

**Xymon Client for z/OS  
Installation and Configuration Guide**

**Version 1.0  
February, 2009**

---

## Table of Contents

Copyright.....	3
Changes.....	3
Introduction.....	4
Design Criteria.....	4
The Client.....	4
Installation of the Client.....	6
Configuring the Client.....	7
Operating the Client.....	9
Acknowledgments.....	10

---

## Copyright

Xymon Client for z/OS  
Copyright (C) 2008-2009 Richard Smrcina

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>

---

## Changes

2009/02/11 – Initial release  
2009/02/11 – Xymon name change

---

## Introduction

Xymon is an open source network services monitor. It runs on Linux (including Linux for System z). The Xymon Client for z/OS will perform tests on a z/OS system and report the status of those tests back to the Xymon server.

The client is considered an open source project. Anyone that is interested can contribute to it's development. I simply ask that any contributions be sent to me for consideration for distribution in future releases of the client.

Xymon is written and Copyright © 2007-2009 Henrik Storner.

---

## Design Criteria

The Xymon Client for z/OS is not intended as a performance monitor. The fact that it gathers some performance metric data is rooted in the nature of Xymon's role as a network host threshold monitor. It collects information every five minutes and reports it back to the Xymon server; this is much too sparse for true performance monitoring. It should not take on the task of a performance monitor as that is not it's purpose.

---

## The Client

The client is written primarily in REXX, with some assembler functions, and a COBOL program. It uses REXX sockets to send the status message(s) to the Xymon server.

The tests that are performed are as follows:

CPU, PAGING – Uses SDSF to determine current CPU Utilization and Page Rate

CICS – The supplied COBOL program gathers CICS Transaction Rate and DSA Utilization

DISK – Collects spool utilization from JES2 (using the SDSF MAS command), disk volume utilization, paging space utilization and z/OS Unix file system utilization

MAXUSER – Tests MAXUSERS, RSVTSTRT and RSVNONR from the ASVT

MEMORY – Tests CSA, ECSA, SQA and ESQA memory areas

PROCS – Uses SDSF DA output to get a list of the running jobs on the system

PORTS – Uses the output from NETSTAT

Additionally, it will graph the traffic for any interface that is reported by the NETSTAT DEV command.

Although the client was written on a system with JES2 and SDSF and is targeted for

those environments, it does not mean that it is limited to just those systems. The routines or functions that are specific to JES2 or SDSF can be replaced for sites that do not use those subsystems.

Figure 1 provides a list of the members provided in the distributed client and the function of each member.

<i>Filename</i>	<i>Usage</i>
DEFHOBIT	Jobstream to define a VSAM file used by HOBCICOL
HOBCICOL	COBOL program to collect task and DSA data and write it to the HOBBIT VSAM file
HOBDSCFG	Member used to indicate on which volumes to perform the disk utilization test
HOBLVTOC	Jobstream used to execute the LISTVTOC utility
HOBTEST	Member that contains the list of other execs to run, it is read by the driver exec
HOBVARS	Variables used during the execution of the client. Currently this includes the IP Address and port number of the Xymon server.
HOBZOS	The jobstream used to run the client
HOBZOSCD	Collects the data stored in the members in hlq.HOBBIT.DATA and sends it to the Xymon server
HOBZOSCI	Reads the VSAM file written by the CICS portion of the client and reports the data to the Xymon server.
HOBZOSCP	Performs the CPU Utilization, Paging, Procs and Maxusers test
HOBZOSDR	The driver program, controls the execution of all the other programs
HOBZOSDS	Performs the Disk Utilization test
HOBZOSIF	Performs interface traffic graphing by examining the output of NETSTAT DEV
HOBZOSME	Performs the Memory Utilization test
HOBZOSPO	Performs the ports test
LISTVTOC	Utility to list the VTOCs of selected volumes and report on their space utilization
RXVSAM	Assembler program that implements REXX function to access VSAM files
WAITREXX WAITTCB	Assembler programs to provide the sleep function for REXX and allow it to intercept the MVS P command for clean termination

*Figure 1 - List of files distributed with the package*

---

## Installation of the Client

The installation of the client is quite simple. The client code is distributed as a zip file. The zip file expands into this PDF file and an XMIT file.

The XMIT file should be transferred into a z/OS dataset. When using FTP to transfer the XMIT file, use the following SITE commands:

```
SITE RECFM=F  
SITE LRECL=80
```

Use the TSO RECEIVE command to expand the XMIT file into a PDSE that contains the distributed Xymon Client for z/OS. The default name of the PDSE is hlq.HOBBIT.REXX. One cylinder of disk space is allocated and it does not require any expansion. The record length is 80 bytes and the block size 8000.

An additional PDSE should be created called hlq.HOBBIT.DATA. This PDSE is used to store the data collected by the client code. One cylinder of disk space is sufficient for this dataset. The record format should be fixed blocked, record length 250 bytes, block size 8000.

The client runs as a started task called XYMONZOS. The provided jobstream in the PDSE is called XYMONZOS should be tailored to fit site standards, namely the SYSEXEC DD referring to the High Level Qualifier. It should then be copied to a system PROCLIB.

The jobstream has a JOBLIB specifically for the load modules that the Xymon client for z/OS uses. If site standards dictate a different place for those load modules, change the DD statement or remove it in favor of a system wide load library.

The assembler programs RXVSAM, WAITREXX and WAITTCB must be assembled and added to a load library for use by the client code. They provide the REXX VSAM access function and the wait function for REXX to allow the started task to be terminated cleanly with the MVS P command. After the three programs are assembled and put into a load library, the source can be moved out of hlq.HOBBIT.REXX, if desired, since they don't have any direct relation to the normal operation of the client.

If you wish to use the CICS portion of the client, compile HOBCICOL as a COBOL CICS program. Define it to each CICS that will need to report data to Xymon. Run the job stream to define the Xymon VSAM file, changing it based on site standards. Define the file to each CICS region as HOBBIT. The file will need all operations activated; READ, ADD, BROWSE, DELETE and UPDATE. Add the program HOBCICOL to the startup PLT.

SDSF security needs to be addressed through RACF and/or the ISFPRMxx member in your parmlib. This change is so that the Xymon client can issue the SDSF DA and MAS commands. One suggested change would be to add the following group after ISFSPROG as shown in Figure 2.

```

/*****/
/* GROUP ISFXYM - Xymon */
/*****/
GROUP NAME (ISFXYM), /* Group name */
IUID (XYMON), /* User must have JCL and OPER */
ACTION (ALL), /* All route codes displayed */
ACTIONBAR (YES), /* Display action bar on panels */
APPC (ON), /* Include APPC sysout */
AUPDT (2), /* Minimum auto update interval */
AUTH (DA, MAS), /* Authorized functions */
CMDAUTH (ALL), /* Commands allowed for all jobs */
CMDLEV (7), /* Authorized command level */
CONFIRM (ON), /* Enable cancel confirmation */
CURSOR (ON), /* Leave cursor on last row processed */
DADFLT (IN, OUT, TRANS, STC, TSU, JOB), /* Default rows shown on DA */
DATE (MMDDYYYY), /* Default date format */
DATESEP ('/'), /* Default datesep format */
DISPLAY (OFF), /* Do not display current values */
DSPAUTH (USERID, NOTIFY, AMSG), /* Browse authority */
GPLEN (2), /* Group prefix length */
ILOGCOL (1), /* Initial display column in log */
ISYS (LOCAL), /* Initial system default for DA */
LANG (ENGLISH), /* Default language */
LOGOPT (OPERACT), /* Default log option */
OWNER (NONE), /* Default owner */
UPCTAB (TRTAB2), /* Upper case translate table name */
VALTAB (TRTAB), /* Valid character translate table */
VIO (SYSALLDA) /* Unit name for page mode output */

```

Figure 2 – Example of SDSF security settings for Xymon

Then add the following NTBL list later in the member:

```

NTBL NAME (XYMON)
NTBLENT STRING (START2), OFFSET (1)

```

Refresh the SDSF configuration with F SDSF, REFRESH.

---

## Configuring the Client

There are a couple of configuration members in the PDS that will need some attention.

HOBVARS contains the IP address and port number that the Xymon server listens on. Figure 3 shows the member.

```

/* Xymon z/OS client options member

   Address of Xymon server          */
bbdisplay='192.168.128.10'
/* Xymon port number              */
bbport=   '1984'

```

Figure 3 – HOBVARS Configuration Member

Edit the member, change the value of *bbdisplay* and *bbport* (if necessary) and save the member.

Multiple *bbdisplays* are supported to send client data to more than one Xymon server. Separate the IP addresses of the *bbdisplays* with one or more spaces. The port number can also be specified for each *bbdisplay* by adding a colon and the port number to the IP Address, eg: 192.168.1.72:1984. The *bbport* specification specifies the port number of the *bbdisplay* and must be specified, even if the port number is coded on the *bbdisplay*.

HOBDSCFG contains a list of disk volumes that are examined for the disk utilization test. Figure 4 shows an example of the member.

```

VOL=OS39M1
VOL=OSWRK1
VOL=OS39PG

```

Figure 4 - Sample list of volumes

Volumes can be added to this member for the disk utilization test. Add volumes to be tested to this member in the format

*VOL=valid*

Where *valid* is a volume identifier. Specify one volume per line.

The remainder of the configuration of the Xymon Client for z/OS is performed on the Xymon server. This involves setting threshold values that may be different than the built in defaults. The configuration file *\$HOME/server/etc/hobbit-clients.cfg* is used. Specify the hostname of the z/OS system and the keywords necessary to set the thresholds. There are standard Xymon client configuration keywords and specific keywords used for System z. The meaning of the client configuration keywords for z/OS are as follows:

- LOAD: CPU Utilization as measured by SDSF
- PAGING: Page rate as measured by SDSF
- MEMCSA: CSA Utilization rate
- MEMECSA: ECSA Utilization rate
- MEMSQA: SQA Utilization rate
- MEMESQA: ESQA Utilization rate

CICS: DSA and EDSA Utilization rate by application ID

MAXUSER: Utilization of Maxuser configured value

PORT: Same usage as Linux client

DISK: Same usage as Linux client

(Volumes are represented as /VOL/*valid*, JES spool space is represented as /JES/SPOOL,

z/OS paging spaces are represented as /PAGE/PLPA/*valid*, /PAGE/COMMON/*valid*

or /PAGE/LOCAL/*valid*; z/OS Unix file systems are represented by their mount points)

PROC: Used to track running jobs, started tasks or TSO users

FILE, DIR and MSGS: Not available (yet) for z/OS

Figure 5 shows how the configuration file would look for a z/OS system.

```
HOST=zoshost
LOAD      85.0 95.0
PAGING    5 10
CICS      CICS1 DSA 85 90 EDSA 95 97
MEMCSA    90 95
MEMECSA   100 100
MEMSQA    90 95
MEMESQA   90 95
MAXUSER   80 85
PORT      "LOCAL=%[:]21$" STATE=LISTEN min=1 TEXT=FTP
PORT      "LOCAL=%[:]23$" STATE=LISTEN min=1 TEXT=Telnet
PORT      "LOCAL=%[:]80$" STATE=LISTEN min=1 "TEXT=Web Server"
PORT      "LOCAL=%[:]23$" STATE=ESTABLISHED min=0 TRACK=telnet "TEXT=Telnet Connections"
PROC      %.* 1 70 yellow TRACK=AllJobs "TEXT=All Jobs"
PROC      NFSS 1 1 yellow "TEXT=NFS Server"
```

Figure 5 – Client configuration for z/OS

In the example, if the z/OS system *zoshost* goes above 85% CPU Utilization, it's CPU column will turn yellow, if it goes above 95% it will turn red. The paging and memory tests work similarly. The PORT test works exactly the way the Linux PORT test works. The first PROC test will track (graph) how many jobs are running on the system, the second one will actually check for a specific job and turn the test yellow if it isn't running. The CICS test allows the DSA and EDSA utilization to be tested separately for each CICS region running on the system (as long as the CICS client program is executing in that region).

The job HOBLVTOC is used to regularly gather utilization information for selected disk volumes. It reads the HOBDSCFG member for volume identifiers and writes capacity information out to a member for the disk utilization test to read. With a job scheduler or the JES2 \$TA command, set up HOBLVTOC to run at regular intervals, like every hour, or a comfortable interval to gather volume utilization data. The disk test still runs at 5 minute intervals, since it still gathers spool and paging utilization data. It will only see changes to disk volumes at the interval that you select for the HOBLVTOC job.

The REXX programs that are executed in both jobstreams, XYMONZOS and HOBLVTOC require a single parameter. That parameter is the high level qualifier to be used for the execution of the job. It is used in the TSO ALLOCATE statements when accessing members in the hlq.HOBBIT.DATA PDSE dataset.

The representation of the data from the disk test is based on how the data is retrieved from its source. There is no attempt made to convert any of the measurements to a common format. Spool utilization comes from the SDSF MAS command and does not provide any other information than the percent utilized. The volume test is based on the number of tracks as produced by the IEHLIST utility. The page device utilization is based on the number of slots maintained in the MVS control blocks.

---

## Operating the Client

The Xymon Client for z/OS will run as a started task. As such it can be initiated with the MVS command:

```
S XYMONZOS
```

When you want to terminate the client, use the command:

```
P XYMONZOS
```

The Xymon server should be configured to perform at least a basic connection test on the z/OS system where the client is installed. If this is not done, the default behavior of Xymon will be to ignore the status messages received from the client.

---

## Acknowledgments

Code in HOBZOSDR, HOBZOSDS and HOBZOSME borrowed from Mark Zelden's IPLINFO utility, with permission. IPLINFO is available at <http://home.flash.net/~mzelden/mvsutil.html>.

LISTVTOC was written by Joerg Berning and is from CBT tape 344. It is used here with permission.

RXVSAM was written by Mark Wings, redistributed and used here with permission.

Finally, thanks to Henrik Storer for writing the package and for assistance with the C code for the backend to support the clients.