_user_module modules/mod_authz_user.so

# IBM-Rexx Added next 2 line to support SAF authentication

LoadModule authnz_saf_module modules/mod_authnz_saf.so
LoadModule authz_default_module modules/mod_authz_default.so

#LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule include_module modules/mod_include.so
```

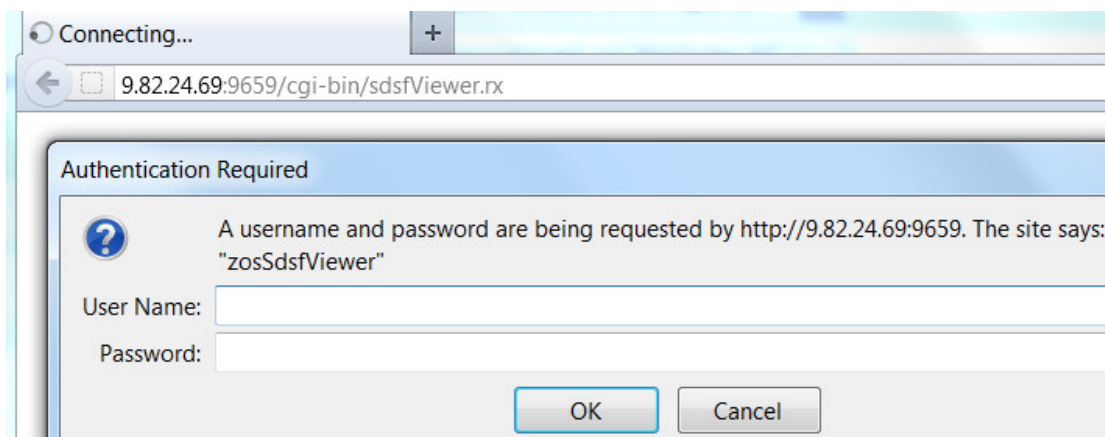
At the bottom of the conf file add these lines:

```
<Location ~ "/(sdsfViewer.rx*)">
  AuthName zosSdsfViewer
  AuthType Basic
  AuthBasicProvider saf
  Require valid-user
  AuthSAFExpiration "EXPIRED! oldpw/newpw/newpw"
  AuthSAFReEnter "Enter new password one more time"
  CharsetSourceEnc IBM-1047
  CharsetDefault ISO8859-1
</Location>
```

The above Location directive tells the IBM HTTP Server that any URL received that ends with sdsfViewer.rx is to have the directives specified within that Location block of directives applied.

You can change the value for the AuthName directive to be whatever you wish, for example it could be: Company ABC z/OS Viewer.

Restart the IHS server to pick up the change. When you use the URL, you will receive a prompt in the browser for a userid and password like this:



Users will need to enter a valid RACF userid and password to continue.

3.5.3 Allowing authenticated user belonging to a group access

The next level of access control is to allow only users who supply a userid and password validated via the z/OS SAF interface **and** who are also a member of a designated group to access spool via the Rexx program.

This setup your organisation to segregate validated users and reduce the scope of STC output they are allowed to view.

This level of controlled access is HTTP server enforced. Please note, at this point we are still in the 'client-as-server' mode, where the SDSF / JES spool access is done under the userid of the HTTP server.

To achieve this, the following modifications to the IHS httpd.conf file are required.

Add the 'Require saf-group' directive to the Location block of directives, which will now look similar to this:

```
<Location ~ "/waslogs*">
  CharsetSourceEnc ISO8859-1
  AddEncoding x-gzip gz tgz
  Header set Content-Disposition "attachment;"
  AuthName zosSdsfViewer
  AuthType Basic
  AuthBasicProvider saf
  Require saf-group E1CELL
  AuthSAFExpiration "EXPIRED! oldpw/newpw/newpw"
  AuthSAFReEnter "Enter new password one more time"
</Location>
```

Note that you cannot use `Require valid-user` when `Require saf-group` is used.

Restart the IHS server. When an end user accesses the URL to run the Rexx program, they will be prompted for their userid and password.

IHS will first validate the userid and password with RACF. Then IHS will validate the userid is connected to the group specified on the directive.

Note multiple group names can be specified on the directive.

3.5.4 Allowing authenticated user access with client credentials

The next level of access control is to configure IHS so that it uses the authenticated and authorised client userid to access the JES Spool rather than the userid of the STC the IHS is running under.

This setup would allow you more granular control over what STC output an end user using this Rexx could view.

Note that while this configuration results in the IHS accessing the JES Spool using the authenticated userid, the STC output they can view will depend on how security has been setup in SDSF and SAF.

To achieve this, the following modifications to the IHS httpd.conf file are required.

Add the 'SAFRunAS' directive to the Location block of directives, which will now look similar to this:

```
<Location ~ "/waslogs*">
  CharsetSourceEnc ISO8859-1
  AddEncoding x-gzip gz tgz
  Header set Content-Disposition "attachment;"
  AuthName zosSdsfViewer
  AuthType Basic
  AuthBasicProvider saf
  Require saf-group E1CFG
  SAFRunAS %%CLIENT%%
  AuthSAFExpiration "EXPIRED! oldpw/newpw/newpw"
  AuthSAFReEnter "Enter new password one more time"
</Location>
```

When IHS receives a request, it validates the RACF userid and password, then when the Rexx program runs and tries to access SDSF, the userid that this access would occur under would be the validated RACF userid.

Note that in the above example we used `Require saf-group`, but you could use `Require valid-user` depending on how you want to control access.

3.5.4.1 *Required SAF definitions*

To use SAFRunAS you need to update RACF or whatever security product you are using to allow use of SAFRunAS.

The RACF commands required are:

```
RDEFINE FACILITY BPX.SERVER UACC(NONE) (if not defined already)
PERMIT BPX.SERVER CLASS(FACILITY) ID(ihs_user_id) ACC(read)
SETROPTS RACLIST(FACILITY) REFRESH
```

If you do not set up RACF rules such as the above, when you try to run the Rexx program in the syslog you will see a message like this:

```
ICH408I USER(WEBSRV1 ) GROUP(SYS1 ) NAME(WEBSRV1 )
BPX.SERVER CL(FACILITY)
INSUFFICIENT ACCESS AUTHORITY
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
```

Note also that the userids of the end users must have an OMVS segment defined.

Further details can be found at:

<http://www-01.ibm.com/support/docview.wss?uid=swg1PM01714>

3.5.5 Use of SSL

The authentication method that we have described is called basic authentication. The result of using this approach is that every request sent from the browser to the IHS server contains the user's userid and password in the HTTP header, in the clear.

If this is a network security concern to your organization, then you could configure the HTTP server to require access over an SSL connection.

Details of how to configure IHS to use SSL can be found in the Infocenter at this location:

http://pic.dhe.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_setupssl.html

3.5.6 Allowing access to show outstanding WTOR messages

The option on the main menu called 'View z/OS Syslog' results in the Rexx program using the SDSF interface to issue a 'd r,l' command to collect any outstanding WTOR messages to display in the browser.

RACF access will need to allow the userid the Rexx program is running under in the IBM HTTP Server access to issue this command otherwise you will get a RACF message like this:

```
ICH408I USER(EDIHS ) GROUP(EDIHSGRP) NAME(#####)
ISFOPER.SYSTEM CL(SDSF )
INSUFFICIENT ACCESS AUTHORITY
FROM ISFOPER.* (G)
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
```

4 Debugging guidance

This chapter provides guidance on issues you may encounter when setting up the supplied Rexx on your z/OS environment.

4.1 Comments on SDSF access

IHS access to SDSF is controlled via the SDSF security controls in the same way that it controls how any batch job or STC is accessing SDSF. Controls around what TSO users can access via SDSF have different control mechanisms in the SDSF security setup than STC or batch jobs do. The IHS STC is not treated as a TSO user.

If when you run this Rexx in IHS on your z/OS LPAR and no STCs are displayed, then you should check the z/OS syslog for RACF violation messages, and the IHS access_log and error_log files.

Depending on how access to SDSF is controlled, no RACF violations messages may appear, and you will need to investigate your SDSF security controls further.

4.1.1 URL to display who you are

To help resolve access issues to SDSF, use this URL:

<http://9.82.24.69:9659/cgi-bin/sdsfViewer.rx?display=whoami>

This will produce output in the browser similar to this:

```
Output from ISFEXEC WHO
```

```
USERID=WEBSRV1  
PROC=REXX  
TERMINAL=  
GRPINDEX=1  
GRPNAME=SDSFPROG  
MVS=z/OS 01.13.00  
JES=z/OS1.13  
SDSF=HQX7780  
ISPF=N/A  
RMF/DA=NOTACC  
SERVER=YES  
SERVERNAME=SDSF  
JESNAME=JES2  
MEMBER=SYSA  
JESTYPE=JES2  
SYSNAME=SYSA  
SYSPLEX=WSCPLEX  
COMM=NOTAVAIL  
COMMX=ENABLED
```

These values can assist with determining why you may not be able to view SDSF output.

4.1.2 The isfsysname variable

One of the variables used by Rexx when accessing SDSF is isfsysname. In the sdsfViewer.rx code you will see this line in two places:

```
isfsysname='*'
```

Setting this variable to '*' allows the Rexx program to view STC output from all LPARs in the z/OS sysplex.

On some z/OS systems this Rexx has been tested on, using the '*' value resulted in no STC output being displayed, because of the way SDSF access controls were setup. Commenting out that line allowed STC output on the same LPAR the IHS was running on to be viewed.

If you want to be able to view STC output from any LPAR in the sysplex, then you may need to resolve issues in your SDSF access setup.

4.2 Some RACF and SDSF security issues you may come across

During development of this Rexx, we came across some RACF access restrictions we need to add or modify rules to overcome.

You may or may not experience these violations.

This line of Rexx:

```
Address SDSF "ISFEXEC DA"
```

Caused this RACF violation message:

```
ICH408I USER(WEBSRV ) GROUP(SYS1) NAME(#####)
ISFCMD.FILTER.PREFIX CL(SDSF )
INSUFFICIENT ACCESS AUTHORITY
FROM ISF*.** (G)
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
```

We modified the corresponding RACF rule to allow the required access.

When the Rexx program in the IHS tried to access the SYSLOG we received this RACF error:

```
ICH408I USER(WEBSRV ) GROUP(SYS1) NAME(#####)
SYSPLEX.OPERLOG CL(LOGSTRM )
INSUFFICIENT ACCESS AUTHORITY
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
```

We modified the corresponding RACF rule to allow the required access.

4.3 When things go wrong

The typical issue that we have come across when setting up this Rexx to run in various z/OS LPARs, is around security access to SDSF.

In the Rexx program code, we test after each call to SDSF and if we receive a non-zero return code then related variables are displayed in the browser. The output would look similar to this:

Prefix:

Well this is not so good, contact your z/OS system programmer on 555-1234 and send them this info

Code location: sv102

Return code: 0

isfmsg:
ISF754I Command 'OWNER *' generated from associated variable ISFOWNER.
ISF754I Command 'PREFIX E2' generated from associated variable ISFPREFIX.
ISF754I Command 'SYSNAME *' generated from associated variable ISFSYSNAME.
ISF754I Command 'SORT JNAME A' generated from associated variable ISFSORT.
ISF767I Request completed.

Output from ISFEXEC WHO

USERID=WEBSRV1
PROC=REXX
TERMINAL=
GRPINDEX=1
GRPNAME=SDSFPROG

The above example was manufactured as we did not have any real error problems on our test environment, which is why the Return Code is zero.

A real error would also display additional error messages.

The string value that appears in the line 'Code location' indicates where in the code this error occurred. For example for the value of sv102 corresponds to this piece of the code:

```
Address SDSF "ISFEXEC "displayCmd
rc2 = rc
if rc2 != 0 then DumpRexxErrMsgs('sv102' rc2)
```

*** End of Document ***