

Miklos.Szigetvari@isis-papyrus.com

# MVSEXP

Mainframe WEB browser.

# Contents

---

Introduction .....	3
A very short introduction.....	3
A longer introduction.....	3
Installation .....	4
System Requirements .....	4
Mainframe .....	4
Browser.....	4
XMIT, Datasets .....	4
The HTTP Server.....	5
httpd.conf.....	6
httpd.envvars.....	6
ISPF interface .....	6
First LOGIN .....	7
The directory structure.....	7
Customization.....	9
Company LOGO.....	9
The customization table.....	9
WLM support .....	9
MXI support .....	9
VTOCLIST support.....	9
REXX compiler .....	9
Health Checker .....	9
Application Performance Analyzer .....	10
IODF .....	10
RACF class .....	10
RACF DSMON .....	10
Fault Analyzer history file.....	10
LookAt URL.....	10
LOGREC logstream .....	10
LOGREC duration.....	10
SMF logstream .....	10
RMF III Data Portal.....	10
Email from Email dest and Email class .....	10
SMS Reports.....	10
MQ Series support.....	11
DB2 support.....	11

SMPE inventories .....	11
Printers.....	11
Translate unprintable .....	11
List size limit .....	11
Break at limit .....	11
Plug-in .....	11
Customization library .....	11
Usage .....	12
Screen format.....	12
Frame Set.....	12
Navigation tree .....	12
User defaults .....	12
Base frame .....	13
A short summary .....	15
Configuration.....	15
Diagnostic .....	15
Performance.....	16
Search and find .....	16
SDSF collection.....	16
Look the SYSLOG .....	16
Check your jobs .....	16
Check your datasets .....	16
SQL SPUFI.....	16
Operator .....	17
Unix USS HFS.....	17
MQ Series .....	18
MVSEXP settings.....	18
Bookshelves on the IBM Library Server .....	18
Job details .....	18
Dataset details.....	20
Active job details .....	20
Miscellaneous .....	22
Performance.....	22
Diagnosis .....	22
Used components.....	23
Mainframe software.....	23

# Introduction

---

## A very short introduction

Download from the XMIT dataset the #DOCMP4 member and open as an MP4 video.

## A longer introduction

For experienced and inexperienced host users, there is a demand to access different parts of the z/OS system via an interface:

- Easy to use, like a WEB browser.
- Can add additional information, not or hardly available in z/OS.
- Can define simple operations (archive or compare etc. etc.) difficult to make on z/OS.

I have tried to make a simple “in house” host explorer via the IBM z/OS HTTP Server and HTML sites. The IBM HTTP Server (5.3) is part of the z/OS installation and contains a number of useful features (RACF authentication, REXX CGI support etc.). A collection of REXX CGI programs realize the different functions. I selected REXX as

- Available everywhere
- Contains a number of string manipulation functions
- Easy to DEBUG or TRACE

The REXX CGI programs are calling the z/OS interface components. Some part is free available (as the SDSF REXX interface) some parts are installed from the CBT tape (as the MXI interface) and some parts are free to download from the IBM (as the SQL REXX interface or the MQ REXX interface) or from other source (WLM interface from YCOS software). If there was no any interface, I wrote simple C/C++ programs to access these parts of the system:

- Catalog information
- Log stream's
- ISPF table etc.

The results are presented as dynamic HTML pages, with the help of Java scripts, available in the internet open source communities. I'm using some browser plug-ins (Adobe Acrobat for PDF, ISIS AFP viewer for AFP or Notepad++) and links to the IBM “LookAt” or to the IBM Library Server.

I think the application is extendable in all directions. It is connecting to, some part of the IBM software (RMF Data Portal), excellent to access from the WEB, but avoids example the CICS; as the IBM has the super CICS Explorer.

## Warning

**This is not a replacement for TSO or SDSF ISPF or any other IBM or non IBM product. We are using this in the ISIS Papyrus Software for a while, with a small number of users. It has never been intensively tested for performance or security.**

Miklos Szigetvari  
Januar 2013  
<miklos.szigetvari@isis-papyrus.com>

# Installation

---

## System Requirements

### Mainframe

1. z/OS 1.10 or later with the HTTP Server 5.3 (we are using this since z/OS 1.1.0)
2. Disk space about 1500 tracks for the complete restore.
3. IBM C++ compiler, if you want to recompile the C++ source code.
4. Full function OMVS
5. z9 machine or higher(no any hardware dependency, but the C++ code complied with ARCH(7), target architecture z9 or higher, compiler option).
6. You can configure the usage of the other software components (DB2, MQ Series, Fault Analyzer etc.)

### Browser

We are using Windows XP and Windows 7:

- Everything works with Mozilla Firefox.
- We have tested also Google Chrome, but got sometimes an "ERR\_EMPTY\_RESPONSE" from the browser. Currently not sure if it is an error in Google Chrome or my Pop-Up contains something invalid.
- Tested with Microsoft Internet Explorer, after ignoring some Java Script errors, works except the Pop-Up windows.

## XMIT, Datasets

The XMIT partitioned dataset contains several installation datasets in XMIT format:

1. ADATA assembler compiler ADATA output.

As the largest part of the code is C++ we need to generate the C structures for a number of system control blocks. We are using for this, the DSECT2C utility from the C++ compiler library. The input for this generation is the assembler compiler ADATA dataset.

As the size of the ADATA is very big, we left this dataset intentionally empty.

2. ASM assembler library.

Beside some macros, it contains:

- \$\$ASMIF macro / module, a general interface for C++ modules to system functions.(Logstream access , VTOC access etc.).
- REXXRACF program.

3. COMPLIST library

The compiler lists from the last compilation.

4. CPP library

It is the C++ module library. As we are using here German code page, the routines are in IBM273 code page, and the compiler options also refer to this code page.

5. HPP header files

The headers file definitions for the different C++ external functions.

This HPP library contains the different generated structures from the system control blocks.

Members for CVT, ASCB, RB etc. (see ADATA above).

6. JOBS

Library we used here to compile/link the project.

7. LOADLIB

- DUMPFIL to print a file in EBCDIC, ASCII text and hexadecimal format. It is an alias of the MVSTOOL program
- MVSTOOL the main tool utility entry point
- REXXDSMF read the SMF logstream and generate a REXX stem from the following SMF records:
  - 30 termination record
  - 14-15 dataset close record
  - 92 USS file access record
- REXXGIM call the SMPE “gimapi” interface, and presents the result as a REXX stem.
- REXXJCL process the SDSF JESJCL dataset to extract dataset names, program names.
- REXXLIST from a dataset mask generate a REXX stem with the different dataset attributes.
- REXXRACF call RACROUTE to check RACF access.
- REXX4TAB generate a REXX stem from an ISPF table.
- DB2WWWX the REXX SQL interface module from the IBM.

The different functions are using the modules from the library, so you **should have to define this library in the HTTP server “httpd.envvars” dataset as STEPLIB or put the modules to the linklist.**

## 9. OBJECT

The object library

## 10. OUT

It contains the WLM definition outputs and AFP, PDF samples.

## 11. PROC

The compile/link JCL procedures and compiler options.

## 12. SAMPLIB

It is a sample library with customization and installation jobs. We are using here as high level qualifier “ESA.PRJ.CBTEXP” name. With the CHANGEAL sample job you can change this default to the proper SAMPLIB name.

## 13. WLMDEF

This dataset contains ISPF tables extracted from WLM.

## 14 ZFSDUMP

The HTTP server and all the necessary REXX exec java scripts etc. are in a ZFS dataset. We have dumped here with the standard ADRDSSU utility, you have to restore this ZFS dataset and mount for the USS.

## 15. XMIT

This dataset contains the datasets listed above in XMIT format and the current document in text, PDF document format and an MP4 video. It also contains the INSTALL member to restore the installation material. If you don’t intend to recompile, it is enough to restore the SAMPLIB, OUT and ZFSDUMP datasets.

## The HTTP Server

Before you can start the HTTP server, you should have to restore the ZFS dataset, and mount this ZFS file to a proper mount point:

```

EDIT          ESA.PRJ.CBTEXP.SAMPLIB(MOUNT) - 01.00                               Member MOUN
Command ==>                                     Scroll ==
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 MOUNT FILESYSTEM('MVSEXP.V1R1M0.ZFS')
000002 MOUNTPPOINT('/u/mvsexp')
000003 TYPE(ZFS)
000004 MODE(RDWR)

```

In the following, we assume that the ZFS file restored and mounted in the “/u/mvsexp” mount point. The MVSEXP member in the SAMPLIB is the job, to start the HTTP server. You have to change the port number.

The server will use the “/u/mvsexp/httpd.conf” configuration file and the “/u/mvsexp/httpd.envvars” environment variable file:

### httpd.conf

This is the modified HTTP server configuration file:

- The path name has changed from “/usr/lpp/internet” to “/u/mvsexp”.
- The “Welcome” page has changed to “htdemo.html”.
- The protection option has changed to use SAF (RACF) authentication.

```
#
Protection IMW_Admin {
    ServerId      IMWEBSRV_Administration
    AuthType      Basic
    PasswdFile    %SAP%
    UserID        %CLIENT%
    Mask          All
}
Protection CGI_Admin {
    ServerId      CGI_Administration
    AuthType      Basic
    PasswdFile    %SAP%
    UserID        %CLIENT%
    Mask          All
}

Protect /admin-bin/* IMW_Admin
Protect /Docs/admin-bin/* IMW_Admin
Protect /reports/* IMW_Admin
Protect /Usage* IMW_Admin
Protect /cgi-bin/* CGI_Admin
```

- A new dataset type has added for Notepad++.
- From the newest httpd.conf the JavaScript type definition is missing, we use the JavaScript’s as ASCII.

### httpd.envvars

The httpd.envvars contains the proper STEPLIB, the ISPF CGI interface environment variables and the extended PATH and LIBPATH definitions:

```
EDIT /u/mvsexp/httpd.envvars
Command ==>
***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
000001 SHELL=/bin/sh
000002 OI2=EST5ED1
000003 TZ=MSEZ-1MSEZ-2,M3.5.0/02:00:00,M10.5.0/03:00:00
000004 LANG=C
000005 LC_ALL=en_US.UTF-8
000006 NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/ldap/lib/nls/msg/%L/%N
000007 GSKV3CACHE SIZE=1024
000008 GSKV2CACHE SIZE=512
000009 BPX_SHAREAS=NO
000010 HTTP_CONNECTION_PERCENTAGES=12,25,30,10,20
000011 HTTP_CONNECTION_SIZES=192,584,1544,7168,33728
000012 HTTP_MEMPOOL=32320
000013 CGI_ISPWORK=/var/ispf
000014 CGI_ISPCONF=/etc/ispf
000015 STEPLIB=esa.prj.cbtextp.loadlib
000016 PATH=/usr/lpp/internet/sbin:/usr/bin:/usr/sbin:/usr/lpp/internet/bin:/usr/lpp/ldap/bin:/usr/
000017 LIBPATH=/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/ldap/lib:/usr/lib:/u/mvsexp/server_root/cgi-bin/run
```

### ISPF interface

We are using the “TSO ISPF Client Gateway” interface to call some ISPF functions. The CGI\_ISPCONF variable points to a directory, and it contains the “ISPF.conf” member:

```

BROWSE      /MVID/etc/ispf/ISPF.conf                      Line 00000025 Col 001 080
Command ==>                                         Scroll ==> CSR
*****

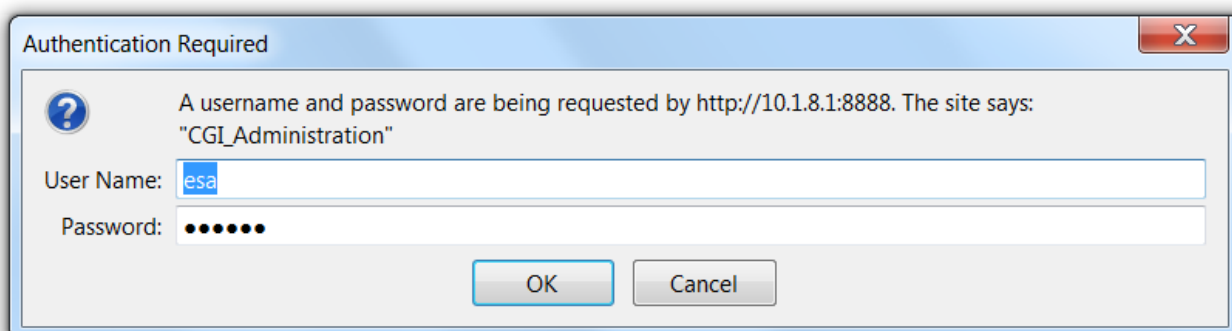
* REQUIRED:
* Below is the minimum requirements for ISPF allocation.
* Change the default ISPF dataset names below to match your host site.
* Add additional dsn concatenations on same line and separate by comma.
* Order of datasets listed is search order in concatenation.

sysproc=ISP.SISPCLIB,sys9.maint.ispplib,sys9.SISPCLIB,sys1.SACBCNTL,sys1.DGTCLIB
ispmlib=ISP.SISPMENU,sys9.maint.ispplib,sys9.SISPMENU,sys1.DGTMLIB
isptlib=ISP.SISPTENU,sys9.maint.ispplib,sys9.SISPTENU,sys1.DGTTLIB
ispplib=ISP.SISPPENU,sys9.maint.isptlib,sys9.SISPPENU,sys1.dgtplib
ispplib=ISP.SISPSLIB,sys9.maint.ispslib,sys9.SISPSLIB,sys1.dgtslib
ispllib=ISP.SISPLoad

```

## First LOGIN

After the HTTP server has started, you can select in your browser the IP address or hostname and port, and you will get the authentication Pop Up window:



**You should have to enter here your RACF (SAF) user name and password, twice.** The authentication request can't cope with password changes; you have to change your password via TSO. After the successful logon every activity will run under the actual RACF (SAF) userid.

## The directory structure

Under the "/u/mvsexp" mount point the "server\_root" directory is the main directory for the HTTP server. Under this directory the subdirectory structure is the following:

- "admin-bin" cut out from the original
- "img-bin" as original
- "icons" as original
- "fcgi-bin" as original
- "labels" as original
- "Counters" as original
- "Admin" cut out from the original
- "Docs" cut out from the original
- "cgi-bin" for the CGI programs. Here in the "run" subdirectory, you will find the REXX execs, realizing the mainframe explorer. These REXX execs are called from this directory, they are editable text files, with the following naming conventions:
  - The called REXX execs have "sh" extension , so examples "jesjob.sh"
  - The REXX source files have "rex" extension , so examples "jesjob.sh.rex"
  - The inner subroutines and functions have no extension, so example "querystr"
  - The generated/processed REXX execs have "inlinere" extension
  - The compiled REXX execs have "cexec" extension.



If you intend to change some REXX function, you can edit the proper file with “rex” extension (We are using the TSO ISPF “oedit” USS editor). From the edited file, you have to generate the executable, callable REXX script. The “PROF” REXX exec will do this processing. So if you call as

“PROF jesjob.sh”

it will take the “jesjob.sh.rexx”, copies all the included subroutines and generates the “jesjob.sh.inlinerexx”. If the REXX compiler installed compiles this “inlinerexx” to “cexec”, else copies this file to “jesjob.sh”.

To generate all the modules, you can call the “DIRPROF” REXX exec.

- “pub” the main public directory for the HTTP server.

Here is the “htdemo.html” welcome page; it contains some more html pages, and the following important subdirectories:

- “js” subdirectory for the Java Scripts. Currently the Java Scripts are coded in ASCII.
- “NewStyle” subdirectory for the Style Sheet definitions.
- “icons” subdirectory for the used icons.

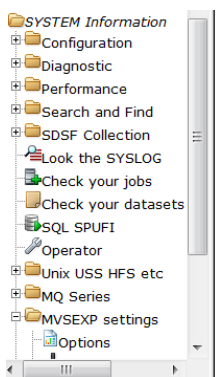
## Customization






### Company LOGO

You can change the default company LOGO in the htdemo.html site, as this site is defined as “welcome” page. As default you will see the Isis Papyrus Software company logo.

### The customization table

The global options are in the “/u/mvsexp/server\_root/cgi-bin/run/OPTIONS” file. Every function reads this option file, and set the proper global options. As initial default, we set off all the additional options. In the “Isis.OPTIONS” file you can see the actual options by Isis. From the navigation tree, you can select the “MVSEXP setting” and the “Options”:



This list conatins the used exteral tools , software components and the global default values			
Option	Link	Cutomize	Value
WLM support		SHOWWLMJ	NO
MXI support		MXISERV	NO
VTOCLIST support			NO
REXX compiler			NO
Health Checker			YES

If the RACF class protection is active, the customization table is protected via the RACF resource class “Customize” profile, only a user, has UPDATE access, can change this values,. In the following, we describe shortly the customization options:

### WLM support

For the WLM support, we are using the SHOWWLM REXX script from Yves Colliard, YCOS Software Gmbh. In the SAMPLIB SHOWWLM REXX script you need to set the target directory:

```
000148 /*
000149 /* WLM USS - OMVS - Output Directory          */
000150 /* outuss                                     */
000151 /* if blank no USS Data Set                  */
000152 OUTUSS="/u/mvsexp/server_root/pub/"
000153 /* */
```

### MXI support

We are using the MXI Server from Rob Scott, Rocket Software, from the CBT tape 408. If you have downloaded and installed the MXI Server, you can use, via a small C/C++ program “mvsstrserv”. This will communicate with the MXI Server via TCP/IP on the MXI Server default port 8001.

### VTOCLIST support

We are using the VTOCLIST program from John Kalinich and Peter Havercan, from the CBT tape 343.

### REXX compiler

If you have the REXX compiler installed.

### Health Checker

You can activate the interface to Health Checker.

## Application Performance Analyzer

The APA (Application Performance Analyzer) tool from the IBM is called via the CGI ISPF interface. If this interface is active, you can call the APA. Enter here the high level qualifier(s) of the measurement datasets generated by the APA. If APA generates “APA.V9R1M0.MIKLOS.R0138.MYEXP.SF” etc., enter here APA.V9R1M0 .

## IODF

There is a very nice HTML, PERL/CGI scripts written by Mark Naughton in CBT tape 736. To use this, you need:

- The PERL interpreter
- Run the IODFGEN job, to download the extracted IODF into a subdirectory of the HTTP server.
- In the iodf.cgi PERL script you need to customize:

```
#####
# ONLY CHANGE THESE USER PARAMETERS
#####
$data_directory = "/u/mvsexp/server_root/cgi-bin/iodf";
$cgi_directory  = "/cgi-bin/run";
$script_name    = "iodf.cgi";
$title          = "ISIS Information Systems:";
```

## RACF class

This is the name of the RACF CDT class. We can use this class, during the build of the navigation tree. If the user has READ access to the proper entry in the navigation tree, only in this case he will see it.

## RACF DSMON

We generate RACF reports via the RACF Data Security Monitor feature. We call the “ICHDSM00” RACF utility from a REXX script ICMDSM00 from the SAMPLIB.

## Fault Analyzer history file

You specify here the name of the Fault Analyzer history file or “NO”.

## LookAt URL

This is the first part of the URL for the IBM LookAt WEB site.

## LOGREC logstream

The LOGREC logstream name used in the LOGREC reports, or “NO” if this option is inactive.

## LOGREC duration

The duration interval in hours, how long we retrieve the logrec records from the logstream.

## SMF logstream

This is the name of the SMF logstream, to retrieve the actual SMF records.

## RMF III Data Portal

Specify the IP address and port of the active RMF Data Portal, or “NO” if you don’t want to use this.

## Email from Email dest and Email class

We can use the IBM SMTP server to send Emails. For this, we need the node name, the SMTP server destination name and the output class.

## SMS Reports

For the different SMS reports, we use the “naviquest” query. Here the query job runs once a day and the reports are generated in a preallocated dataset. Specify the dataset name or “NO”, if this option is inactive.

## **MQ Series support**

For the REXX MQ interface, we need the name(s) of the queue managers to connect.

## **DB2 support**

For the DB2 SQL support we need the name(s) of the different DB2 subsystems.

## **SMPE inventories**

If you specify here “NO”, than there is no SMPE support, but if you specify here a number “n”, you can specify “n” SMPE inventories bellow, the CSI name, the GLOBAL, TARGET and DISTRIB zone names.

## **Printers**

You can define here some printers with symbolic name and output class.

## **Translate unprintable**

Specify “YES”, if you want to translate the unprintable characters from the SDSF spool list to blanks.

## **List size limit**

The size of the generated list and tables can be very large; we have a limit on the generated lines. If a list has reached this limit you can break the list or you will get a warning about this.

## **Break at limit**

Break the list at the limit or just send a warning.

## **Plug-in**

We can use as browser plug-in application:

- ADOBE PDF Acrobat reader to show PDF files
- ISIS Papyrus AFP viewer
- NOTEPAD++ editor

## **Customization library**

This is the dataset name of the library for the customization jobs.

# Usage

## Screen format

### Frame Set

We use a HTML frame set. The “treeframe” contains the navigation tree, the “downframe” the actual user defaults and the “basefrm” the dynamic HTML sites:

The screenshot shows the MVSEXP web interface. On the left is a navigation tree with folders like SYSTEM Information, Configuration, Diagnostic, Performance, Search and Find, SDSF Collection, Look the SYSLOG, Check your jobs, Check your datasets, SQL SPUI, Operator, Unix USS HFS etc, MQ Series, MVSEXP settings, and Options. Below the tree are input fields for User (ESA), JOBS (ESA\*), OWNER (\*), Datasets (ESA.USER.\*\*), LOG minutes (30), Home (/u/mvseexp/ser), Style (grayscale), and DB2 (DSNA). At the bottom left, it says Machine:2096 and System:MV1D. The main area displays a table titled "DSNAME:ESA.USER.\*\*" with "42 entries". The table has columns: DSNNAME, DSORG, CREATED, EXPIRE, REF, VOLSER, TRK, and ATTRIBUTES. It lists various user datasets like ESA.USER.AAA, ESA.USER.ASM, ESA.USER.ASM1, etc., with their respective attributes.

DSNAME	DSORG	CREATED	EXPIRE	REF	VOLSER	TRK	ATTRIBUTES
ESA.USER.AAA					MIGRAT		
ESA.USER.ASM	PO	2003/02/03	1900/01/00	2013/02/01	USERL2	342	PO FB 80 32720
ESA.USER.ASM1					MIGRAT		
ESA.USER.COMPLIST					MIGRAT		
ESA.USER.CP					MIGRAT		
ESA.USER.CPP					MIGRAT		
ESA.USER.DBG					MIGRAT		
ESA.USER.DBGLIB					MIGRAT		
ESA.USER.DLL					MIGRAT		
ESA.USER.EATTR2					MIGRAT		
ESA.USER.EATTR3	????	2012/01/16	1900/01/00	2012/03/29	USER0F	15	???? FB 0 0
ESA.USER.FBCPP					MIGRAT		
ESA.USER.GENCPP					MIGRAT		
ESA.USER.HISTORY					MIGRAT		
ESA.USER.HPP					MIGRAT		
ESA.USER.JOB	PO	2007/09/17	1900/01/00	2013/02/01	USERL0	95	PO FB 80 32720
ESA.USER.JOBVB	PO	2005/01/18	1900/01/00	2013/02/01	USER09	41	PO VB 27994 27998
ESA.USER.LISTOPT					MIGRAT		
ESA.USER.LOAD					MIGRAT		
ESA.USER.OBJECT					MIGRAT		
ESA.USER.PDSE					MIGRAT		
ESA.USER.PSMF.ASM					MIGRAT		

### Navigation tree

In the navigation tree, you can select different folders, and in the folders you call the different options. The content of the navigation tree is built from the “buildtree.txt” file. This file contains the folder definitions, the link titles and targets, and the used icons. You can edit/modify the “buildtree.txt” file to add or remove entries.

If the RACF CDT class is active and defined, during the building of the navigation tree, we check if the user has READ access to the proper profile in the RACF dynamic class.

### User defaults

Here you can define or change the default values for a user. This default values will be used in job selection or dataset selection or SQL query operation etc. The actual default values are stored in the “userid.defaults” file. Beside this default values, the user can select a Style Sheet. We have a number of predefined Style Sheet’s:

User:	ESA	DSNAME	DSORG	CREATED	EXPIRE	REF	VOLSER	TRK	ATTRIBUTES
JOB:	ESA*	ESA.USER.AAA					MIGRAT		
OWNER:	*	ESA.USER.ASM	PO	2003/02/03	1900/01/00	2013/02/01	USERL2	342	PO FB 80 32720
Datasets:	ESA.USER	ESA.USER.ASM1					MIGRAT		
LOG minutes:	30	ESA.USER.COMPLIST					MIGRAT		
Home:	/u/mvsexp/s	ESA.USER.CP					MIGRAT		
Style:	blueonblue	ESA.USER.CPP					MIGRAT		
		ESA.USER.DBG					MIGRAT		

User:	ESA	DSNAME	DSORG	CREATED	EXPIRE	REF	VOLSER	TRK	ATTRIBUTES
JOB:	ESA*	ESA.USER.AAA					MIGRAT		
OWNER:	*	ESA.USER.ASM	PO	2003/02/03	1900/01/00	2013/02/01	USERL2	342	PO FB 80 32720
Datasets:	ESA.USER.**	ESA.USER.ASM1					MIGRAT		
LOG minutes:	30	ESA.USER.COMPLIST					MIGRAT		
Home:	/u/mvsexp/ser	ESA.USER.CP					MIGRAT		
Style:	goldenstyle	ESA.USER.CPP					MIGRAT		
DB2:	DSNA	ESA.USER.DBG					MIGRAT		
		ESA.USER.DBGLIB					MIGRAT		
		ESA.USER.DLL					MIGRAT		

## Base frame

This frame contains the dynamic HTML sites generated by the CGI programs. The frame can contain some common items:

## Tables

We present the replies with the help of HTML tables. You can sort the tables via column names.

## Hyperlinks

A hyperlink can lead to some additional info or can call utility functions.

## Speech Bubbles

If you leave your cursor on a column title, you can get a "Speech bubble" to explain the content of the column. The speech bubble text are defined in the "/u/mvsexp/server\_root/pub/speechdata.txt" text file, you can easily extend or modify this text.

DSNAME	DSORG	CREATED	EXPIRE	REF	VOLSER	TRK	ATTRIBUTES
ESA.USER.AAA					MIGRAT		
ESA.USER.ASM	PO	2003/02/03	1900/01/00	2013/02/01	USERL2	342	PO FB 80 32720
ESA.USER.ASM1					MIGRAT		
ESA.USER.COMPLIST					MIGRAT		
ESA.USER.CP					MIGRAT		
ESA.USER.CPP					MIGRAT		
ESA.USER.DBG					MIGRAT		
ESA.USER.DBGLIB					MIGRAT		
ESA.USER.DLL					MIGRAT		
ESA.USER.EATTR2					MIGRAT		
ESA.USER.EATTR3					MIGRAT		
ESA.USER.FBCPP					MIGRAT		
ESA.USER.GENCPP					MIGRAT		
ESA.USER.HISTORY					MIGRAT		
ESA.USER.HPP					MIGRAT		
ESA.USER.JOB	PO	2007/09/17	1900/01/00	2013/02/01	USERL0	95	PO FB 80 32720
ESA.USER.JOBVB	PO	2005/01/18	1900/01/00	2013/02/01	USER09	41	PO VB 27994 27998
ESA.USER.LISTOPT					MIGRAT		
ESA.USER.LOAD					MIGRAT		
ESA.USER.OBJECT					MIGRAT		
ESA.USER.PDSE					MIGRAT		
ESA.USER.PSMF.ASM					MIGRAT		
ESA.USER.PSMF.COMPLIST					MIGRAT		
ESA.USER.PSMF.JOBS					MIGRAT		
ESA.USER.PSMF.LOADLIB					MIGRAT		

### Pop-Up Windows

We generate “Pop-Up” windows to select the different actions. You have to close currently the “Pop-Up” windows:

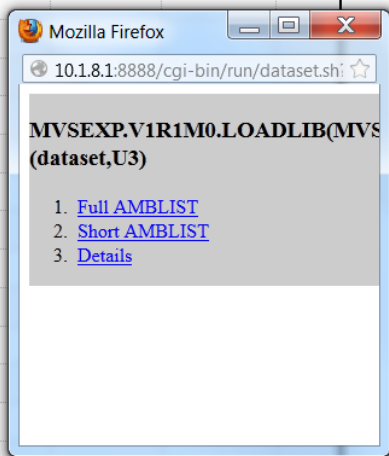
**DSNAME: MVSEXP.V1R1M0.LOADLIB**

**13 members**

DSNAME	DSORG	CREATED	EXPIRE	REF	VOLSER	TRK	ATTRIBUTES
MVSEXP.V1R1M0.LOADLIB	PO	2013/02/04	1900/01/00	2013/02/05	USER0D	92	PO U 0 32760

MEMBER	USER	CREATED	MODIFIED	SIZE
DUMPFIL				
MVSTOOL				
REXXDSI				
REXXDSMF				
REXXENFS				
REXXGIM				
REXXJCL				
REXXLIST				
REXXRACF				
REXXRMF2				
REXXSMF				
REXXVSM				
REXX4TAB				



### Command Stacks

In different sites (Operator commands, SQL query, etc.) we use a command stack to store the used commands. Every user has a unique command stack, “userid.cmdstack”:

**Enter an SQL statement for subsystem DSNA or select from the list**

Submit to DB2 Edit Delete Reset

1

Position: Ln 1, Ch 1      Total: Ln 1, Ch 0

☒ Toggle editor

Commnad

☐ SELECT \* FROM PHFTTEST.ISIS\_DOCAR

☐ SELECT \* PHFTTEST.ISIS\_DOCAR

☐ SELECT \* FROM DSN8910.EMP WHERE WORKDEPT = 'A00'

☐ SELECT \* FROM DSN8910.EMP

☐ SELECT \* FROM SYSIBM.SYSTABLES

☐ SELECT \* FROM SYSIBM.SYSPLAN

☐ SELECT \* FROM SYSIBM.SYSPACKAGE

☐ SELECT \* FROM SYSIBM.SYSCOLUMNS

### Editable text area

The editable text area is an input text area; with a tiny editor you can edit the text.

## A short summary

In the following, we describe shortly the main properties of the different elements in the navigation tree.

### Configuration

#### Software

It contains several links to the MXI software queries, and a link to the WLM definitions.

#### Hardware

It has a link to the actual IODF, and several MXI hardware queries.

#### Defined users, RACF

It has links to several MXI RACF queries, and it contains a list of all defined RACF users. The “RACF Reports” folder contains several links to the RACF “Data Security Monitor” reports.

#### SMP/E

You can make queries for the primary (first) SMP/E inventory GLOBAL, TARGET and DLIB zones via “GIMSMP” list commands. The SMP/E GIMAPI interface was called by the “REXXGIM” utility.

#### DFSMS HSM

This folder contains a link to the HSM command stack, here with “QUERY” or “HLIST” commands you can query the HSM configuration.

#### DFSMS

This folder contains links to the pre-generated “naviquest” reports for SMS.

#### Network

This folder has links to the “netstat” and “nfsstat” commands. In the “netstat” query, you can specify filters for the proper query (for example IP address or application name) or you can use the “netstat” options without filters. The “nfsstat” command asks for all the mounted “nfs” files.

#### HTTP Server

You can query the different statistics, maintained by the HTTP server.

### Diagnostic

#### LOGREC summary

If you configured a LOGREC logstream, a script will make a query for the LOGREC via the “IFCEREPI” logrec utility, to list all the software records, for the last “duration” period. From the list you can see the logrec software records in details.

#### FAULT Analyzer

If you configured the Fault Analyzer history file name, you can call the Fault Analyzer utility “IDIUTIL” to get a list of the history file. In the history list, you can select a “Fault id”, to get the details of the event. The list is processed by the “IDIDA” Fault Analyzer program. To work this program properly you need an ISPF environment and in this environment the “RUNDA1” REXX exec will call the “IDIDA” utility, to format the Fault Analyzer report.

#### Health Checker

If the Health Checker is active, you can list all the active checks with severity code not zero. From the list you can select a check to show the detailed report or activate, deactivate or refresh the check.

#### MXI reports

You can call the System Exception or DAE report from MXI.



## Performance

### Active jobs

This is similar as SDSF DA report.

### RMF III Data Portal

If the RMF Data Portal is active you can link to the actual RMF Data Portal home.

### MXI Reports

You can call the “Active Address Space” “Real Storage Usage” and the ”LLA Fetch Statistic” MXI reports from here.

### Search and find

You can search the content of a group of datasets via the “gnugrep” USS utility. The utility is called from the “\$(SEVER\_ROOT)/pub/bin” directory. It can search for strings, defined by regular expression, in a group of datasets defined by a dataset mask.

The “find member” functions are searching for a member pattern in a group of datasets. For the “LINKLIST” “PARMLIB” “PROCLIB” or “SYSEXEC” search, the “fixlocate.sh” script opens the proper USS file in the current directory, and reads the actual dataset list from this file.

The “compare” functions can compare two datasets, via a Java Script.

### SDSF collection

Like the SDSF “ST” and “DA” option:

- In the “ST” site, you can purge, dump or resubmit the job, you can go the job details, link to the jobs owners RACF details and if the job active you can get a Pop-Up window for the different actions, if the job has finished you can go to examine the job events.
- From the “DA” site, you can link to different MXI reports or WLM definitions, or examine the job events.

### Look the SYSLOG

From the current time, you can look into the SYSLOG (operator log) and link to the message descriptions in the LOOKAT web site or link to the job details. You can change the time frame via the “LOG minutes” user default value.

### Check your jobs

You can select from the JES SPOOL jobs via a job name prefix and a user name. The selected jobs are listed in a HTML table. From this table you have a link:

- To the job details (see later).
- To the user’s RACF definition,
- If the job is active, link to the “active job” Pop-Up window or if inactive to the “event log”.

### Check your datasets

From the dataset name pattern defined in the “User Defaults” frame the “REXXLIST” utility generates a REXX stem with the dataset attributes. From the generated HTML table, you can link:

- To the dataset details.
- To the volume list.
- To the member list, if it is a PO datasets.

### SQL SPUFI

You can make SQL queries in the DB2 subsystem, defined in the “User Defaults”. The SQL command stack will be called, to list the SQL commands already issued. You can:

- Execute the SQL command.
- Edit the SQL statement.
- Delete from the SQL command stack.

The SQL commands will be executed with the help of the DB2WWWX SQL REXX interface utility from the IBM. The results are presented as HTML tables.

### Operator

You get from here the command stack for the operator commands. You can:

- Execute the selected command.
- Edit and execute.
- Delete from the command stack.

### Unix USS HFS

#### Your home directory

You can list your home directory defined in the “User Defaults”.

In the HTML table you see:

- Entry type, directory or regular file etc.
- The entry name and here you can link to the file details, or a directory list.
- Size in K bytes
- Owner
- Creation date
- Modification date
- Content. We try to “guess” the file content via the USS “file” command. This command is using “magic” file to determine the file type from the first few bytes.

#### Active processes

This is an active process list, via the MXI interface.

#### Jes process list

The SDSF “PS” (process list) command, selected via the “User Defaults”.

You can call here:

- MXI PID command, to display the process details.
- MXI DA command, to display the address space details.
- Display the process owner details.
- Issue a “D OMVS,PID=xxxx” operator command.

#### HFS mounted file systems

You can list the mounted file system via MXI.

#### ZFS aggregates

You get a HTML tables with all the ZFS files.

#### ZFS aggregates via RMF III

This will link to the RMF Data Portal ZFS aggregate list

#### ZFS summary via RMF III

This is the ZFS summary report from the RMF Data Portal.

## MQ Series

The MQ Series interface will use the MQ Series REXX support function package (MA95). The actual queue manager name is taken from the “User Defaults”.

### Queue manager

Get a list of the queue manager attributes.

### Channels

Get a HTML table with the defined channels, and you can link from here to the channels details.

### Queue's

This is a HTML table with the defined queues. The name column will link to the queue definition details, and if the queue is not empty, i.e. the MSGNO column not zero, the MSGNO column will link to the queue list. In the queue list you will see the message attributes and if you go further, you will see the complete message.

## MVSEXP settings

This part contains information about the current MVSEXP defaults and about the actual environment.

### Options

If you authorized you can change the global options, see the “Customization” section above.

### IBM REXX

You will get a HTML site, which describes the different REXX interfaces, from the IBM, to DB2 OTMA IMS.

### small White Paper

This is a small “pdf” document, to put together the main properties of the MVSEXP.

### Environment

These are the actual environment variables, used by the HTTP server. From these values we are using:

- QUERY\_STRING to get the actual query string.
- HTTP\_REFERER to get the caller of the actual site. We would like to avoid, to call some HTML site outside the “Navigation Tree”. We check the HTTP\_REFERER, and go further if it is not empty. For the Microsoft Internet Explorer this field is empty, so we avoid this check with Internet Explorer.
- FSCP and NETCP. We use the NETCP (network code page) and FSCP (local HTTP server code page) values to convert properly the HTTP QUERY\_STRING. The QUERY\_STRING is already converted to EBCDIC, before the REXX CGI program starts. But the special characters, sent by the browser, are remaining in hexadecimal encoded in the NETCP code page. We use the NETCP and FSCP values to convert back the special characters via the “iconv” USS utility.

## Bookshelves on the IBM Library Server

This is an arbitrary collection of Library Server bookshelves.

### Job details

If you link to the job details:

- You will get a HTML table with the JES “JESMSGLOG” dataset. This list will contain links to the LookAt message descriptions.
- A HTML table with all the SYSOUT datasets in the JES SPOOL.

From this list:

Via the DDNAME column you can go to the SYSOUT actions for a selected output.

Via the DSNAMES column you can open the selected output.

If the dataset is a PAGE mode data, we assume that it is an AFP data stream and we will try to open with the Isis Papyrus AFP viewer plug-in.

If the dataset has the FORMS JES parameter as PDF, we try to open with the Adobe Acrobat Reader PDF plug-in.

If the dataset has the FORMS JES parameter as TOC, we try to call the TOC Java Script, and it will generate a “Table of Content” from every line with the “<H2>” HTML tag.

In every other case we send the data to the browser, but first we examine the program name and DDNAME, produced this output. If the program name and DDNAME is in our list, we insert references to the LookAt site for the program messages:

Via the FORM column we can call the NotePad++ plug-in program to present the output in an editable format.

- The third part is an “Editable Area” with the actual JCL. You can change the JCL and resubmit the job.

In the top of the site you can find two links.

#### Event log

In the event log, we collect all the relevant information about the actual job:

- Information from the JESMSGGLG
- The SYSOUT dataset creation events
- SMF 30 (termination) SMF 14-15 (dataset close) SMF 92 USS file events from the SMF logstream

09:35:28:55	JOB09059	-ESASMC01 PLICOMP PLI	00	150	337	.00	.00	.0	451	0	0	0	1	0	#09352855
09:35:39:56	JOB09059	-ESASMC01 PLILINK LKED	04	454	879	.00	.00	.1	1343	0	0	0	0	0	#09353956
09:35:40:01	JOB09059	-ESASMC01 PLIRUN	00	62	222	.00	.00	.0	139	0	0	0	0	0	#09354001
09:35:44:19	JOB09059	-ESASMC01 COICOB COBOL	00	491	852	.00	.00	.0	647	0	0	0	53	0	#09354419
09:35:44:74	JOB09059	-ESASMC01 COICOB LKED	00	161	277	.00	.00	.0	536	0	0	0	2	0	#09354474
09:35:45:60	JOB09059	-ESASMC01 COICOB GO	00	170	695	.00	.00	.0	229	0	0	0	0	0	#09354560
09:35:45:60	JOB09059	-ESASMC01 ENDED. NAME=ESA	TOTAL TCB CPU TIME= .01 TOTAL ELAPSED TIME= 2.3 #09354560												
09:35:45:999	JOB09059	\$HASP395 ESASMC01 ENDED													
09:33:18:00	JES2	JES	ESA.ESASMC01.JOB09059.D0000003.JESJCL	Ddname:JESJCL	Records:308										
09:33:18:00	JES2	JES	ESA.ESASMC01.JOB09059.D0000004.JESYSMSG	Ddname:JESYSMSG	Records:313										
09:35:00:00	SCGICPP.COMPILE	JES	ESA.ESASMC01.JOB09059.D0000103.?	Ddname:SYSOUT	Records:7										
09:35:18:00	SCGICPP.LKED	JES	ESA.ESASMC01.JOB09059.D0000107.?	Ddname:SYSOUT	Records:69										
09:35:26:00	SCGICPP.LKED	JES	ESA.ESASMC01.JOB09059.D0000109.?	Ddname:SYSPRINT	Records:208										
09:35:27:00	RUNCPP.	JES	ESA.ESASMC01.JOB09059.D0000110.?	Ddname:SYSPRINT	Records:5										
09:35:27:00	PLICOMP.PLI	JES	ESA.ESASMC01.JOB09059.D0000111.?	Ddname:SYSPRINT	Records:80										
09:35:37:00	PLILINK.LKED	JES	ESA.ESASMC01.JOB09059.D0000112.?	Ddname:SYSPRINT	Records:406										
09:35:39:00	PLIRUN.	JES	ESA.ESASMC01.JOB09059.D0000113.?	Ddname:SYSPRINT	Records:3										
09:35:42:00	CGICOB.COBOL	JES	ESA.ESASMC01.JOB09059.D0000116.?	Ddname:SYSPRINT	Records:80										
09:35:43:00	CGICOB.LKED	JES	ESA.ESASMC01.JOB09059.D0000117.?	Ddname:SYSPRINT	Records:231										
09:35:44:00	CGICOB.GO	JES	ESA.ESASMC01.JOB09059.D0000118.?	Ddname:SYSPRINT	Records:5										
09:35:02:07	COMPILE .	Input	ESA.IBMH610.CPP(SCGICPP)	Ddname:SYSIN	Excp:2	DASD									
09:35:02:28	COMPILE .	Output	SYS13037.T093502.RA000.ESASMC01.R0187931	Ddname:OPTFILE	Excp:1	Temporary									
09:35:02:29	COMPILE .	Input	SYS13037.T093502.RA000.ESASMC01.R0187931	Ddname:OPTFILE	Excp:3	Temporary									
09:35:02:84	COMPILE .	Input	SYS13037.T093502.RA000.ESASMC01.R0187931	Ddname:OPTFILE	Excp:5	Temporary									
09:35:09:86	COMPILE .	Output	ESA.IBMH610.COMPLIST(SCGICPP)	Ddname:SYSCPRT	Excp:8	DASD									
09:35:09:95	COMPILE .	Input	SYS13037.T093502.RA000.ESASMC01.R0187931	Ddname:OPTFILE	Excp:7	Temporary									
09:35:10:26	COMPILE .	Input	ESA.IBMH610.CPP	Ddname:SYS00002	Excp:2	DASD									
09:35:10:48	COMPILE .	Input	CEE.SCEEH.H	Ddname:SYS00003	Excp:14	DASD									

### Sysout action

The “Sysout action” site gives the possibility to call some commands for a group of outputs:

- Email to send an Email via the SMTP server
- Print in a JES printer
- Copy into a dataset
- Open as PDF
- Open as AFP

### Dataset details

In the dataset list, if you select the link in the dataset name column, you get a pop-up menu with the different actions for the dataset. This action list depends on the dataset location (i.e. online or migrated) and on the datasets format. If the datasets is a PO the link under the ATTRIBUTES column give you to the member list. If you select a member in the member list you will get a pop-up menu with the different actions for a member:

**DSNAME:ZOSDEMO.SAMPLE.JOBS**

**9 members**

DSNAME	DSORG	CREATED	EXPIRE	REF	VOLSER	TRK	ATTRIBUTES
ZOSDEMO.SAMPLE.JOBS	PO	2010/03/25	1900/01/00	2013/02/06	USERL1	190	PO FB 80 32720

MEMBER	USER	CREATED	MODIFIED	SIZE
ABEND	ESA	11/04/27	11/04/27	39
COMPLIE				
IEFBR14	ESA	12/03/26	12/03/26	3
LONGRUN	ESA			
SAMPDSN	ESA			
SAMPLIST	ESA			
SAMPSUBM	ESA			
SAMP700	ESA			
SMSDAILY	ESA			

**ZOSDEMO.SAMPLE.JOBS(LONG(dataset,FB7))**

1. [Browse](#)
2. [Notepad](#)
3. [Edit](#)
4. [Submit](#)
5. [Dump Dataset](#)
6. [EBCDIC/ASCII Dump](#)
7. [Details](#)

### Active job details

If a job is active, you can get a pop-up menu to select different actions with this active job:

## Job status display from spool

Selection through PREFIX and OWNER

**SYSTEM Information**

- Configuration
- Diagnostic
- Performance
- Search and Find
- SDSF Collection
- Look the SYSLOG
- Check your jobs
- Check your datasets
- SQL SPUFI
- Operator
- Unix USS HFS etc
- MQ Series
- MVSEXP settings
- Bookshelves on IBM L

PREFIX:

OWNER:

JOBID	JOBNAME	OWNER	RETCODE	START	END	PCT	C
TSU09045	ESA	ESA	active	20			
JOB09059	ESASMCGL	ESA	CC 0004	20			
JOB09041	ESACSM1D	IBMUSER	CC 0000	20			
JOB09025	ESACT1D	IBMUSER	CC 0004	20			
JOB09008	ESAT	ESA	CC 0000	20			
JOB09005	ESAT	ESA	CC 0000	20			
JOB09003	ESAT	ESA	CC 0000	20			
JOB09002	ESAT	ESA	CC 0000	20			
JOB08999	ESAT	ESA	CC 0000	20			
JOB08998	ESAT	ESA	CC 0000	20			
JOB08997	ESAT	ESA	CC 0000	20			
JOB08965	ESAT	ESA	CC 0000	2013.2.5 12:51:42.46	2013.2.5 12:51:43.21	0.06	A

User:

JOBs:

OWNER:

Datasets:

LOG minutes:

Home:

Style:

Mozilla Firefox

10.1.8.1:8888/cgi-bin/run/jeslist.sh

**(TSU09045,ESA)(JES,active9)**

- [1. SMF activity report](#)
- [2. MXI active job report](#)
- [3. start APA request for a 10 seconds](#)
- [4. dump this address space](#)
- [5. dump and cancel](#)
- [6. cancel the job](#)
- [7. display active \(DA\)](#)
- [8. display progress in 10 seconds \(DA\)](#)
- [9. RMF III delay report](#)

# Miscellaneous

## Performance

For a better performance, consider the following options:

- Avoid the STEPLIB, if you can, in the “httpd.envvars”.
- Use the REXX compiler to compile the REXX scripts.
- Put the heavily used SDSF REXX interface routines into the LPA.

You can see the actual performance statistics under Configuration-HTTP Server-Server Activity.

## Diagnosis

To diagnose the problems with the REXX scripts, we put time to time the “TRACE I” REXX command into the scripts, in some place, after the call of the “commonhead” routine. Via the browsers “View Frame Source” option, this trace can be helpful. As the REXX scripts are getting everything from the environment variables, it is also possible to set/export these variables and debug the script from the USS shell:

```
export QUERY_STRING="JOBID=JOB8877"
MVIb:/u/mvsexp/server_root/cgi-bin/run>A188u:export QUERY_STRING="JOBID=JOB8877"
MVIb:/u/mvsexp/server_root/cgi-bin/run>A189u:export REFERER_URL="any"
MVIb:/u/mvsexp/server_root/cgi-bin/run>A190u:qedit jesjob.sh
MVIb:/u/mvsexp/server_root/cgi-bin/run>A191u:jesjob.sh
10 *-* parse upper var querystring "JOBID=" jobid ",DSNAME=" dsname ",FORM=" form ",PROGRAM=" program
>>> "JOB8877"
>>> ""
>>> ""
>>> ""
IRX01001 +++ Interactive trace. TRACE OFF to end debug, ENTER to continue. +++
==> _
```

## Used components

---

### Mainframe software

SHOWWLM from Yves Colliard

MXI from Rob Scott (CBT Tape 408)

VTOCLIST from John Kalinich and Peter Havercan (CBT Tape 343)

IODF to HTML Marc Naughton (CBT Tape 736)

Web Sphere REXX support (MA 95)

DB2WWW DB2 REXX support