



An apple a day helps keep the outages away: IBM Health Checker for z/OS® - Setup

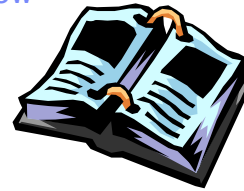


IBM Health Checker for z/OS is the very popular potential problem catcher . Originally delivered as a Web prototype, the IBM Health Checker for z/OS has matured into a z/OS R7 base function that is also available for z/OS R4, R5, and R6. It provides a very robust framework for automating the identification of potential problems.

This session tells users how to set up and start IBM Health Checker for z/OS. Also included is a suggested implementation for reviewing check output, resolving exceptions and providing your own overrides. In addition, the new SDSF CK pane, which simplifies check management is also included in the screen shots.

Recommendation: Review the IBM Education Assistant Module - IBM Health Checker for z/OS - Overview

- Why do you need a health checker?
- History of IBM Health Checker for z/OS
 - Prototype
 - IBM Health Checker for z/OS integration into z/OS
 - IBM Health Checker for z/OS check concepts
- Making your own checks for IBM Health Checker for z/OS



Please review the IBM Education Module – zOS Health Checker Overview. This module will cover:

What analysis of outages showed, what factors may exacerbate the outage situation, and how the health checker can help to avoid outages.

Details of the prototype developed, access to prototype and install instructions, prototype user's requests

Two parts of the Health Checker

Framework

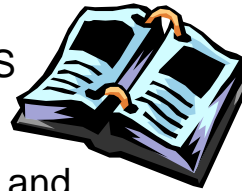
The individual checks that look for component, element, or product specific z/OS settings and definitions, checking for potential problems. The specific component or element owns, delivers, and supports the checks.

A check is actually a program or routine that identifies potential problems that may impact your availability or, in worst cases, cause outages. A check is owned, delivered, and supported by the component, element, or product that writes it.

How to create an IBM Health Checker for zOS check for your component or product.

Agenda

- Setting up IBM Health Checker for z/OS
- Sample check outputs
- Check exceptions – what they look like and what to do about them
- Screen shots – setting up and using the IBM Health Checker for z/OS on a z/OS R7 system



We've identified 9 steps needed to set up and start IBM Health Checker for z/OS.

Provided a sample check output for:

- When a check is executed without finding an exception.
- If a check is not appropriate for the current environment.
- Sample check outputs when exceptions are found.

What to do with exceptions identified - Users should spend time to review the exceptions identified and evaluate your system.

The checks reflect generally accepted recommendations, but you will need to evaluate whether each suggestion is appropriate for your system. Exceptions are a means for you to evaluate potential availability impacts and take action.

Also included are screen shots on the setup process.

IBM Health Checker for z/OS – Setup

Use the following steps to set up and start IBM Health Checker for z/OS:

1. Satisfy software requirements for IBM Health Checker for z/OS
2. Allocate the HZSPDATA data set to save check data between restarts
3. Set up the HZSPRINT utility
4. Define log streams to keep a record of the check output, as needed
5. Create security definitions
6. Set up customization and security for SDSF support for IBM Health Checker for z/OS in IBM Health Checker for z/OS Small Programming Enhancement in *z/OS SDSF Operation and Customization*
7. Create HZSPRMxx from the HZSPRM00 parmlib member
8. Start IBM Health Checker for z/OS
9. Obtain checks for IBM Health Checker for z/OS



Here are the 9 steps that this module will discuss.

1. Satisfy software requirements



Software Requirements for IBM Health Checker for z/OS:

- **Framework:** z/OS R4, z/OS R5, z/OS R6, or z/OS R7
 - **Checks:** may be dependent upon the level of z/OS you have
 - Currently, checks are available for: Consoles, Contents Supervisor, RRS, XCF/XES, GRS, SDUMP, Security Server, RSM, VSM, and z/OS UNIX System Services.
 - We plan to add more check periodically via APARs and integrated into z/OS releases
 - For obtaining the most recent appropriate checks for your z/OS level, use the Enhanced PSP Tool at http://techsupport.services.ibm.com/390/psp_main.html
- You can identify available checks by retrieving the Extract File, and running the tool against it. Retrieve the Extract File by selecting a Type of *Function* and a Category of *Health Checker*.
- Note that the names of some checks have changed since the prototype!

5

© 2005 IBM Corporation

We plan to add more checks periodically via APARs and integrated into z/OS releases, both APARs and integrated into z/OS. To identify checks that have been provided in PTFs, use the Enhanced Preventive Service Planning Tool, available at the following Web site: http://techsupport.services.ibm.com/390/psp_main.html.

For check output messages, see the component message books or use message explanations directly from the **LookAt** Web site at <http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/>. Because checks along with their output messages might be added by PTF between releases of component message books, LookAt will contain the most up to date message information.

Currently, we have the following checks available for z/OS R4, z/OS R5, z/OS R6, and z/OS R7 components:

Component: Consoles, R4	APAR: OA09095,	Applicable to: z/OS R7, R6, R5, and R4
Component: CSV,	APAR: not yet available,	Applicable to: z/OS R7, R6, R6, and R4
Component: RRS,	APAR: not yet available,	Applicable to: z/OS R7, R6, R6, and R4
Component: XCF/XES, R4	APAR: OA07513,	Applicable to: z/OS R7, R6, R6, and R4
Component: GRS,	PTFs: UA19875 (R7), UA19874 (R6), UA19873 (R5), UA19872 (R4)	
Component: SDUMP, R4	APAR: OA09306,	Applicable to: z/OS R7, R6, R6, and R4
Component: Sec Srvr RACF,	APAR: not yet available,	Applicable to: z/OS R7, R6, R6, and R4
Component: RSM,	APAR: not yet available,	Applicable to: z/OS R7, R6, R6, and R4
Component: VSM	APAR: OA09367,	Applicable to: z/OS R7, R6, R6, and R4
Component: z/OS USS,	APAR: OA09276,	Applicable to: z/OS R7, R6, R6, and R4

Several check names have changed since they were released in the prototype. There is a table in the *IBM Health Checker for z/OS User's Guide* that lists the old and new check names.

2. Allocate HZSPDATA data set

Allocate the HZSPDATA data set to save check data between restarts

- Some checks use the HZSPDATA data set to save data required as part of their processing between restarts of the system or IBM Health Checker for z/OS.
- Use the HZSALLCP sample JCL from SYS1.SAMPLIB.
- Must have one HZSPDATA data set per system image! Recommended to use system name in the HZSPDATA name, therefore.

```
//HZSALLCP EXEC PGM=HZSAIEOF,REGION=4096K,TIME=1440
//HZSPDATA DD DSN=SYS1.system_name.HZSPDATA,DISP=(NEW,CATLG),
//          SPACE=(4096,(100,400)),UNIT=SYSDA,
//          DCB=(DSORG=PS,RECFM=FB,LRECL=4096)
//SYSPRINT DD DUMMY
```

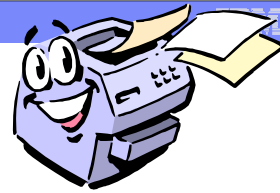
Will be in HZSPROC

Don't change these attributes

Each system that starts IBM Health Checker for z/OS must have a unique HZSPDATA dataset. IBM recommends that the system name be included in name of the HZSPDATA dataset.

HZSPDATA DD name must reflect the name of the dataset to be allocated. The data set must be FB with an LRECL of 4096.

Retain the name of the HZSPDATA data set so you can specify it in the IBM Health Checker for z/OS start up procedure, HZSPROC.



3. Set up HZSPRINT utility

Set up the HZSPRINT utility to view print output

- HZSPRINT utility allows you to see check output in the message buffer.
 - May also or instead use SDSF or log streams
- HZSPRINT writes the current message buffer for the target checks to SYSOUT.
- Use the HZSPRINT sample JCL from SYS1.SAMPLIB
 - If LOGSTREAM is specified, then RACF auth to the requested logstream is required.
 - If LOGSTREAM is not specified, then RACF auth to the requested check(s) is req.

```
//HZSPRINT EXEC PGM=HZSPRINT,TIME=1440,REGION=0M,
//  PARM=( 'CHECK( check_owner , check_name )' ) Defaults to *,*
// *  PARM=( 'LOGSTREAM( logstreamname ) , 'EXCEPTIONS' ,
// *  'SYSNAME( sysname )' ) Only show exceptions
// *  'CHECK( owner , name )' Defaults to *
//SYSOUT DD SYSOUT=A,DCB=( LRECL=256 )
When using a data set, it
must be seq, FB, LRECL 256
```

7

© 2005 IBM Corporation

The HZSPRINT print utility is used to write the current message buffer for the target check(s) to the specified SYSOUT.

HZSPRINT Parameters (parameters must be separated from each other by a comma):

LOGSTREAM(log_stream_name) Where log_stream_name is the name of a log stream that is (or was) in use by IBM Health Checker for z/OS. If LOGSTREAM is specified, the message buffers are obtained from the specified log_stream_name. If LOGSTREAM is not specified, the message buffers are obtained from the checks that are currently registered with the IBM Health Checker for z/OS.

SYSNAME(system_name) Where system_name is the name of a system where the checks were executed. Wildcard characters '*' and '?' may be used if the output of multiple systems is desired. The SYSNAME keyword is only valid when LOGSTREAM is also specified. If the SYSNAME keyword is not specified with LOGSTREAM, processing is equivalent to a specification of SYSNAME(*).

CHECK(check_owner,check_name) Where check_owner and check_name identify the desired check(s) to print. Wildcard characters '*' and '?' may be used if the output of multiple checks is desired. Note that it might not be necessary to surround the CHECK parameter specification with quotes, but it often is and is always safe to do so. If the CHECK keyword is omitted, processing is equivalent to a specification of CHECK(*,*).

EXCEPTIONS If the EXCEPTIONS keyword is specified, the message buffer output is included only if the check resulted in exception(s).

SYSOUT DD is the output for the print command. If you want to allocate a data set for HZSPRINT output, the data set must be fixed length, blocked records (for example, RECFM=FBA or RECFM=FBM) and logical record length of 256. Add the name of the output data set allocated above to the HZSPRINT JCL. For example:

```
//SYSOUT DD DISP=SHR,DSNAME=D10.HCHECKER.REPORT.FEB2505,DCB=(LRECL=256)
```

Note that the first character of each line of HZSPRINT output is a carriage control character.

If LOGSTREAM is specified then RACF authorization to the requested logstream is required.

If LOGSTREAM is not specified then RACF authorization to the requested check(s) is required.

Refer to the *IBM Health Checker for z/OS User's Guide* for information on return codes for the HZSPRINT utility.

4. Define log streams

Define log streams to keep a record of the check output, as needed

- Only the last iteration of a check is retained in the message bugger.
 - It's a good idea to retain historical data about your check results. To do that, you must define and connect to a log stream.
1. Plan for and set up the log streams. You may have one log stream per system or one log stream for multiple systems. Must begin with *HZS*.
 2. Enable your log streams via MODIFY command or in HZSPRMxx parmlib member.
 3. You can disable the log stream with a MODIFY command.



8

IBM Health Checker for z/OS retains only the check results from the last iteration of a check in the message buffer. If you want to retain a historical record of check results, which is a good idea, you must define and connect to a log stream. When you have a log stream connected, the system writes check results to the log stream every time a check completes.

1. Plan for and set up the log streams, including allocation of coupling facility and DASD space. Careful planning of DASD and coupling facility space is important because if the log stream fills up, no additional data will be written to it and data will be lost. Define either one log stream for each system or one log stream for multiple systems to use. *HZS* must be the first letters of log stream names you define. For example, you might define a log stream name of *HZS.HEALTH.CHECKER.HISTORY*.

Security for the log stream is accomplished for the most part when you set up security for the IBM Health Checker for z/OS super User ID above. However, The user who will be setting up log stream and structure definitions for the IBM Health Checker for z/OS log stream using the IXCMIAPU administrative data utility program must have authorization to a number of resources. See the "Define Authorization for Setting Up Policies" section of *z/OS MVS Setting Up a Sysplex*.

2. Enable log streams in one of the following ways: use the MODIFY command (`f hzsproc,logger=on,logstreamname=logstreamname`) -or- use the LOGGER parameter in the HZSPRMxx parmlib member (`LOGGER(ON) LOGSTREAMNAME(logstreamname)`).
3. To disable a log stream, issue the following MODIFY command: `f hzsproc,logger=off`

The following shows a log stream definition example using the coupling facility:

Coupling Facility Structure definition: The following example shows a log stream structure definition defined in the CFRM policy using the administrative data utility, IXCMIAPU:

```
STRUCTURE NAME(HZS_HEALTHCHKLOG) SIZE(8000)
PREFLIST(CF25, CF01C, CF1)
MAXBUFSIZE(65532)
```

The value defined for SIZE should be no less than 8000 to ensure adequate space for check data.

Log stream definition: The following example shows a coupling facility log stream definition defined in the LOGR policy using the administrative data utility, IXCMIAPU:

```
DEFINE LOGSTREAM NAME(HZS.HEALTH.CHECKER.HISTORY)
DESCRIPTION(HEALTH_CHECK_RPT)
STRUCTNAME(HZS_HEALTHCHKLOG)
STG_DUPLEX(NO)
LS_DATACLAS(NO_LS_DATACLAS) LS_MGMTCLAS(NO_LS_MGMTCLAS) LS_STORCLAS(NO_LS_STORCLAS)
LS_SIZE(4096) AUTODELETE(YES) RETPD(14) HIGHOFFLOAD(80) LOWOFFLOAD(0) DIAG(YES)
```


5. Create security definitions



Create security definitions

1. Set up security for the IBM Health Checker for z/OS started task, similar to other started tasks:
 - a) Create user ID for IBM Health Checker for z/OS that has superuser authority. Connect that superuser ID to a group.
 - b) Associate that superuser ID to the started task, HZSPROC.
 - c) Give that superuser ID access to HZSPDATA data set you allocated on each system.
 - d) Give that superuser ID read access to HZSPRMxx parmlib members.
 - e) If using log streams, give read access for that superuser ID to each log stream(s).
2. If using HZSPRINT utility, follow instructions in the *User's Guide*.
3. If using SDSF CK panel, set up security for SDSF.

NOTE: RACF supplies new classes - XFACILIT and GXFACILI - which are used. Support for these new classes is integrated in z/OS R7, and supplied back to z/OS R4 via APAR OA10774.

9

© 2005 IBM Corporation

Both IBM Health Checker for z/OS and users looking at check output need access to resources. You must create security definitions to control access and maintain security for these resources. Do the following to create security definitions for IBM Health Checker for z/OS:

1. You must set up security for IBM Health Checker for z/OS the same way you would for any other started task. To do this task with RACF, do the following steps:
 - Create a user ID for IBM Health Checker for z/OS with superuser authority (UID(0)) and connect this superuser user ID to a group. For example: `ADDUSER hcuserid OMVS(UID(0) HOME('/') PROGRAM('/bin/sh')) NOPASSWORD`, then `ADDGROUP OMVSGRP OMVS(GID(46))`, then `CONNECT hcuserid GROUP(OMVSGRP)`.
 - Associate the superuser User ID, *hcuserid*, with the IBM Health Checker for z/OS started task, HZSPROC. For example: `SETROPTS GENERIC(STARTED)`, then `RDEFINE STARTED HZSPROC.* STDATA(USER(hcuserid) GROUP(OMVSGRP))`, then `SETROPTS CLASSACT(STARTED) RACLIST(STARTED)`, then `SETROPTS RACLIST(STARTED)`. If you had already RACLISTed the STARTED class, the last statement would have to be `SETROPTS RACLIST(STARTED) REFRESH`.
 - Give the IBM Health Checker for z/OS started task super User ID access to the HZSPDATA data set on each system where you'll run IBM Health Checker for z/OS. For example, you might specify the following: `ADDSD 'SYS1.PRODSYS.HZSPDATA' UACC(NONE) PERMIT SYS1.PRODSYS.HZSPDATA CLASS(DATASET) ID(hcuserid) ACCESS(UPDATE)`
 - Give IBM Health Checker for z/OS started task super User ID READ access to the HZSPRMxx parmlib member(s). For example, you might specify the following: `ADDSD 'SYS1.PARMLIB' UACC(NONE) PERMIT 'SYS1.PARMLIB' CLASS(DATASET) ID(hcuserid) ACCESS(READ)`
 - If you will be using a log stream, you must define READ access for the IBM Health Checker for z/OS started task super User ID to each RESOURCE(*logstreamname*) CLASS(LOGSTRM). IBM Health Checker for z/OS connects directly to the defined log stream or streams. For example, you might specify the following: `RDEFINE LOGSTRM logstreamname UACC(NONE)`, then `PERMIT logstreamname CLASS(LOGSTRM) ID(hcuserid) ACCESS(UPDATE)`, then `SETROPTS CLASSACT(LOGSTRM) RACLIST(LOGSTRM)`, then `SETROPTS RACLIST(LOGSTRM)`. If you had already RACLISTed the LOGSTRM class, the last statement would have to be `SETROPTS RACLIST(LOGSTRM) REFRESH`.
2. If you will be using HZSPRINT to view check output, refer to the *IBM Health Checker for z/OS User's Guide* for detailed instruction on creating the necessary security definitions, based on how you will be using HZSPRINT.
3. Set up customization and security for SDSF support for IBM Health Checker for z/OS in IBM Health Checker for z/OS Small Programming Enhancement in *z/OS SDSF Operation and Customization*.

Security Server provides two new RACF classes, which are used by the IBM Health Checker for z/OS. These new RACF classes are: XFACILIT and GXFACILI. Support is integrated in z/OS V1.7 and provided in the following PTFs for earlier z/OS releases: z/OS V1.6 (HRF7709) — UA18194, z/OS V1.5 (HRF7708) — UA18193, z/OS V1.4 (HRF7707) — UA18192.



6. Set up SDSF support

Set up customization and security for SDSF support, if desired

- SDSF provides support to make management of your checks easier with the new CK panel for the IBM Health Checker for z/OS.
- You can use the CK panel to display checks, attributes, and status, taking advantage of standard SDSF sort, filter, and arrange support.
- The SDSF support will also be made available in APAR PK00561 for z/OS R4, R5, and R6. It is integrated into z/OS R7.
- Use “IBM Health Checker for z/OS Small Programming Enhancement” instructions in *z/OS SDSF Operation and Customization*:
 - Authorize users to the CK command using either the AUTH parameter of ISFPARMS or SAF.
 - Authorize users to action characters or overtypable fields with ISFPARMS or SAF
 - Customize columns on the panel for groups of users with ISFPARMS.
 - Customize data displayed or user authority with user exits.

z/OS R7 SDSF provides support to make management of your checks easier with the new CK panel for the IBM Health Checker for z/OS. You can use the CK panel to display checks, attributes, and status, taking advantage of standard SDSF sort, filter, and arrange support. The SDSF support will also be made available in APAR PK00561 for z/OS R4, R5, and R6.

You can find more information in the following manual: *z/OS V1R7.0 SDSF Operation and Customization (SA22-7670-08)* .

7. Create HZSPRMxx parmlib member



Create HZSPRMxx from the HZSPRM00 parmlib member

- At first, don't modify your HZSPRMxx parmlib member. Check to see what you get as output from IBM Health Checker for z/OS. Then, make permanent updates for your environment.
- HZSPRMxx should ONLY include:
 - Policy statements (for changes applied to added or refreshed checks)
 - LOGGER parameter (for indicating what log stream to use)
- Other non-policy statements are ineffective, because the HZSPRMxx specified in your HZSPROC is processed before any checks are begun.
- HZSPRM00 is supplied in your DDDEF's PARMLIB data set:

```

/* LOGGER=ON, LOGSTREAMNAME=HZSLogstream          Remember, must start with HZS */
/*                                                                                   */
/* {ADD | ADDREPLACE}, POLICY, STATEMENT=statementname, UPDATE, filters,           */
/*      update_options, REASON=(reason text), DATE=yyyymmdd                       */

```

At first, you'll want to run IBM Health Checker for z/OS without modifying the HZSPRMxx member to see what check output you get on your installation. Later, as you evaluate your check output, you should use an HZSPRMxx parmlib member to make permanent updates to check values and parameters or to keep a check from running (deactivating the check). Your HZSPRMxx parmlib member should include **only**:

- Policy statements, to make changes that are applied to checks that are added or refreshed.
- The LOGGER parameter, if you want to use a log stream: LOGGER(ON) LOGSTREAMNAME(logstreamname)

Including other non-policy statements in your HZSPRMxx member will be ineffective, because the parmlib member specified in the *hzsproc* procedure is processed before any checks are added or begin running.

You can create the policy statements in your HZSPRMxx parmlib member using input from the sample syntax in the HZSPRM00 member in SYS1.PARMLIB.

The policy parameters are used to define policy statements that modify the behavior of specified checks:

```
{ADD | ADDREPLACE},
POLICY, STATEMENT=statementname, UPDATE, filters, update_options, REASON=(reason text), DATE=yyyymmdd
```

Where

ADD - This is a new policy statement that is not active. If the named policy statement is already active, the new policy statement is rejected.

ADDREPLACE - The specified policy statement may already be active. If the policy statement is already active, the existing policy statement is replaced.

statementname - 1-16 character policy statement name used to identify the policy statement.

UPDATE - Indicates the policy statement overrides check defaults

filters - Filters that indicate which check(s) are targeted by this policy statement:

CHECK(owner, name) - (Required.) The 1-6 character check owner, and 1-32 character check name. Wild card symbols '*' and '%' are permitted.

[CATEGORY]([filter_type],[category-1],[category-2...]) Allows additional filter capacity based on the current assigned categories

[EXITRTN=exitrtn] - The name of the HZSADDCHECK dynamic exit routine that was used to add the check.

Update_options-The options used to override the check defaults:

[ACTIVE|INACTIVE] Indicates the target check(s) are ACTIVE or INACTIVE.

[,ADD CAT=(cat1,...,cat16)] Add the target check(s) to the specified categories

[,DESC CODE=(desc code1,...,desc code#)] Additional descriptor code(s) which will be used when an exception message is written by the target check(s)

[, [INTERVAL=ONETIME|INTERVAL=h hh:mm]] The interval at which the target check(s) will be run.

[, [PARAM=parameter]] The check specific parameter that will be passed to the target check.

[, [ROUT CODE=(route code1, ..., route code#)] Additional route code(s) which will be used when an exception message is

8. Start IBM Health Checker for z/OS

Start IBM Health Checker for z/OS



1. Copy SAMPLIB's HZSPROC into your proclib data set.
2. Update to point to your HZSPDATA data set and HZSPRMxx parmlib member you previously set up.
3. Update your COMMNDxx parmlib member with `COM='START HZSPROC'`
4. Start the started task: `S HZSPROC,HZSPRM=00`

```
//HZSPROC JOB JESLOG=SUPPRESS
//HZSPROC PROC HZSPRM='00'
//HZSSTEP EXEC PGM=HZSINIT,REGION=0K,TIME=NOLIMIT,
//          PARM='SET PARMLIB=&HZSPRM'
//HZSPDATA DD DSN=SYS1.&SYSNAME..HZSPDATA,DISP=OLD
//          PEND
//          EXEC HZSPROC
```

Annotations in the original image:

- Red arrow pointing to `HZSPRM='00'`: Your parmlib member you set up
- Red arrow pointing to `DSN=SYS1.&SYSNAME..HZSPDATA`: Your saved check data set you set up

Note that the HZSPROC looks like a batch job, but it **is a started task**. IBM Health Checker for z/OS is set up this way in order to suppress messages to the JESLOG, which might otherwise overflow your JESLOG data set.

1. Copy the sample IBM Health Checker for z/OS procedure, HZSPROC, into a PROCLIB data set.
2. Update the procedure to make sure that the procedure includes the name of the HZSPDATA data set you defined previously, and make sure that the procedure includes the name of your HZSPRMxx member you previously set up.
3. Update the COMMNDxx parmlib member with the IBM Health Checker for z/OS procedure, as shown in the following example: `COM='START hzsproc'`
4. Start IBM Health Checker for z/OS:
 - Start IBM Health Checker for z/OS with one or more HZSPRMxx parmlib members using one of the following commands: `START hzsproc,HZSPRM=xx` -or- `START hzsproc,HZSPRM=(x1,...,xn)` where `xx` is the suffix of the HZSPRMxx member you want to use.
 - You can start IBM Health Checker for z/OS **without** specifying an HZSPRMxx parmlib member using one of the following commands: `START hzsproc`. The system uses default HZSPRMxx member HZSPRM00 if you do not specify a member on the HZSPRM= parameter.

In subsequent IPLs, the IBM Health Checker for z/OS procedure will start automatically, as prompted in the COMMMDxx parmlib member. If you start HZSPROC without specifying an HZSPRMxx member, the system uses HZSPRM=00 as the default.

The very first time you start IBM Health Checker for z/OS, you might see a message such as the following:

```
HZS00101 THE HZSPDATA DATA SET CONTAINS NO RECORDS
```

This output reflects the fact that the HZSPDATA data set as yet contains no data.

9. Obtain checks

Periodically, obtain checks for IBM Health Checker for z/OS

- We plan to add more checks, periodically look for new ones we may provide
- Reminder, for obtaining the most recent appropriate checks for your z/OS level, use the Enhanced PSP Tool at http://techsupport.services.ibm.com/390/psp_main.html
- The PTF letter for a check will describe how to add the new check.



Checks for the IBM Health Checker for z/OS are delivered both as an integrated part of a z/OS release or separately, as PTFs. New and updated checks are planned to be distributed as PTFs, so that they are not dependent on z/OS release boundaries and can be added at any time. To identify checks that have been provided in PTFs, use the Enhanced Preventive Service Planning Tool, available at: http://techsupport.services.ibm.com/390/psp_main.html .

You can identify checks by checks by selecting type **Function** and category **Health Checker**.

You can also find PTF numbers for checks in the PSP (preventive service planning) buckets. PSP buckets are identified by UPGRADEs, which specify product levels, and SUBSETs, which specify the FMIDs for a product level. For IBM Health Checker for z/OS, HCHECKER is the upgrade name, the subset for the framework is HZS7720, and the subset for the checks is **checks**.

Some sample check outputs - SUCCESS

```
CHECK( IBMRSM,RSM_MAXCADS )
START TIME: 06/07/2005 10:55:38.139127
CHECK DATE: 20041006 CHECK SEVERITY: MEDIUM
CHECK PARM: THRESHOLD(80%)
IARH108I The current number of in use CADS entries is
17, which represents 34% of the total allowed CADS
entries of 50. The highest usage of CADS entries
during this IPL is 34%, or 17 total entries. This is
below the current IBMRSM supplied threshold of 80%.
END TIME: 06/07/2005 10:55:38.139653 STATUS: SUCCESSFUL
```

If a check runs without finding an exception, it should issue an informational message with that information to the message buffer. This example shows a clean check run case, viewed in the message buffer. Note that the status of the check - **STATUS: SUCCESSFUL**.

Some sample check outputs – Not appropriate

```
CHECK( IBMCNZ ,CNZ_SYSCONS_MASTER )
START TIME: 07/05/2005 14:45:22.739250
CHECK DATE: 20040816 CHECK SEVERITY: HIGH
HZS1003E CHECK( IBMCNZ ,CNZ_SYSCONS_MASTER ) :
THE CHECK IS NOT APPLICABLE IN THE CURRENT SYSTEM
ENVIRONMENT.
CNZHF1004I The system console is not present. The check
is not applicable in this environment.
END TIME: 07/05/2005 14:45:22.740948 STATUS: ENV N/A
```

If a check is not appropriate for the current environment, it should issue an informational message with that information to the message buffer.

Some sample check outputs – EXCEPTION

```
CHECK(IBM CNZ,CNZ_CONSOLE_MSCOPE_AND_ROUTCODE)
START TIME: 06/08/2005 09:49:17.410704
CHECK DATE: 20040816 CHECK SEVERITY: LOW
* Low Severity Exception *
CNZHF0003I One or more consoles are configured with a combination of message
scope and routing code values that are not reasonable.
Explanation: One or more consoles have been configured to have a multi-system
message scope and either all routing codes or all routing codes except
routing code 11. Note: For active MCS and SMCS consoles, only the consoles
active on this system are checked. For inactive MCS and SMCS consoles, all
consoles are checked. All EMCS consoles are checked.
System Action: The system continues processing.
Operator Response: Report this problem to the system programmer.
System Programmer Response: To view the attributes of all consoles, issue the
following commands:
    DISPLAY CONSOLES,L
    DISPLAY EMCS,FULL,STATUS=L
Update the MSCOPE or ROUTCODE parameters of MCS and SMCS consoles ...
```

If a check finds an exception to a suggested value, or another potential problem, the check issues an exception message. The exception message might be accompanied by supporting information in report format. For an exception message, the system issues a WTO with just the message text by default. The system issues both the message text **and** details to the message buffer. This example shows the output from a check, including a report and an exception message in the message buffer (as continue on next foil).

EXCEPTION ...continued

Problem Determination: n/a

Source: Consoles (SC1CK)

Reference Documentation:

z/OS MVS Initialization and Tuning Reference

z/OS MVS System Commands

z/OS MVS Planning: Operations

Automation: n/a

Check Reason: Reduces the number of messages sent to a console in the sysplex

END TIME: 06/08/2005 09:49:17.451937 STATUS: EXCEPTION-LOW

Some considerations about exceptions

- You may have had your system configured a certain way for a Long Time, and now, IBM Health Checker for z/OS is telling you there are exceptions!
 - It's worth it to look over exceptions and make an evaluation. YOU need to decide what is appropriate for your system.
- Just because you get an exception, doesn't mean that there is a problem to report to IBM. Exceptions are meant to tell you about potential availability impacts.
- When you get an exception, it will be issued to:
 - Message buffer – you can view via SDSF, HZSPRINT or in log stream
 - WTO message – by default. Format is HZS message number then component message ID. Such as: *HZS0011 IXL0021*
- **No automatic correction of exceptions is done by IBM Health Checker for z/OS.**

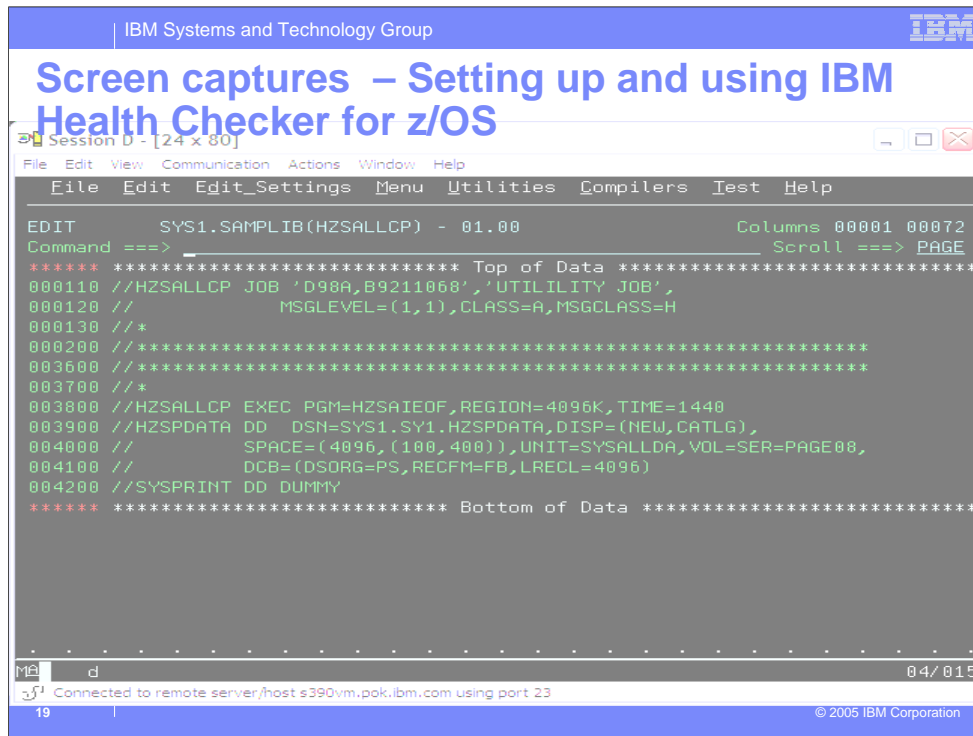


Hey! My system has been configured like this for years, and now I'm receiving exceptions!

Some customers may be startled by the exception messages that IBM Health Checker for z/OS issues on systems that have been running just fine the way they were. But it's really worth your time and attention to look over the exceptions and evaluate your system, because IBM Health Checker for z/OS reflects suggestions to help improve your system's availability and avoid problems. The checks reflect generally accepted recommendations, but you will need to evaluate whether each suggestion is appropriate for your system.

One important thing to note is that an exception does not imply that there is a problem to report to IBM. Exceptions are a means for you to evaluate potential availability impacts and take action, if appropriate.

Exception messages are the most important check output, because they identify potential problems and suggest a possible solution. The complete explanation and details for exception messages are issued to the message buffer, where you can view it with either SDSF, HZSPRINT, or in the log stream. By default, the exception message text is also issued as a WTO, prefaced by an HZS WTO message. The HZS message issued reflects the "SEVERITY" and "WTOTYPE" parameters defined for the check. (You can update these parameters to control the severity and descriptor code for the check.)



I'm using a z/OS R7 system for my demo. Here I'm allocating my IBM Health Checker for z/OS HZSPDATA data set, to save the check information.

The screenshot shows a terminal window titled "Session D - [24 x 80]" with a menu bar including File, Edit, View, Communication, Actions, Window, and Help. The main content area displays the following text:

```
EDIT          $$DRIVER.INFO(HC) - 01.01          Columns 00001 00072
Command ==>          Scroll ==> PAGE
***** ***** Top of Data *****
000001 //HCRACF  JOB 'D98A,B9211068','UTILITY JOB',
000002 //          MSGLEVEL=(1,1),CLASS=A,MSGCLASS=H
000003 //RACFTGT1 EXEC PGM=IKJEFT01
000004 //SYSPRT DD  SYSOUT=*
000005 //SYSTEM DD  DUMMY
000006 //SYSIN DD  *
000007     ADDUSER  +
000008         HZSPROC  +
000009         OMVS(UID(0) HOME('/ ') PROGRAM('/bin/sh'))  +
000010         NOPASSWORD  +
000011         NOIDCARD  +
000012         DFLTGRP(SYS1)
000013     CONNECT HZSPROC  +
000014         GROUP(SYS1)  +
000015         AUTH(USE)
000016     RDEFINE          +
000017         STARTED          +
000018     HZSPROC.*          +
```

At the bottom of the terminal, a status bar shows "ME d" and "09/002". A footer line reads "Connected to remote server/host s390vm.pok.ibm.com using port 23" and "© 2009 IBM Corporation".

Now, I submit a job to define the proper authority on my system. My IBM Health Checker for z/OS user ID is HZSPROC.

The screenshot shows a terminal window titled "Session D - [24 x 80]" with a menu bar including File, Edit, View, Communication, Actions, Window, and Help. The main content area displays the output of the command "VIEW SYS1.PARMLIB.INSTALL(HZSPRM00) - 01.00". The output shows the installation of the HZSPRM00 member, including the following lines:

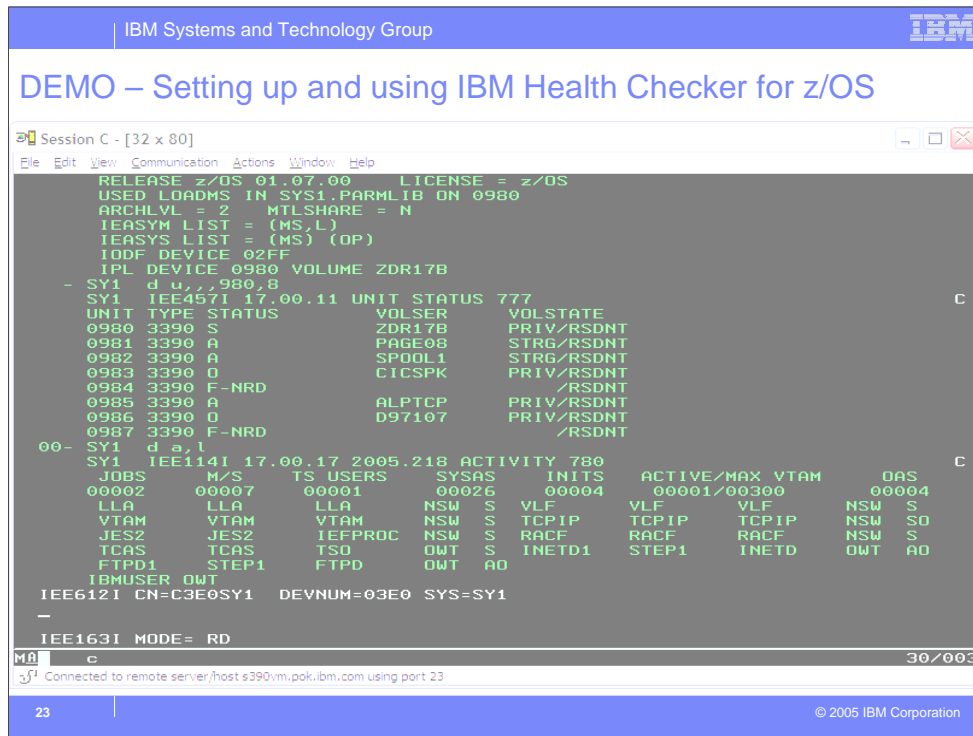
```
VIEW      SYS1.PARMLIB.INSTALL(HZSPRM00) - 01.00      Columns 00001 00072
Command ==>
***** ***** Top of Data *****
000001 /* ----- */
- - - - - 16 Line(s) not Displayed
000018 /*  LOGGER=ON,LOGSTREAMNAME=HZSLogstream          */
- - - - - 5 Line(s) not Displayed
000024 /* {ADD | ADDREPLACE},POLICY,STATEMENT=statementname,UPDATE,filters,*/
000025 /*      update_options,REASON=(reason text),DATE=yyyymmdd          */
- - - - - 63 Line(s) not Displayed
000089 /* ----- */
***** ***** Bottom of Data *****
```

The terminal window also shows a status bar at the bottom with "21" and "Connected to remote server/host s390vm.pok.ibm.com using port 23".

I'm using the default parmlib member, HZSPRM00 that was installed on my z/OS R7 system. Notice I'm not using system logger log streams for this demo, but it's a good idea to!

```
IBM Systems and Technology Group IBM  
  
DEMO – Setting up and using IBM Health Checker for z/OS  
  
Session D - [24 x 80]  
File Edit View Communication Actions Window Help  
File Edit Edit_Settings Menu Utilities Compilers Test Help  
  
VIEW SYS1.PROCLIB.POK(HZSPROC) - 01.00 Columns 00001 00072  
Command ==> Scroll ==> PAGE  
***** Top of Data *****  
000001 //HZSPROC JOB JESLOG=SUPPRESS  
000002 //*****/  
- - - - - 32 Line(s) not Displayed  
000035 //*****/  
000036 //HZSPROC PROC HZSPRM='00'  
000037 //HZSSTEP EXEC PGM=HZSINIT,REGION=0K,TIME=NOLIMIT,  
000038 // PARM='SET PARMLIB=&HZSPRM'  
000039 //HZSPDATA DD DSN=SYS1.&SYSNAME..HZSPDATA,DISP=OLD  
000040 // PEND  
000041 // EXEC HZSPROC  
***** Bottom of Data *****  
  
MA d 09/002  
Connected to remote server/host s390vm.pok.ibm.com using port 23
```

I copied my IBM Health Checker for z/OS started task into a proclib in my concatenation. I verify that my HZSPRM00 is being used, and that that my HZSPDATA data set name is correct for my system.



This just shows that I'm on a z/OS R7 system, and I don't yet have IBM Health Checker for z/OS started...

IBM Systems and Technology Group

DEMO – Setting up and using IBM Health Checker for z/OS

Session C - [32 x 80]

```

File Edit View Communication Actions Window Help
- SY1 HZS0010I THE HZSPDATA DATA SET CONTAINS NO RECORDS
SY1 HZS0403I SET PARMLIB PROCESSING HAS BEEN COMPLETED
- SY1 HZS0103I HZSPROC INITIALIZATION COMPLETE
SY1 HZS0001I CHECK(IBMUSS,USS_MAXSOCKETS_MAXFILEPROC):
BPXH032E MaxFileProc value is too low.
SY1 HZS0001I CHECK(IBMUSS,USS_MAXSOCKETS_MAXFILEPROC):
BPXH033E MaxSockets value for AF_INET is too low.
SY1 HZS0001I CHECK(IBM CNZ,CNZ_AMRF_EVENTUAL_ACTION_MSGS):
CNZHF0004I Retaining eventual action messages may consume storage
needed by critical or immediate action messages.
SY1 HZS0002E CHECK(IBM XCF,XCF_TCLASS_CLASSLEN):
IXCH0420E Transport class definitions do not provide sufficient size
segregation
SY1 HZS0001I CHECK(IBM CNZ,CNZ_CONSOLE_ROUTCODE_11):
CNZHF0005I One or more consoles are configured to receive messages
intended only for programmers.
SY1 HZS0001I CHECK(IBM RSM,RSM_MEMLIMIT):
IARH109E MEMLIMIT SET TO MAXIMUM
SY1 HZS0002E CHECK(IBM SDUMP,SDUMP_AUTO_ALLOCATION):
IEAH701I SDUMP is not using automatic allocation.
SY1 HZS0001I CHECK(IBM RSM,RSM_REAL):
IARH101E V=R Storage is Defined
SY1 HZS0001I CHECK(IBM CNZ,CNZ_SYSCONS_ROUTCODE):
CNZHF0008I System console SY1 is not configured to receive the
minimum set of routing codes (1, 2, and 10).
00 *SY1 *HZS0003E CHECK(IBM RACF,RACF_SENSITIVE_RESOURCES):
*IRRH204E The RACF_SENSITIVE_RESOURCES check has found one or more
*potential errors in the security controls on this system.
IEE612I CN=C3E0SY1 DEVNUM=03E0 SYS=SY1

IEE163I MODE= RD
MA c 30/003
Connected to remote server/host s390vm.pok.ibm.com using port 23
24 © 2005 IBM Corporation

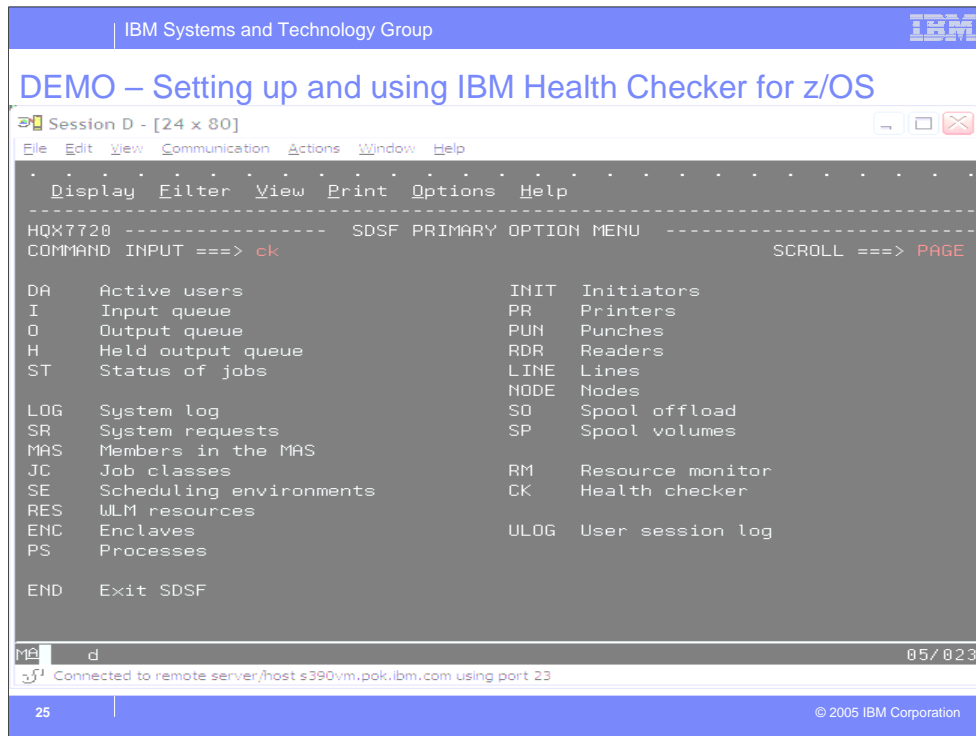
```

I issued “S HZSPROC,HZSPRM=00” and a flurry of activity happened! HZSPROC started, and did all the appropriate checks.

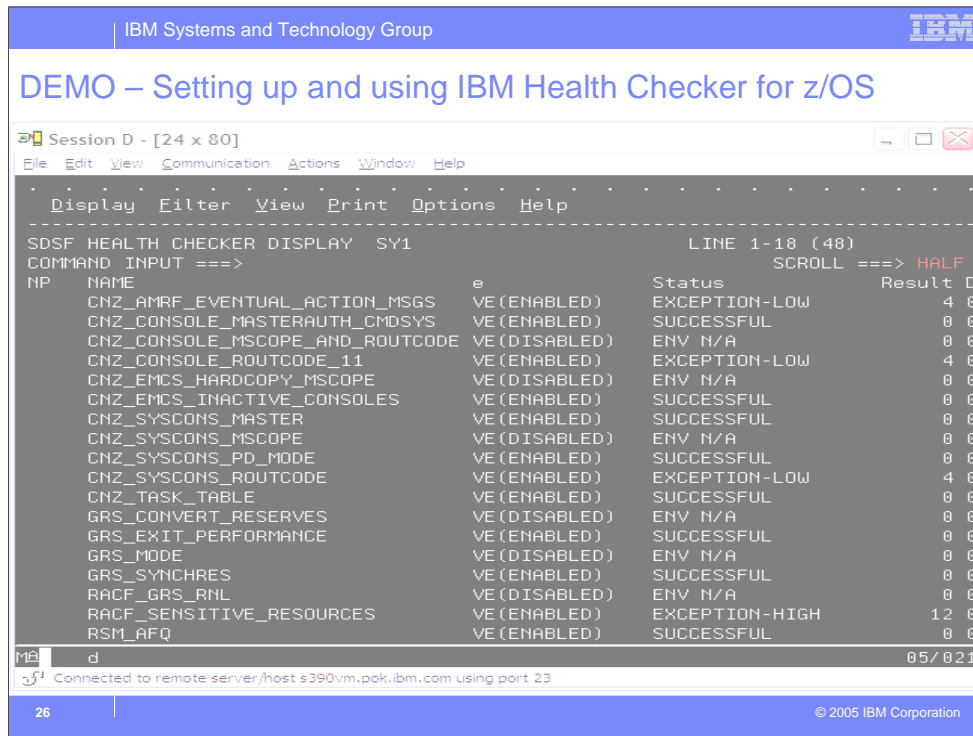
You see that I received the message that my HZSPDATA data set was initially empty – which is fine.

Also, notice HZS0103I – that HZSPROC initialization was complete.

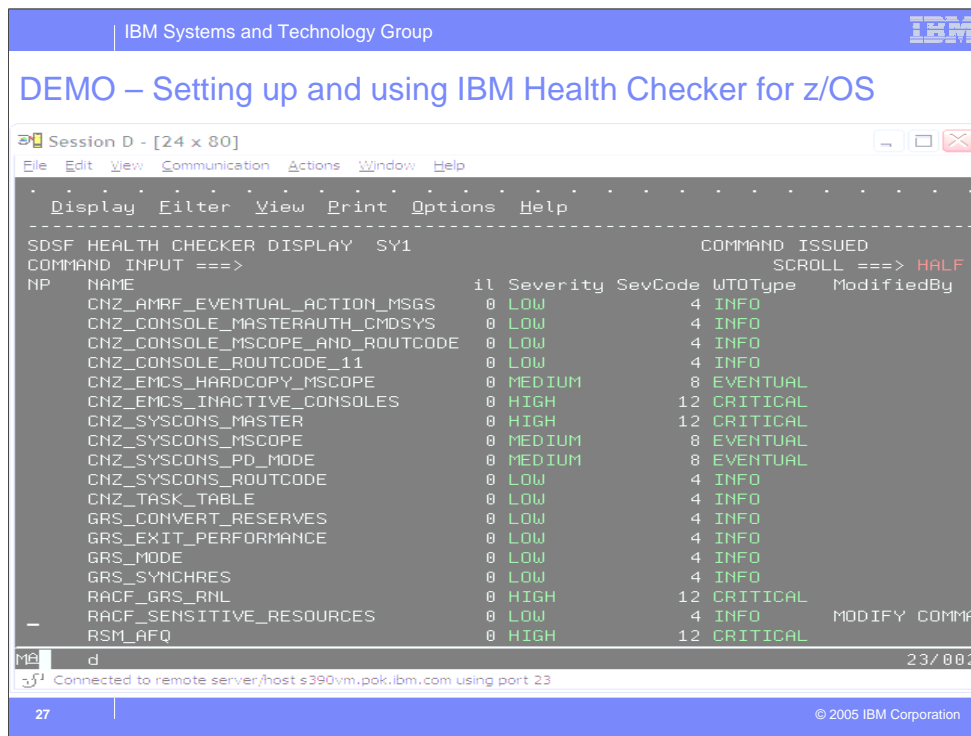
Many checks ran, and you can see that I had a high severity exception, which shows as a red WTO message.



Now, I want to look more at the checks that ran, so I go into SDSF and issue the CK command.



Yup, there's that high exception RACF one. I'm going to do some modifying of that one...



For the purposes of learning, I overtype the severity and make it LOW (from HIGH) and change the WTOType to be INFO (from CRITICAL). I hit enter, and see that it was modified.

IBM Systems and Technology Group

DEMO – Setting up and using IBM Health Checker for z/OS

Session D - [24 x 80]

File Edit View Communication Actions Window Help

Display Filter View Print Options Help

```

SDSF HEALTH CHECKER DISPLAY SY1
COMMAND INPUT ==>
COMMAND ISSUED
SCROLL ==> HALF
NP NAME tart-Date-Time Interval NextSch-Date-T
CNZ_AMRF_EVENTUAL_ACTION_MSGS 8/06/2005 17:02:14 24:00 08/07/2005 17:
CNZ_CONSOLE_MASTERAUTH_CMDSYS 8/06/2005 17:02:14 24:00 08/07/2005 17:
CNZ_CONSOLE_MSCOPE_AND_ROUTCODE 8/06/2005 17:02:14 24:00 ***** N/A ***
CNZ_CONSOLE_ROUTCODE_11 8/06/2005 17:02:14 24:00 08/07/2005 17:
CNZ_EMCS_HARDCOPY_MSCOPE 8/06/2005 17:02:14 24:00 ***** N/A ***
CNZ_EMCS_INACTIVE_CONSOLES 8/06/2005 17:02:14 24:00 08/07/2005 17:
CNZ_SYSCONS_MASTER 8/06/2005 17:02:14 24:00 08/07/2005 17:
CNZ_SYSCONS_MSCOPE 8/06/2005 17:02:14 24:00 ***** N/A ***
CNZ_SYSCONS_PD_MODE 8/06/2005 17:02:14 1:00 08/06/2005 18:
CNZ_SYSCONS_ROUTCODE 8/06/2005 17:02:14 24:00 08/07/2005 17:
CNZ_TASK_TABLE 8/06/2005 17:17:14 0:15 08/06/2005 17:
GRS_CONVERT_RESERVES 8/06/2005 17:02:14 ONETIME ***** N/A ***
GRS_EXIT_PERFORMANCE 8/06/2005 17:02:14 24:00 08/07/2005 17:
GRS_MODE 8/06/2005 17:02:14 ONETIME ***** N/A ***
GRS_SYNCHRES 8/06/2005 17:02:14 1:00 08/06/2005 18:
RACF_GRS_RNL 8/06/2005 17:02:14 0:00 ***** N/A ***
RACF_SENSITIVE_RESOURCES 8/06/2005 17:19:27 0:01 08/06/2005 17:
RSM_AFO 8/06/2005 17:02:14 ONETIME ***** N/A ***

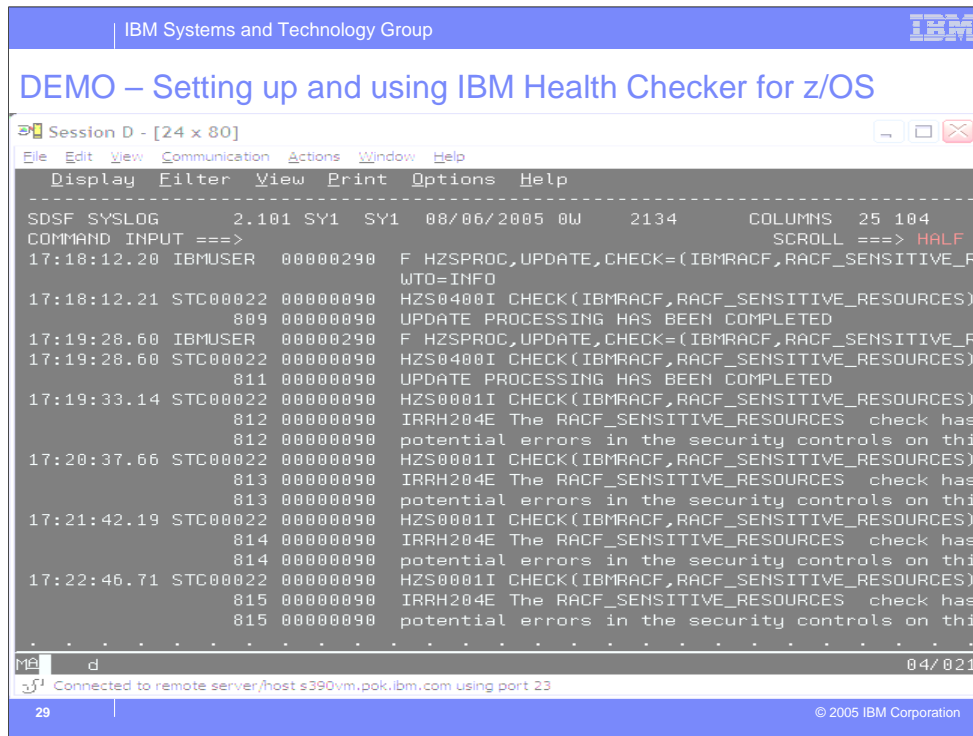
```

MA d 23/002

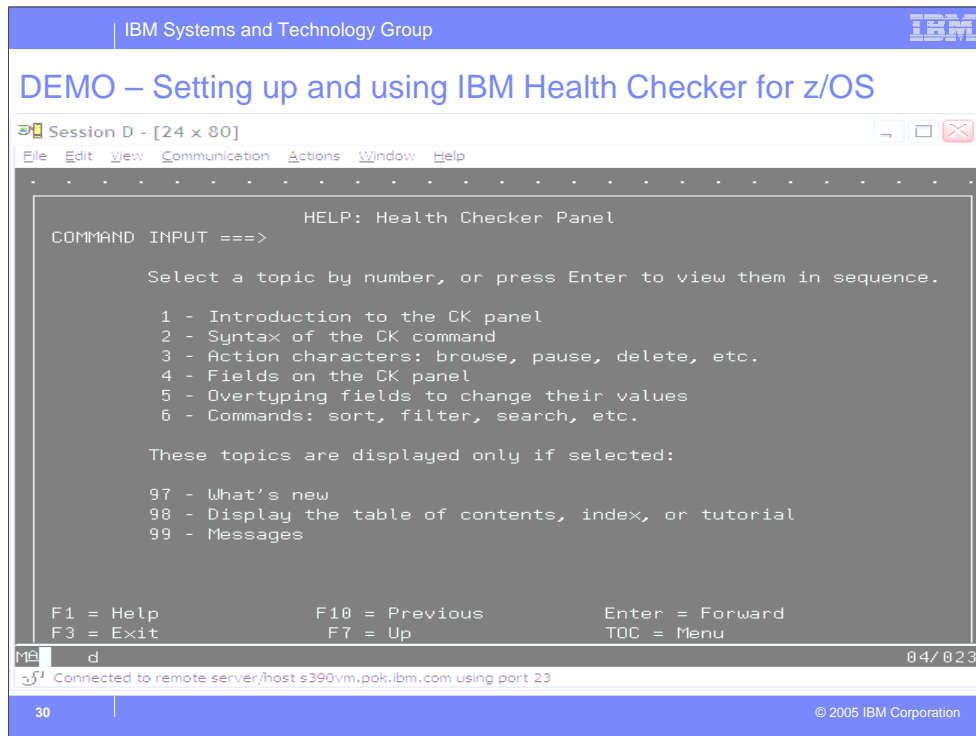
Connected to remote server/host s390vm.pok.ibm.com using port 23

28 © 2005 IBM Corporation

I'm going to do some more changes on that check...I make it execute every minute. I changed it to 0:01 from 4:00 (every four hours).



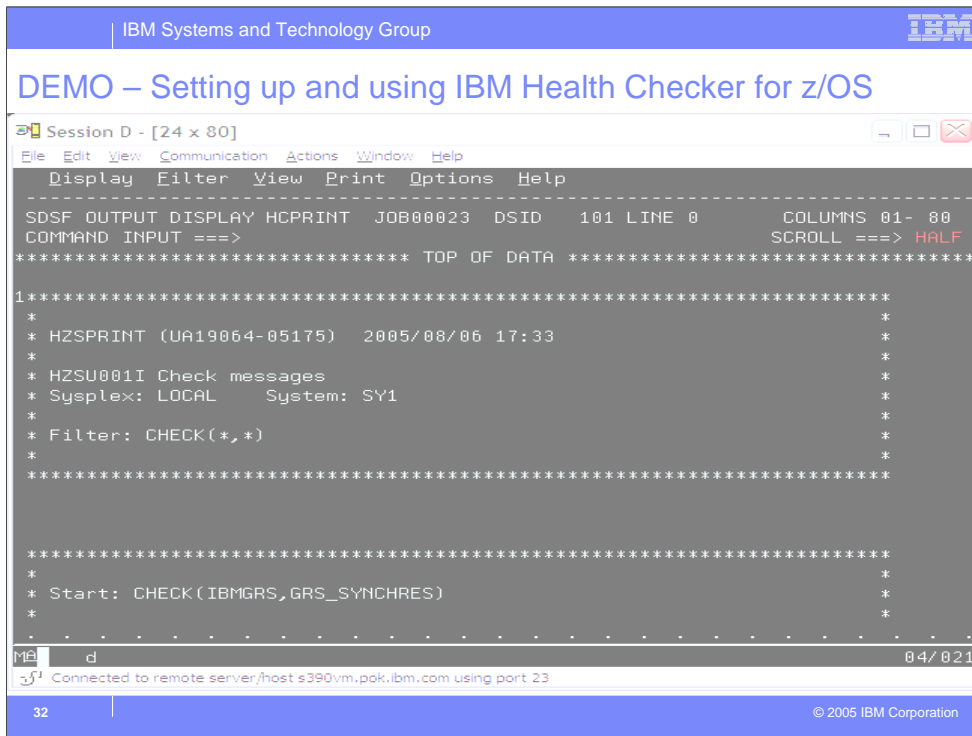
This time, I go into the syslog in SDSF, and look and see what's happening. That check has been executing every minute, just like I told it to. Also, it's not a high severity one anymore, and it's not a red WTO message on my console.



There's so much you can do in SDSF! Just type **HELP** from the CK panel and start exploring!



One more thing I want to try out – the HZSPRINT utility. I specify that I want to see the output from all the checks...



...and look at the SYSOUT. Looks as I expected.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

DB2*	System z
DB2 Connect	Tivoli*
DB2 Universal Database	VM/ESA*
e-business logo	WebSphere*
GDPS*	z/OS*
Geographically Dispersed Parallel Sysplex	z/VM*
HyperSwap	zSeries*
IBM*	
IBM eServer	
IBM logo*	
Parallel Sysplex*	

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Intel is a registered trademark of the Intel Corporation in the United States, other countries or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.