Z/OS 1.7: IBM Health Checker for z/OS

Mike Phillips

IBM Health Checker for z/OS Development
mphillip@us.ibm.com
845-435-5646

Trademarks

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

IBM *
eServer *
z/OS *
z/OS Unix System Services
RACF *

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
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.

Agenda

- z/OS 1.7 IBM Health Checker for z/OS functional overview:
  - Component and its interfaces
  - Solution – set of components delivering supporting function
  - Programmable interfaces for end user exploitation
- Technical Q & A

z/OS 1.7: IBM Health Checker for z/OS

History

- **Multi system outage analysis**
  - 15-20% system outages attributed to Setup/Config
- **Outage avoidance initiative**
  - Health checker was a tool developed to address component configuration and setup errors commonly made by installations
  - **Initial version:**
    - **Unsupported tool** available to customer installations via web download
      - Implemented as **Batch job** to provide:
      - Component level information
      - Configuration/customization related component warnings
    - **37 Component checks** available Feb-July of 2003
  - **3000+ downloads to date**
- **Great customer interest in Health checker tool**
  - Many installation policies prevent use of non-supported code in the production environment
  - z/BLC customers generate high priority requirement for type 1 product solution incorporated into z/OS

z/OS 1.7: IBM Health Checker for z/OS

- **Great customer interest in Health checker tool**
  - Many installation policies prevent use of non-supported code in the production environment
    - More formal product, with more formal service support
    - More checks from more components and products
    - Checks from ISV products
    - Checks they write themselves
  - z/BLC customers generate high priority requirement for type 1 product solution incorporated into z/OS

IBM Health Checker for z/OS – Highlights

- IBM Health Checker for z/OS
  - Framework implementation
  - z/OS Component Health Check programs
    - Ported & new checks
  - Programmable Interfaces
  - Command interfaces
  - Setup samples, utilities
  - MVS parmlib support
  - Logger support
  - CTRACE Support

z/OS 1.7: IBM Health Checker for z/OS

- **IBM Health Checker for z/OS is integrated into z/OS R1.7**
  - Backbone infrastructure shipped in its own unique FMID
    - HZS7720
  - Can be installed on R1.4 and above
    - R1.4 – R1.6 is obtained via web deliverable concurrent with z/OS 1.7 GA (9/2005)
      - **www.ibm.com/servers/eserver/zseries/zos/hchecker/**
  - Health Check programs
    - Shipped with the individual components that own them
    - R1.4 – R1.6 checks shipped via APARs
    - Virtual PSP bucket utilized for complete list of Component Health check program APARs
      - **http://techsupport.services.ibm.com/390/psp_main.html**

z/OS 1.7: IBM Health Checker for z/OS

- **z/OS 1.7 Integrated Health Checker solution includes SDSF support for browsing the output of checks and ease of check management**
  - R1.4 – R1.6 support shipped via APARs

IBM Health Checker for z/OS – The components

- Server address space framework
  - Provides a framework for any software on z/OS to instantiate, execute, and manage checks
- Health Check programs
  - Compare the current system settings against ‘best practices’
    - Check programs look for problems and report them
  - Health check programs are separate entities

IBM Health Checker for z/OS – The components (cont.)

- Programmable interfaces
  - Allow customer installation/vendor developed checks
- Command functions
  - Control and manipulate Health Checker backbone framework and Component health check programs
- Samples
  - Provides ease of set up
    - JCL, Utilities, Procs, Parmlib etc.
- Parmlib support
  - Pervasive check parameter overrides and POLICY statements
- Logstream support
  - Provides capability for historical check messages
- CTRACE Support

IBM Health Checker for z/OS – The components (cont.)

Fifty-five z/OS Component health check programs

- Ported and New
- Health check programs ship with representative component
- -Separately packaged and shipped
  - -Do not need to be predefined to health checker
  - -Can be local to a system or global in Sysplex
  - -Can run on timed basis or as a one time check
    - Interval can be customized
  - -Customer chooses which checks to run
    - All checks enabled to run as default
  - Support for user written checks
- Health check programs register a dynamic exit routine to define Health Check
- IBM Health Checker for z/OS STC registers with DYNEXIT when HC A.S. Activates
  - Health check programs do not execute if IBM Health Checker for z/OS not active

IBM Health Checker for z/OS – Component health check programs

- Ported health check programs –
  - In some cases, there may not be a one to one correspondence to checks ported from the prototype tool
  - Consoles (IBMCNZ)

1. CNZ_AMRF_EVENTUAL_ACTION_MSGS
2. CNZ_CONSOLE_MASTERAUTH_CMDSYS
3. CNZ_CONSOLE_MSCOPE_AND_ROUTCODE
4. CNZ_CONSOLE_ROUTCODE_11
5. CNZ_EMCS_HARDCOPY_MSCOPE
6. CNZ_EMCS_INACTIVE_CONSOLES
7. CNZ_SYSCONS_MASTER
8. CNZ_SYSCONS_MSCOPE
9. CNZ_SYSCONS_PD_MODE
10. CNZ_SYSCONS_ROUTCODE
11. CNZ_TASK_TABLE (**new**, 1.7 only)

IBM Health Checker for z/OS – Consoles Health Check summary

- **Most Consoles checks apply to z/OS 1.4 +**
- **Defaults can be overridden by POLICY statement or Modify command**
- **Consoles Health Check Summary**
  - Eventual Action messages not retained when AMRF active
  - Assign MASTER auth and proper cmd association to MCS, EMCS, SMCS consoles
  - All MCS, EMCS, SMCS consoles defined with multi-system message scope should only receive routing codes specific to that console's function
  - MCS/SMCS consoles should not receive ROUTCODE 11 messages
  - EMCS consoles defined with multi-system message scope should not receive the hardcopy message set
  - Excessive number of inactive EMCS consoles
  - System console has Master authority
  - System console has single system message scope
  - System console not in problem determination mode (PD)
  - System console (minimally) receiving routing codes 1,2 & 10
  - Report status of important tasks running in Console asid

IBM Health Checker for z/OS – Component check programs

- GRS (IBMGRS)

1. GRS_CONVERT_RESERVES
2. GRS_EXIT_PERFORMANCE
3. GRS_MODE (**new**)
4. GRS_SYNCHRES (**new**)

IBM Health Checker for z/OS – GRS Health Check summary

- **GRS checks apply to z/OS 1.4 +**
- **Defaults can be overridden by POLICY statement or Modify command**
- **GRS Health Check Summary**
  - Mode of the GRS complex – STAR recommended
  - GRS synchronous reserve processing can prevent deadlock conditions
  - Checks if RESERVEs are being converted to global ENQs – STAR mode
  - Certain GRS dynamic exits can degrade performance

IBM Health Checker for z/OS – Component check programs

- RACF (IBMRACF)

1. RACF_GRS_RNL (**new**)
2. RACF_SENSITIVE_RESOURCES (**new**)

IBM Health Checker for z/OS – RACF Health Check summary

- **RACF checks apply to z/OS 1.4 +**
- **Defaults can be overridden by POLICY statement or Modify command**
- **RACF Health Check Summary**
  - Converting RACF System ENQs to System ENQs can corrupt the RACF dataset
  - System is critically exposed if system-critical datasets are not properly protected

IBM Health Checker for z/OS – Component check programs

- **SDUMP (IBMSDUMP)**

1. SDUMP_AVAILABLE
2. SDUMP_AUTO_ALLOCATION

IBM Health Checker for z/OS – SDUMP Health Check summary

- **SDUMP checks apply to z/OS 1.4 +**
- **Defaults can be overridden by POLICY statement or Modify command**
- **SDUMP Health Check Summary**
  - Ensure SDUMP is enabled to collect SVC dumps
  - Automatic allocation of SVC dumps is enabled

IBM Health Checker for z/OS – Component check programs

- z/OS Unix Systems Services (IBMUSS)

1. USS_FILESYS_CONFIG
2. USS_AUTOMOUNT_DELAY (**new**)
3. USS_MAXSOCKETS_MAXFILEPROC **(new)**

IBM Health Checker for z/OS – USS Health Check summary

- **USS checks apply to z/OS 1.4 +**
- **Defaults can be overridden by POLICY statement or Modify command**
- **USS Health Check Summary**
  - File system configuration evaluation
  - MaxSockets & MaxFileProc values
  - Sysplex Automount delay configuration values

IBM Health Checker for z/OS – Component check programs

- Virtual Storage Management (IBMVSM)

1. VSM_CSA_LIMIT
2. VSM_SQA_LIMIT
3. VSM_PVT_LIMIT
4. VSM_CSA_CHANGE
5. VSM_CSA_THRESHOLD
6. VSM_SQA_THRESHOLD

IBM Health Checker for z/OS – Component check programs

- Real Storage Management (IBMRSM)

1. RSM_REAL
2. RSM_RSU
3. RSM_AFQ (Available Frame queues)
4. RSM_MEMLIMIT (**new**)
5. RSM_MAXCADS (**new**)
6. RSM_HVSHARE (**new**)

IBM Health Checker for z/OS – Storage Health Check summary

- **RSM/VSM/ASM checks apply to z/OS 1.4 +**
- **Defaults can be overridden by POLICY statement or Modify command**
- **Storage Health Check Summary**
  - Checks HVSHARE (high virtual shared area) isn't too low
  - SMFPRMxx MEMLIMIT value
  - MAXCADS value and # of in use data spaces
  - Available threshold queue values are too low
  - IEASYSxx "REAL" setting
  - Current CSA & SQA size against minimum suggested value
  - Private storage size
  - CSA & SQA threshold value against current allocations
  - Changes in size of CSA or private, since last IPL

IBM Health Checker for z/OS – Component check programs

- XES/XCF (IBMXCF)

1. XCF_CDS_SEPARATION
2. XCF_CF_STR_PREFLIST
3. XCF_CF_STR_EXCLLIST
4. XCF_TCLASS_HAS_UNDESIG
5. XCF_TCLASS_CONNECTIVITY
6. XCF_TCLASS_CLASSLEN
7. XCF_SIG_CONNECTIVITY
8. XCF_DEFAULT_MAXMSG
9. XCF_MAXMSG__NUMBUF_RATIO
10. XCF_SIG_PATH_SEPARATION
11. XCF_CF_CONNECTIVITY
12. XCF_FDI
13. XCF_SFM_ACTIVE
14. XCF_CLEANUP_VALUE
15. XCF_SYSPLEX_CDS_CAPACITY **(new)**
16. XCF_SIG_STR_SIZE **(new)**

IBM Health Checker for z/OS – XES/XCF Health Check summary (cont.)

- **XES/XCF checks apply to z/OS 1.4 +**
- **Defaults can be overridden by POLICY statement or Modify command**
- **XES/XCF Health Check Summary**
  - System single point of failure avoidance: links & connectivity to CFs
  - Check status of Sysplex Failure Management (SFM)
  - XCF clean-up interval setting recommendation
  - SYSPLEX, CFRM & LOGR primary CDS's reside on different volumes
  - Ensure SYSPLEX CDS is large enough for growth of System, Groups & Members
  - All transport classes setup to service pseudo group 'UNDESIG'
  - All defined pathout classes have minimum 1 pathout per target system

IBM Health Checker for z/OS – XES/XCF Health Check summary

- **XES/XCF checks apply to z/OS 1.4 +**
- **Defaults can be overridden by POLICY statement or Modify command**
- **XES/XCF Health Check Summary**
  - Minimum of 2 different transport classes with unique lengths
  - Multiple pathin / pathout pairs in the working state for each system
  - Minimum MAXMSG value of 2000 for each transport class
  - Each inbound signal path has buffer space to receive 30 simultaneous messages
  - Single point of failure for paths to all connected systems
  - Enough signalling structures to support full SYSPLEX connectivity
  - Each structure is allocated in accordance with CFRM policy preference list
  - Each structure is excluded in accordance with exclusion list

IBM Health Checker for z/OS – Component check programs

- RRS (IBMRRS)

1. RRS_RMDATALOGDUPLEXMODE (**new**)
2. RRS_RMDOFFLOADSIZE (**new**)
3. RRS_MUROFFLOADSIZE (**new**)
4. RRS_DUROFFLOADSIZE (**new**)
5. RRS_RSTOFFLOADSIZE (**new**)

IBM Health Checker for z/OS – RRS Health Check summary

- **RRS checks apply to z/OS 1.4 +**
- **Defaults can be overridden by POLICY statement or Modify command**
- **RRS Health Check Summary**
  - Duplexing scheme for RM Data Log Stream
  - Size of Delayed UR log’s offload dataset
  - Size of Main UR log’s offload dataset
  - Size of the Restart log’s offload dataset

IBM Health Checker for z/OS – Component check programs

- Contents Supervisor (IBMCSV)

1. CSV_APF_EXISTS
2. CSV_LNKLST_SPACE
3. CSV_LNKST_NEWEXTENTS

IBM Health Checker for z/OS – Contents Supervisor Health Check summary

- **CSV checks apply to z/OS 1.4 +**
  - **z/OS 1.4-1.7 CSV Health Check APAR OA12777 not Cor-closed**
- **Defaults can be overridden by POLICY statement or Modify command**
- **CSV Health Check Summary**

  – Determine if a LNKLST contains a data set that has expanded into new extents

  – Checks for LNKLST data set(s) that could expand into new extents

  – Checks that data sets described by entries in the APF list are consistent with data sets that exist on the system

IBM Health Checker for z/OS – Obtaining new checks

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

- New checks will be provided on an ongoing basis in the service stream
- To obtain the most recent 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.

IBM Health Checker for z/OS – Publications

- IBM Health Checker for z/OS User's Guide

// http: www.ibm.com/servers/eserver/zseries/zos/hchecker

IBM Health Checker for z/OS – Managing the components of Health Checker: Command Interface

- **Command Interface Support**
  - ➢ **MVS Modify command**
    - ✓ Manage Health Check programs with command parameter functions
      - – Activate, deactivate, run, update, delete, addnew,display…
      - – Many sub-parameter functions supported
      - – Wild carding supported for parameter value specifications
      - – ‘F *hzsproc, < command_parameter* >,CHECK=(**IBM*,***) ,…’
    - ✓ Manage Health Checker Started Task A.S.
      - – Display, Refresh, Stop, Logger, etc.
  - ➢ **New SDSF “CK” panel**
    - – Actions and overtypes perform command functions

IBM Health Checker for z/OS: Component Check Program Output

- **Three kinds of messages can result in component check program output**
  - **Exception Messages**
  - **System OK messages**
    - **No exception found**
  - **Report Messages**

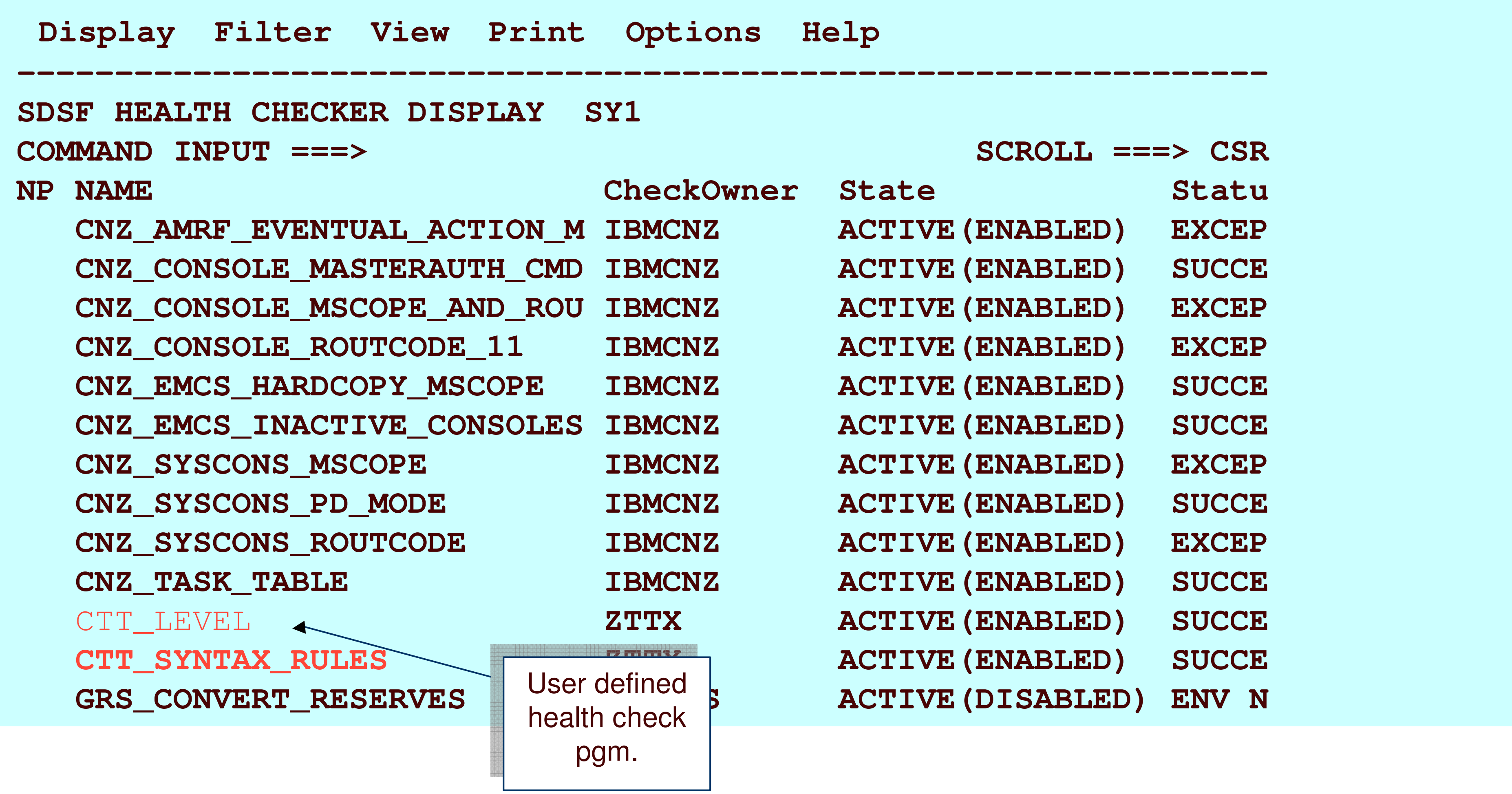
IBM Health Checker for z/OS: Component Check Program Output

- **Exception messages**
  - WTO with summary text issued to Console / Syslog
    - Entire message output written to message buffer
      - Viewed with HZSPRINT utility or SDSF
  - WTO issued based on check severity
    - HI - HZS0003E issued as critical eventual action message
    - MED - HZS0002E issued as eventual action message
    - LOW - HZS0001I, issued as informational message
    - None, HZS0004I, issued as hardcopy only message
    - Default Severities of checks can be modified via operator command or parmlib
- **System OK messages**
  - Indicate an exception was NOT found
    - Message buffer only

IBM Health Checker for z/OS: Component Check Program Output (continued)

- **Report messages**
  - Messages written to message buffer
  - Report messages can be used to report on system-neutral information.
    - VSM "VSM_CSA_CHANGE" check is used to identify 5 highest users of common storage – may or may not be of concern
  - Also used to generate detail information about exceptions

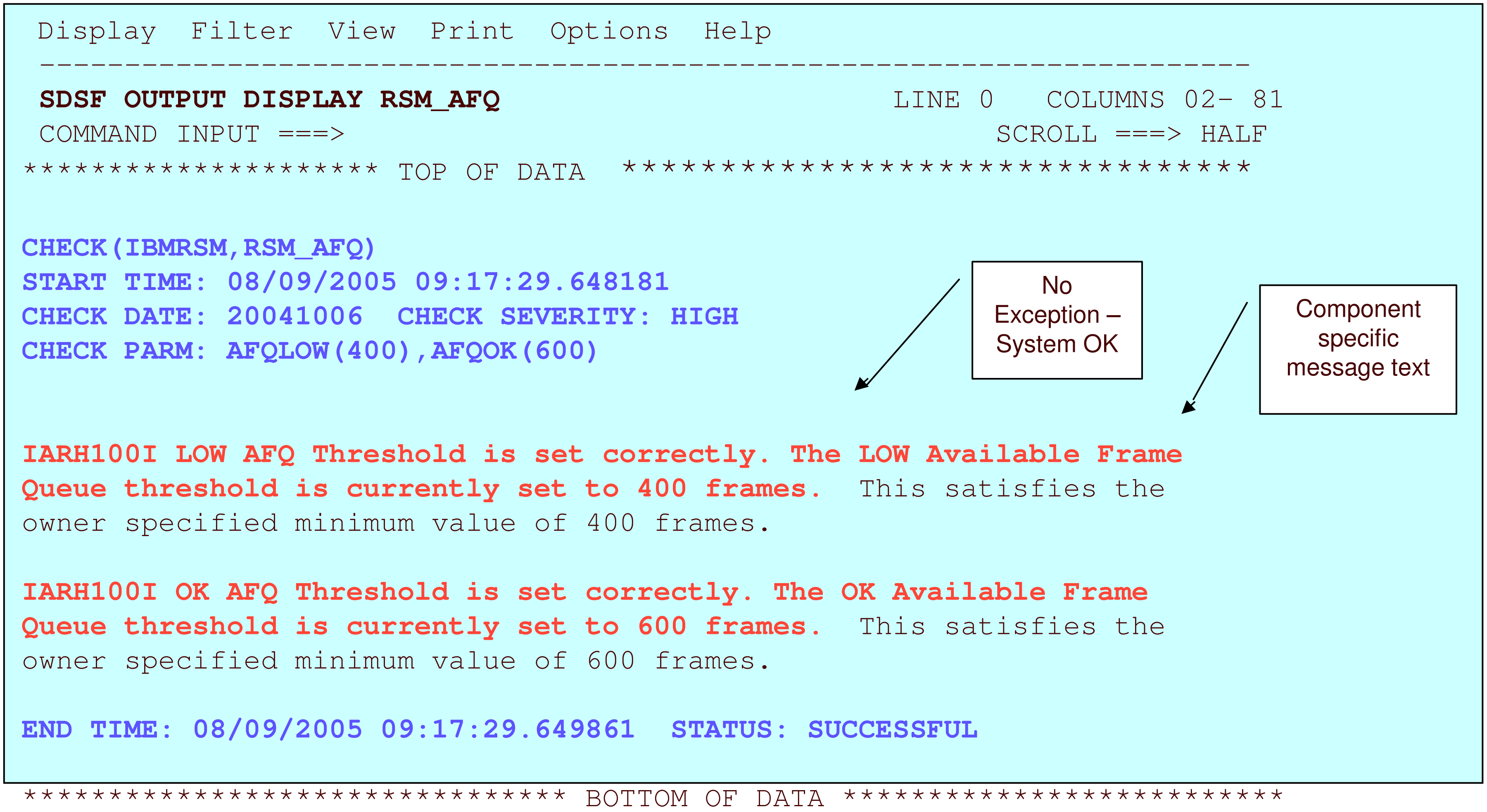
IBM Health Checker for z/OS – Managing the components of Health Checker -SDSF "CK" panel



```
 Display  Filter  View  Print  Options  Help
----------------------------------------------------------------
SDSF HEALTH CHECKER DISPLAY  SY1
COMMAND INPUT ===>                               SCROLL ===> CSR
NP NAME                       CheckOwner  State            Statu
   CNZ_AMRF_EVENTUAL_ACTION_M IBMCNZ      ACTIVE(ENABLED)  EXCEP
   CNZ_CONSOLE_MASTERAUTH_CMD IBMCNZ      ACTIVE(ENABLED)  SUCCE
   CNZ_CONSOLE_MSCOPE_AND_ROU IBMCNZ      ACTIVE(ENABLED)  EXCEP
   CNZ_CONSOLE_ROUTCODE_11    IBMCNZ      ACTIVE(ENABLED)  EXCEP
   CNZ_EMCS_HARDCOPY_MSCOPE   IBMCNZ      ACTIVE(ENABLED)  SUCCE
   CNZ_EMCS_INACTIVE_CONSOLES IBMCNZ      ACTIVE(ENABLED)  SUCCE
   CNZ_SYSCONS_MSCOPE         IBMCNZ      ACTIVE(ENABLED)  EXCEP
   CNZ_SYSCONS_PD_MODE        IBMCNZ      ACTIVE(ENABLED)  SUCCE
   CNZ_SYSCONS_ROUTCODE       IBMCNZ      ACTIVE(ENABLED)  EXCEP
   CNZ_TASK_TABLE             IBMCNZ      ACTIVE(ENABLED)  SUCCE
   CTT_LEVEL                  ZTTX        ACTIVE(ENABLED)  SUCCE
   CTT_SYNTAX_RULES           ZTTX        ACTIVE(ENABLED)  SUCCE
   GRS_CONVERT_RESERVES            S      ACTIVE(DISABLED) ENV N
```

User defined health check pgm.

IBM Health Checker for z/OS – Managing the components of Health Checker -SDSF Browse of “Successful” check output (System OK)

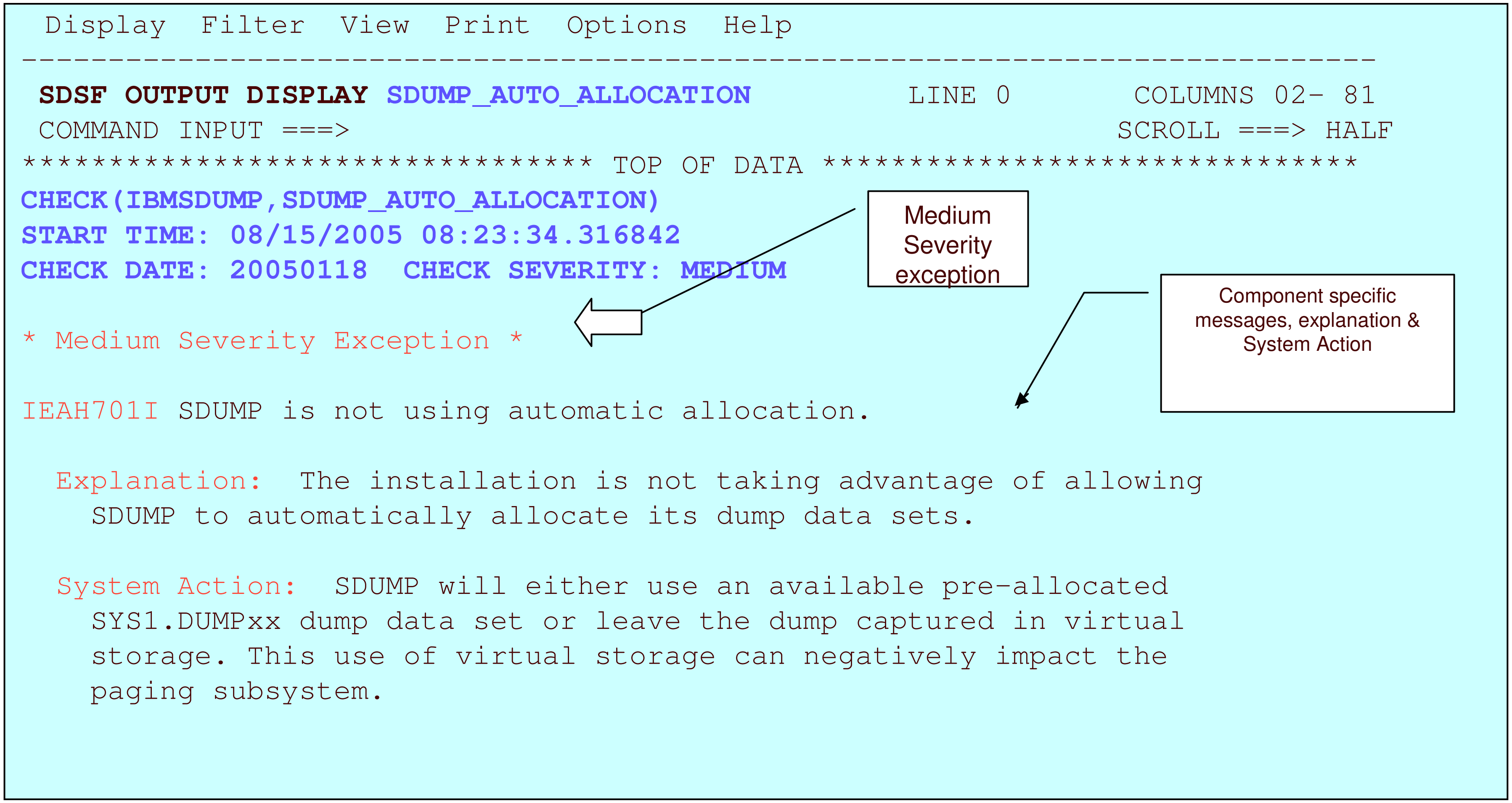


```
 Display  Filter  View  Print  Options  Help
 -----------------------------------------------------------------------
 SDSF OUTPUT DISPLAY RSM_AFQ                            LINE 0    COLUMNS 02- 81
 COMMAND INPUT ===>                                           SCROLL ===> HALF
******************** TOP OF DATA  ************************************

CHECK(IBMRSM,RSM_AFQ)
START TIME: 08/09/2005 09:17:29.648181
CHECK DATE: 20041006  CHECK SEVERITY: HIGH
CHECK PARM: AFQLOW(400),AFQOK(600)


IARH100I LOW AFQ Threshold is set correctly. The LOW Available Frame
Queue threshold is currently set to 400 frames.  This satisfies the
owner specified minimum value of 400 frames.

IARH100I OK AFQ Threshold is set correctly. The OK Available Frame
Queue threshold is currently set to 600 frames.  This satisfies the
owner specified minimum value of 600 frames.

END TIME: 08/09/2005 09:17:29.649861  STATUS: SUCCESSFUL

******************************* BOTTOM OF DATA *************************
```

No Exception – System OK

Component specific message text

IBM Health Checker for z/OS – Managing the components of Health Checker -SDSF
Browse of “Exception” check output (1 of 3 SDUMP)

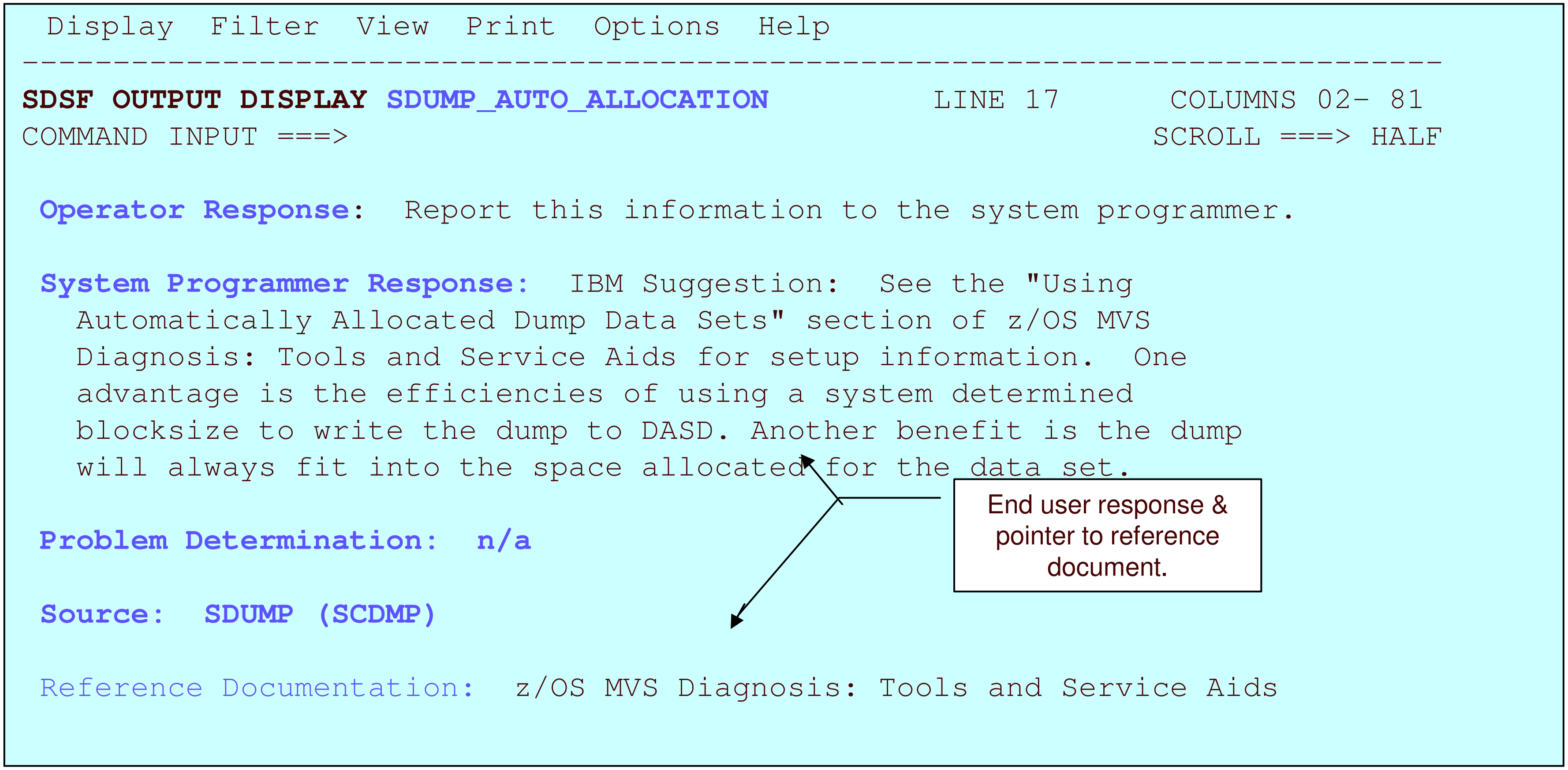


```
 Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY SDUMP_AUTO_ALLOCATION         LINE 0       COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> HALF
********************************* TOP OF DATA ********************************
CHECK(IBMSDUMP,SDUMP_AUTO_ALLOCATION)
START TIME: 08/15/2005 08:23:34.316842
CHECK DATE: 20050118  CHECK SEVERITY: MEDIUM

* Medium Severity Exception *

IEAH701I SDUMP is not using automatic allocation.

  Explanation:  The installation is not taking advantage of allowing
    SDUMP to automatically allocate its dump data sets.

  System Action:  SDUMP will either use an available pre-allocated
    SYS1.DUMPxx dump data set or leave the dump captured in virtual
    storage. This use of virtual storage can negatively impact the
    paging subsystem.
```

Medium Severity exception

Component specific messages, explanation & System Action

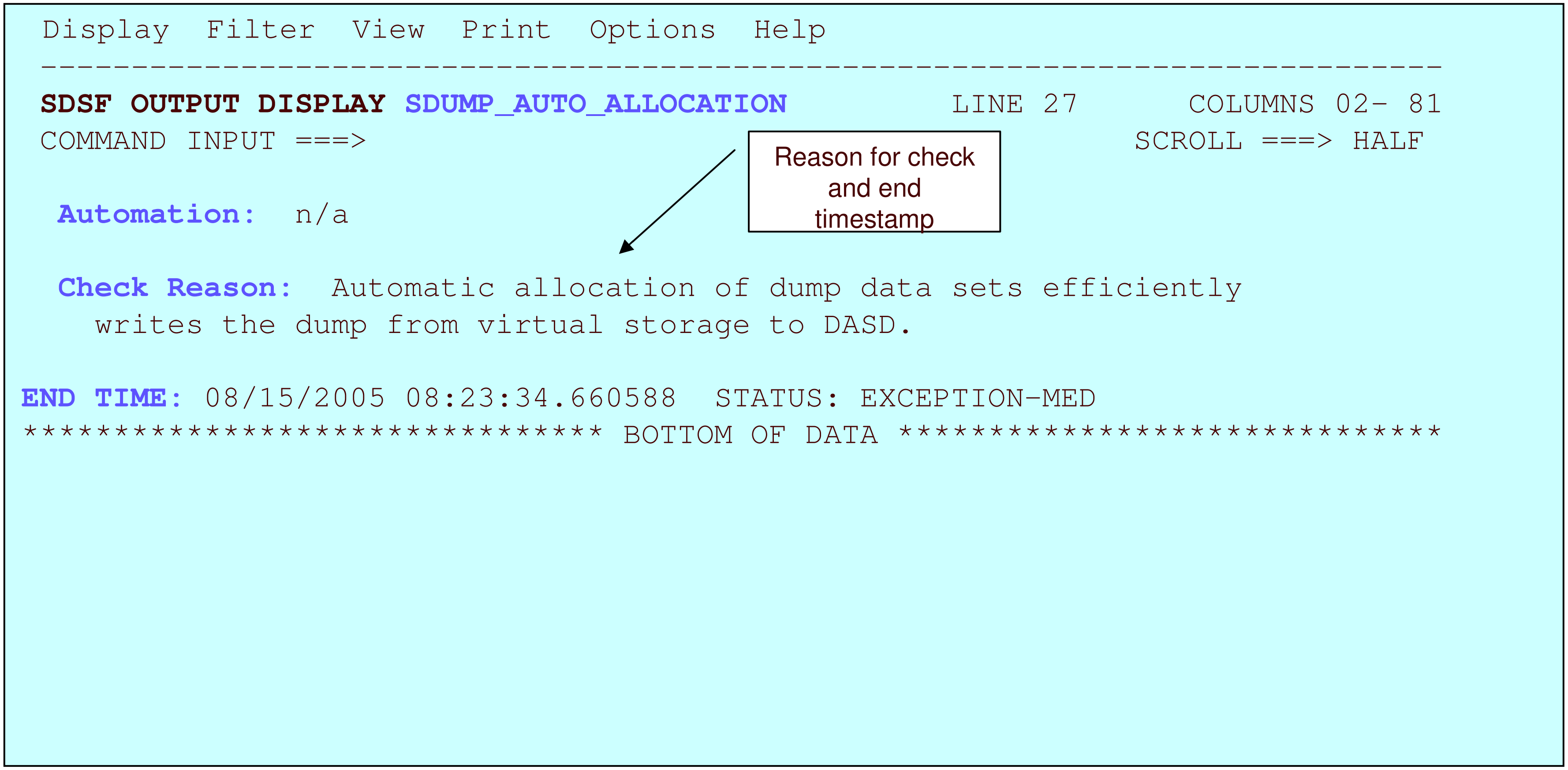
IBM Health Checker for z/OS – Managing the components of Health Checker -SDSF Browse of “Exception” check output (2 of 3 SDUMP)



```
 Display  Filter  View  Print  Options  Help
-----------------------------------------------------------------------------
SDSF OUTPUT DISPLAY SDUMP_AUTO_ALLOCATION        LINE 17      COLUMNS 02- 81
COMMAND INPUT ===>                                           SCROLL ===> HALF

 Operator Response:  Report this information to the system programmer.

 System Programmer Response:  IBM Suggestion:  See the "Using
   Automatically Allocated Dump Data Sets" section of z/OS MVS
   Diagnosis: Tools and Service Aids for setup information.  One
   advantage is the efficiencies of using a system determined
   blocksize to write the dump to DASD. Another benefit is the dump
   will always fit into the space allocated for the data set.

 Problem Determination:  n/a

 Source:  SDUMP (SCDMP)

 Reference Documentation:  z/OS MVS Diagnosis: Tools and Service Aids
```

End user response & pointer to reference document.

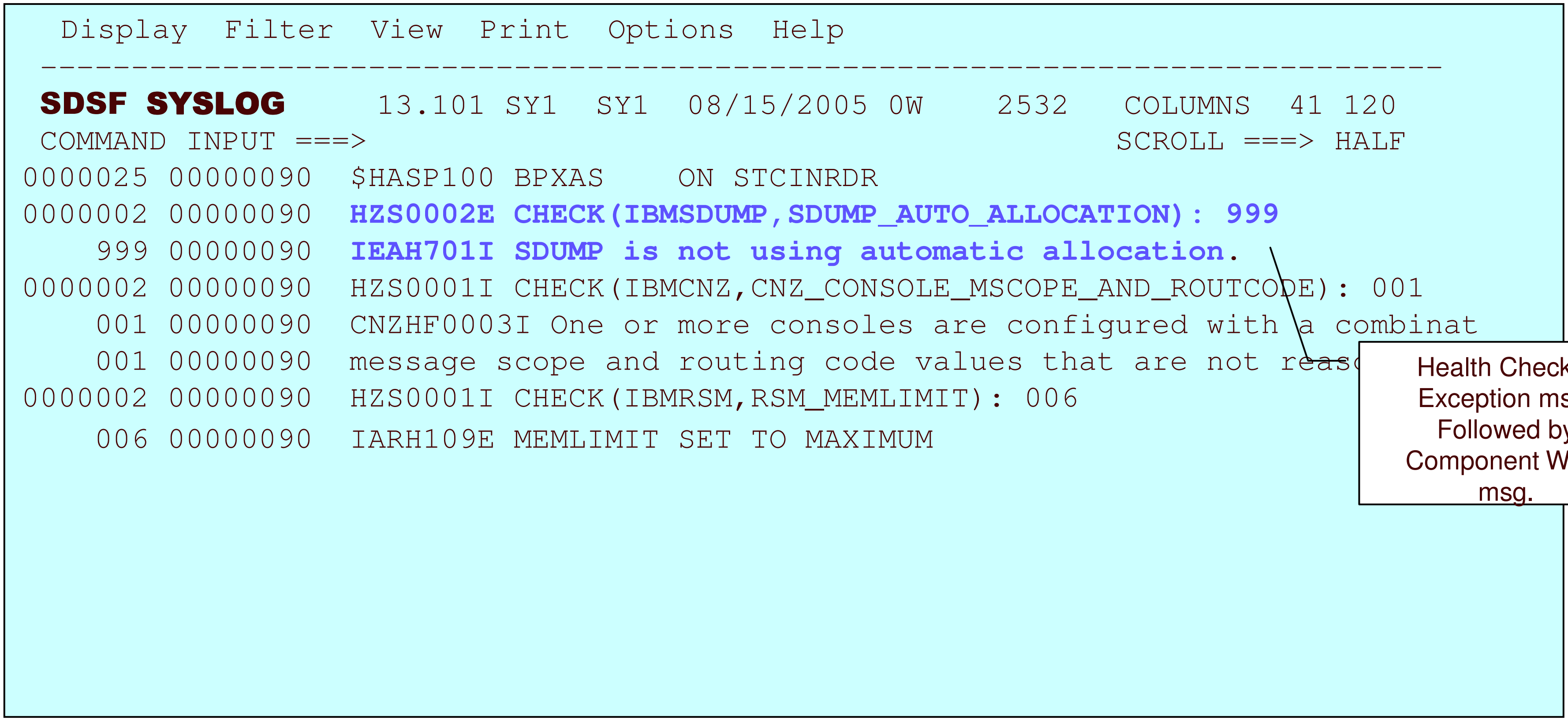
IBM Health Checker for z/OS – Managing the components of Health Checker -SDSF Browse of “Exception” check output (3 of 3 SDUMP)



```
 Display  Filter  View  Print  Options  Help
 ------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY SDUMP_AUTO_ALLOCATION         LINE 27      COLUMNS 02- 81
 COMMAND INPUT ===>                                             SCROLL ===> HALF

  Automation:   n/a

  Check Reason:  Automatic allocation of dump data sets efficiently
    writes the dump from virtual storage to DASD.

END TIME: 08/15/2005 08:23:34.660588  STATUS: EXCEPTION-MED
******************************** BOTTOM OF DATA ******************************
```

Reason for check and end timestamp

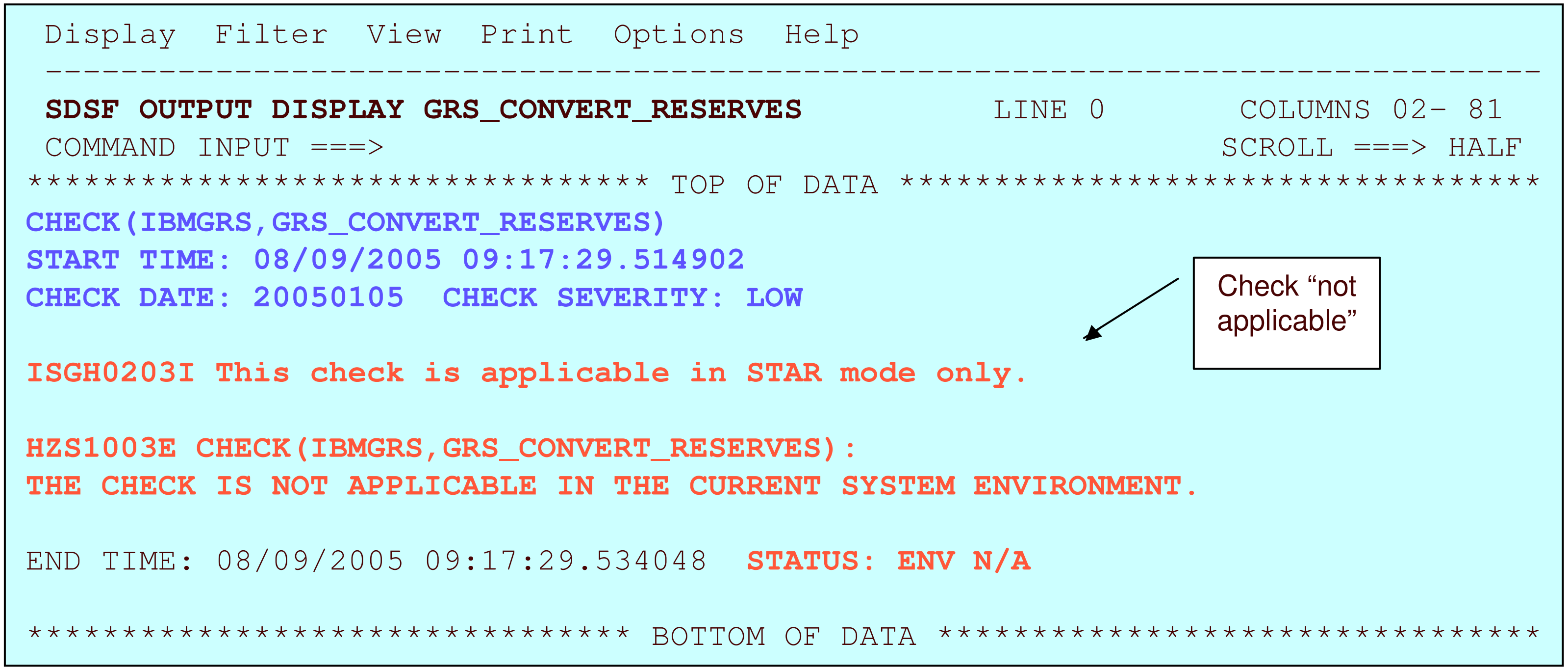
IBM Health Checker for z/OS – Managing the components of Health Checker -SDSF SYSLOG view of IEAH701I SDUMP Health Checker Exception



```
  Display  Filter  View  Print  Options  Help
 -----------------------------------------------------------------------------
 SDSF SYSLOG     13.101 SY1   SY1   08/15/2005 0W    2532    COLUMNS   41 120
 COMMAND INPUT ===>                                            SCROLL ===> HALF
0000025 00000090  $HASP100 BPXAS    ON STCINRDR
0000002 00000090  HZS0002E CHECK(IBMSDUMP,SDUMP_AUTO_ALLOCATION): 999
    999 00000090  IEAH701I SDUMP is not using automatic allocation.
0000002 00000090  HZS0001I CHECK(IBMCNZ,CNZ_CONSOLE_MSCOPE_AND_ROUTCODE): 001
    001 00000090  CNZHF0003I One or more consoles are configured with a combinat
    001 00000090  message scope and routing code values that are not reaso
0000002 00000090  HZS0001I CHECK(IBMRSM,RSM_MEMLIMIT): 006
    006 00000090  IARH109E MEMLIMIT SET TO MAXIMUM
```

Health Checker Exception msg. Followed by Component WTO msg.

IBM Health Checker for z/OS – Managing the components of Health Checker -SDSF
Browse of “System neutral (N/A) ” check output



```
  Display  Filter  View  Print  Options  Help
  -------------------------------------------------------------------------------
  SDSF OUTPUT DISPLAY GRS_CONVERT_RESERVES          LINE 0       COLUMNS 02- 81
  COMMAND INPUT ===>                                            SCROLL ===> HALF
********************************* TOP OF DATA **********************************
CHECK(IBMGRS,GRS_CONVERT_RESERVES)
START TIME: 08/09/2005 09:17:29.514902
CHECK DATE: 20050105  CHECK SEVERITY: LOW

ISGH0203I This check is applicable in STAR mode only.

HZS1003E CHECK(IBMGRS,GRS_CONVERT_RESERVES):
THE CHECK IS NOT APPLICABLE IN THE CURRENT SYSTEM ENVIRONMENT.

END TIME: 08/09/2005 09:17:29.534048  STATUS: ENV N/A

******************************** BOTTOM OF DATA ********************************
```

Check “not applicable”

Some considerations about exception messages

- 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
  - Decide what is appropriate for your system

- Exceptions are not an indicator that there is a problem to report to IBM
  - Exceptions are meant to tell you about potential availability impacts

- **No automatic correction of exceptions is done by IBM Health Checker for z/OS.**

Resolving Exceptions

- Resolving your exceptions is important, or they will be repeatedly reported (And probably get mixed up with ones you don't want to ignore!)
- You can resolve exceptions in the following ways:
  - Make the suggested updates to your system
    - This is the recommended approach
  - Review the system values that the check is using for applicability to your system environment
    - If the values aren't appropriate, tailor the values for your system to eliminate the exception when the check runs again
  - Make the check *Inactive* or delete the check
    - Check not applicable to your system environment

IBM Health Checker for z/OS - Persistent data across IPLs

- Some component check programs require historical data in order to perform the system check
  - VSM “VSM_CSA_CHANGE” is an example
    - **Reports on changes in CSA or Private storage since the last IPL**
- Requirement to run HZSPDATA allocation JCL sample to set up area to accumulate data to persist across IPLs

IBM Health Checker for z/OS –

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 -- Recommend using system name in the HZSPDATA name

Will be in HZSPROC

```
//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
```

Don’t change these attributes

IBM Health Checker for z/OS - Print Utility

- HZSPRINT utility provided to write check message buffers to a dataset/sysout
  - HZSPRINT utility can be used to print the current message buffers
  - HZSPRINT can also be used to print message buffers that have been archived to a log stream
  - Access to output of checks can be controlled through SAF (System Authorization Facility) profiles

IBM Health Checker for z/OS – HZSPRINT utility

- **Set up the HZSPRINT utility to view print output**
- HZSPRINT utility allows you to see check output in the message buffer
  - Can also 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 required

```
//HZSPRINT EXEC PGM=HZSPRNT,TIME=1440,REGION=0M,
//    PARM=('CHECK(check_owner,check_name)')
//*   PARM=('LOGSTREAM(logstreamname)','EXCEPTIONS',
//*   'SYSNAME(sysname)',
//*   'CHECK(owner,name)')
//SYSOUT   DD SYSOUT=A,DCB=(LRECL=256)
```

Defaults to *,*

Only show exception

Defaults to *

When using a data set, it must be seq, FB, LRECL 256

IBM Health Checker for z/OS - Logger support

- Archival of message buffers to Log Stream supported for retention of historical check output
  - Requires defining and connect to Log stream
  - Logger support can be turned on and off via operator command function and Parmlib

IBM Health Checker for z/OS –
Define log streams

Define log streams to keep historical records of check output, as needed

- Only the last iteration of a check is retained in the message buffer.
- It may be 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 log streams via MODIFY command or in HZSPRMxx parmlib member.
3. Log stream can be disabled via MODIFY command

IBM Health Checker for z/OS - Parmlib Support

- MVS Parmlib support available
  - Simplifies parameter specification for Health check programs
  - Sample member HZSPRM00 in SYS1.PARMLIB
    - Defines check parameter defaults, execution options, execution interval, etc.
    - Optionally include pervasive POLICY statement override definitions
    - Supports automatic commands
      - Enter commands to get executed at IBM Health Checker for z/OS start up
    - Wild carding supported for ease of parameter specification

IBM Health Checker for z/OS – Policy statement

- A Policy statement is used to override check defaults or to delete a check
  - Policy statements are applied whenever a check is added or refreshed, or whenever the policy changes
- Policy statements can be added, replaced or deleted via PARMLIB
  - Specify in HZSPRMxx PARMLIB member
    - Persistent across IPLs and A.S. restarts
    - Use Policy statements instead of Operator command for persistent changes across IPLs and A.S. restarts

IBM Health Checker for z/OS - Sample parmlib member HZSPRMxx with policy example

```
-----------------------------------------------------------------
 CHECK(IBMGRS,GRS_MODE)
-----------------------------------------------------------------
Check that GRS is in the suggested mode

PARAMETERS:
     1. Mode. Required. Must be STAR, RING or NONE.
-----------------------------------------------------------------

ADD POLICY STATEMENT(IBMGRS_DEFAULT01)
 UPDATE CHECK(IBMGRS,GRS_MODE)
 SEVERITY(LOW) INTERVAL(ONETIME) PARM('STAR') DATE(20050105)
 REASON('GRS should run in STAR mode to improve performance.')
```

Defaults shipped by IBM can be overridden

IBM Health Checker for z/OS - Getting Started

✓ **Run Setup samples and utilities to allocate needed data sets, etc.**

✓Define STC related user id requirements

- ✓MVS / USS user id implications

✓Security product definitions

IBM Health Checker for z/OS - Getting Started (continued)

✓ Starting the Health Checker address space

- Start as a started task

- Sample proclib member HZSPROC

```
//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
```

► Start via MVS operator command: 'S HZSPROC'

IBM Health Checker for z/OS – Getting Started

- Component health check programs with established EXIT routines via DYNEXIT are now available to run/report
- Other Component health check programs can be added without shutting down Health Checker A.S.
- IBM Health Checker for z/OS task and component health check programs can be manipulated with command or program interfaces
- IBM Health Checker for z/OS Parmlib can be used for persistent product customization

IBM Health Checker for z/OS – Managing the components of Health Checker: Assembler Programmable Interfaces

- **HZSCHECK**

  Dynamically delete or run specified check(s)

- **HZSQUERY**

  Obtain check status information and messages

- **HZSADDCK**

  Add health check program to IBM Health Checker for z/OS

- **HZSFMSG**

  Format and issue check message

IBM Health Checker for z/OS – User written health check programs

Component Check program requirements

- Checks run in Health Checker address space
- Must be fetched from an APF authorized library
- Checks are "added to" Health Checker

Adding a Health check program to IBM Health Checker for z/OS

- Checks are added via the **HZSADDCHECK dynamic exit.**
  - Exit routines for **HZSADDCHECK dynamic exit**
  - Most components add the HZSADDCHECK dynamic exit at component initialization through the **CSVDYNEX** program interface
  - Alternative: **SETPROG** operator command.

User witten Health check programs: HZSADDCHECK exit routine

- **The HZSADDCHECK exit routine is used to add one or more checks to Health Checker via HZSADDCK macro**
  - Define Unique Check(owner,name)
  - Define check routine and message tables
  - Define check scope
  - Define check defaults
  - Date & Reason

User written Health check programs: Health check program defaults

- Check severity (High/Medium/Low)
- Interval
- Check parameters (if applicable)
- Active/Inactive
- Check defaults can be overridden

User written Health check programs: Health check program message table

- Each check has a corresponding message table
- Messages are written in SGML based language
- ‘Standard’ message structure required
- All messages should be self contained
  - Contain explanations and recommended actions
  - Avoid required additional references
- Message table assembler CSECT produced by an Exec from the message source

User written Health check programs: Health check program details

- Checks run in the Health Checker address space
- Checks are given a 2K persistent area to save data between check iterations
- Checks are given a 4K dynamic area
- Input area provided that contains the current state of the check (including installation overrides)
- Supervisor state, Key 8
- Checks issue messages via HZSFMSG macro

User written Health check programs: Health Check program phases

- **Initialization**
  - •When check is added/activated
- **Check**
  - -When check is added/activated
  - -As per check interval/run request/parameter update
- **Cleanup**
  - •Immediately after each check call
- **Delete**
  - •When check is deleted/deactivated

IBM Health Checker for z/OS - Summary

- IBM Health Checker for z/OS integrated into z/OS R7
- Available as a web deliverable back to z/OS R4
- Setting up IBM Health Checker for z/OS
  1. Get right software installed
  2. Allocate HZSPDATA for saving check data
  3. Define log streams
  4. Perform security definitions
  5. SDSF customization
  6. Create HZSPRMxx parmlib member, and HZSPROC started task
  7. Start IBM Health Checker for z/OS
  8. Utilize virtual PSP tool to look for new checks that may have been delivered as PTFs
- Know what check exceptions are, and what to do about them
- Create your own checks

IBM Health Checker for z/OS – Comparison

Comparison to z/OS prototype health check tool

Function	z/OS prototype health check tool	IBM Health Checker for z/OS
Invocation	Batch job	Started Task address space
Check programs	Included in base	Integrated with component(s)/PTFs
Check Management & User Overrides	USERPARM Members	•SDSF •Modify cmd •Programmable interfaces •HZSPRMxx parmlib (includes symbolic and policy stmt. support
User defined groups of checks (categories)	Not supported	Supported
Output viewing	SYSOUT dsn	•Display cmd •WTOs, messages reporting check results •Log stream •SDSF •HZSPRINT utility

IBM Health Checker for z/OS – Technical Q & A

Q & A