



CICS Health Check

- ◆ Eugene S Hudders
- ◆ C\TREK Corporation
- ◆ ehudders@ctrek.com
- ◆ (787) 397-4150



Disclaimers

- ◆ Remember YMMV
- ◆ Remember the political factor
 - ◆ “If it ain’t broke, don’t fix it!”
- ◆ The following products are trademarks of IBM Corporation, Armonk, NY
 - ◆ IBM
 - ◆ CICS Transaction Server (CICS TS)
 - ◆ COBOL LE – Enterprise COBOL
 - ◆ MVS
 - ◆ z/OS
 - ◆ z/OS Health Checker
 - ◆ VSAM
 - ◆ DB2



Agenda

- ◆ Health Check Definition
- ◆ Introduction
- ◆ Top 6 Benefits of a Periodic Health Check
- ◆ Sources of Information
- ◆ Traditional Health Checks
 - ◆ Examples
 - ◆ Attention/Warning Messages
- ◆ Major Considerations
- ◆ Health Check
 - ◆ Message Classifications
 - ◆ Health Check Categories
 - ◆ Sample of Non-Traditional Health Checks
 - ◆ Actions
- ◆ IBM Health Checker
- ◆ Closing

Health Check Definition

- ◆ What is a Health Check?
 - ◆ An examination to determine whether a person or animal is suffering from illness or injury (Noun) (Oxford Dictionary)
- ◆ For CICS TS
 - ◆ An examination of the entire address space to determine how well your CICS TS system is executing and if it is efficiently using the resources assigned and provides sub-second response times
 - ◆ CPU
 - ◆ Virtual/Real Storage
 - ◆ I/O
- ◆ An overall assessment that accurately determines how your CICS TS system is performing



Introduction

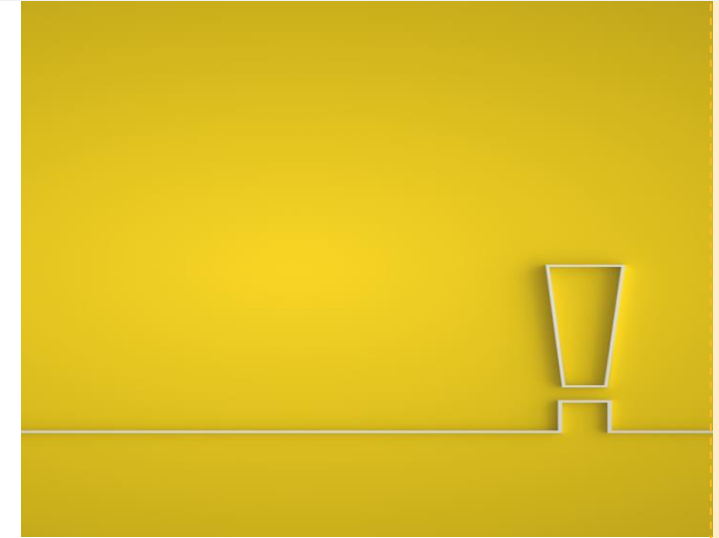
- ♦ Most installations today rely on some type of software products to help them identify potential situations that could affect the performance of their CICS system
 - ♦ A dependence mainly on performance monitors
- ♦ Major issue is the number of CICS systems that need to be monitored and the number of system programmers needed to accurately assess and maintain the CICS systems
 - ♦ In many installations performance tuning and problem determination are done when a problem occurs
 - ♦ Availability of experienced system programmers
 - ♦ Educational expenditures are limited if at all available



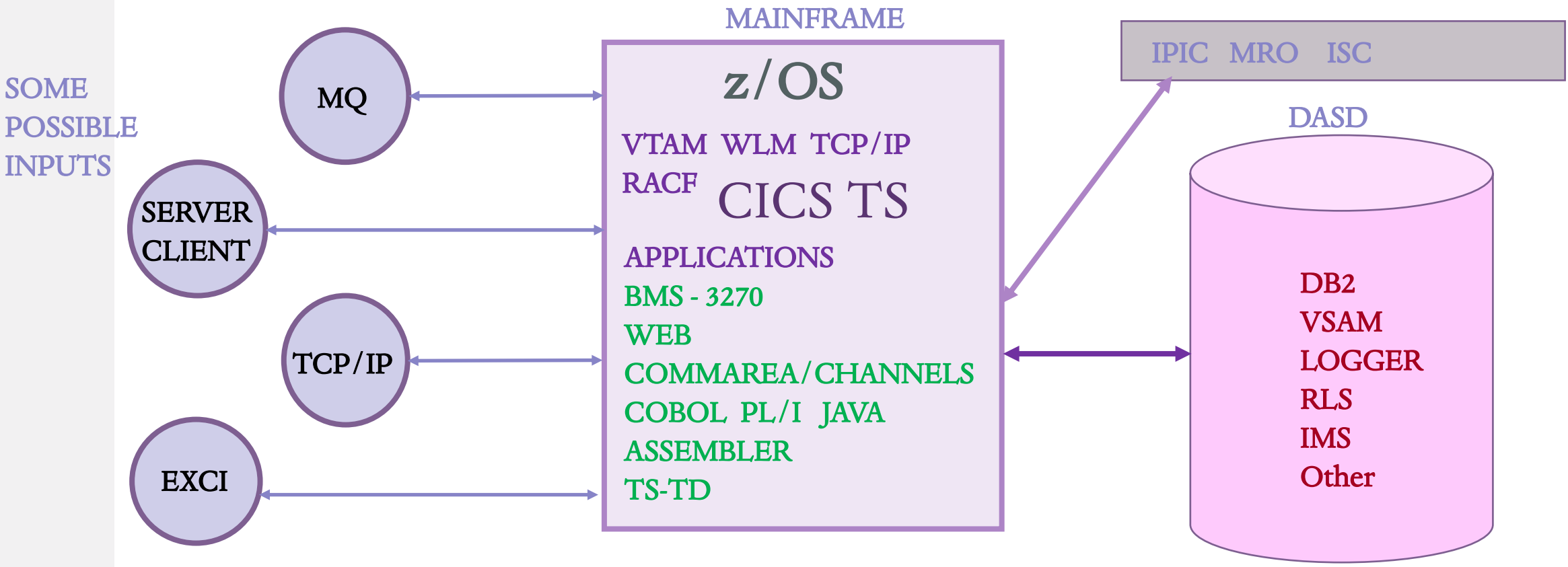
Health Check Advantages

- ◆ Top 6 Benefits of a Periodic Health Check

- ◆ **OPERATIONAL COST:** Keeps expenses and outages at a minimum
- ◆ **FUTURE COST:** Potential to defer system upgrades
- ◆ **PREVENTION:** Correct anomalies before they cause outages
- ◆ **STABILITY:** Maintain CICS overall performance at peak performance
- ◆ **STRESS INDICATORS:** Spot abnormalities that directly impact Transactions Per Second (TPS) response like CPU, storage and I/O
- ◆ **CONTROL:** Gives the System Programmer(s) the ability to see a problem before it becomes fatal and tells him/her the root cause of the problem



General CICS TS Region



CICS HEALTH CHECK HAS TO LOOK AT THE BIG PICTURE

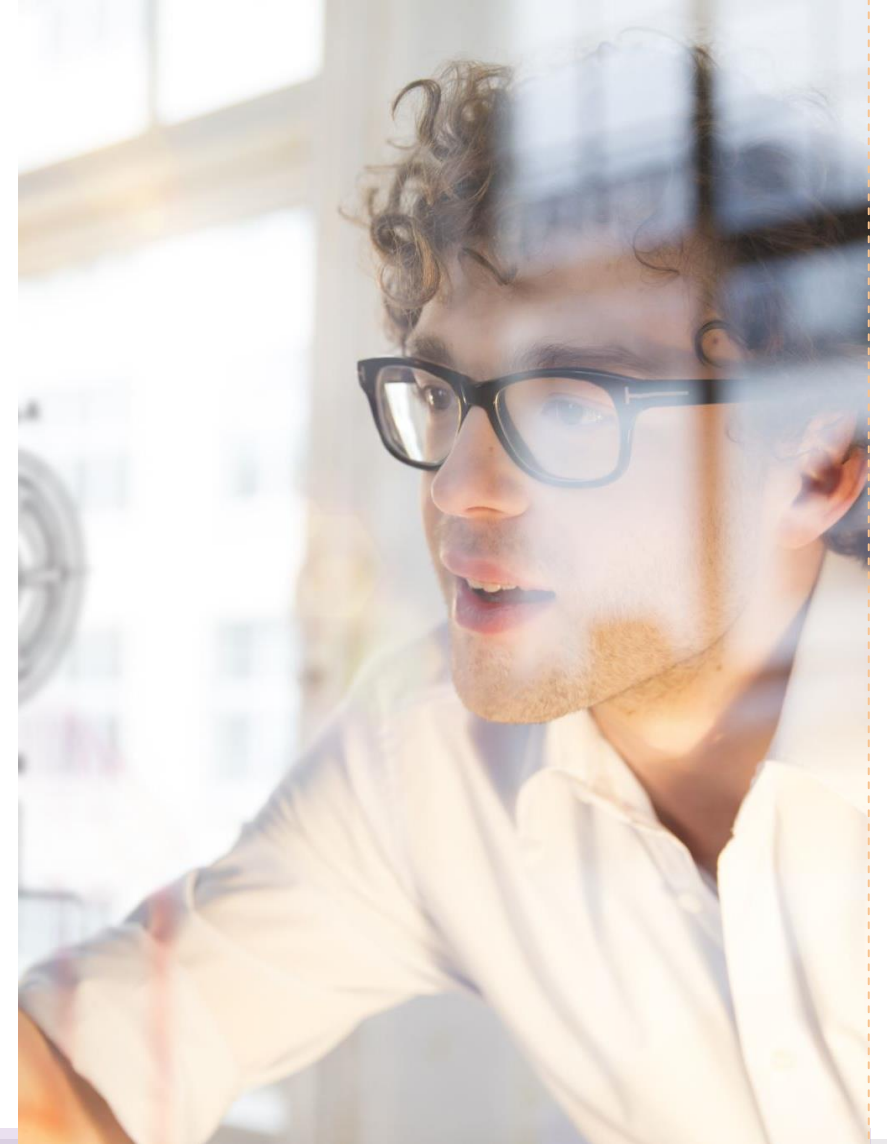


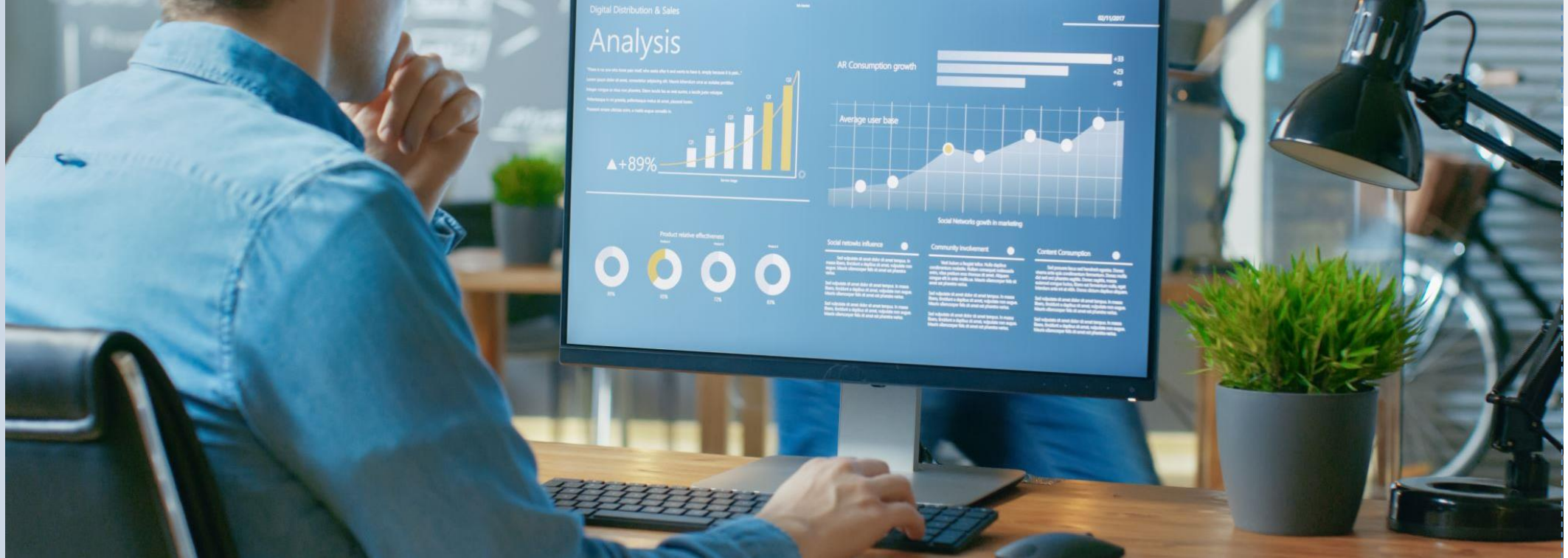
Some Sources of Information

- ◆ SMF
- ◆ RMF
- ◆ Third Party Monitors
- ◆ Internal CICS Transactions (e.g., CEMT, etc.)
- ◆ STAT Transaction
- ◆ z/OS Health Checker
- ◆ EOD Statistics
- ◆ Third Party Tools
- ◆ Dumps (Real Headache!)

Traditional CICS Health Checks

- ◆ **Most common areas monitored:**
 - ◆ Overall CPU and CICS CPU Utilization
 - ◆ Long Running Tasks (FISH tasks)
 - ◆ Heavy CPU Consuming Tasks
 - ◆ CPU to Dispatch Ratio
 - ◆ MXT and TCLASS Limits
 - ◆ DSA/EDSA/GDSA Usage (SOS, Cushion, etc.)
 - ◆ Storage Violations
 - ◆ Logger
 - ◆ Overall Paging and CICS Paging
 - ◆ LSR Pool Look-Aside Percent (Data and Index)
 - ◆ TCB Usage (e.g., QR, L8/L9, MAXOPENTCBS etc.) (DB2, MQ, etc.)
 - ◆ Waits (String, Buffer, Enqueue, etc.)
 - ◆ Temporary Storage and Transient Data
 - ◆ Communications Session Usage
 - ◆ SIT Parameter Settings (ICV, ICVR, ICVTSD, etc.)





Traditional CICS Health Checks

- ◆ In general, less than 50 conditions are monitored and produce messages
- ◆ Some require that you research the information

Some Examples of Traditional Messages

- ◆ Wait on Strings
 - ◆ Two types of common messages
 - ◆ Wait on Strings at the LSR pool level
 - ◆ Wait on strings at the file level (NSR/LSR)
 - ◆ Other types
 - ◆ Temporary Storage (TS) DFHTEMP
 - ◆ Transient Data (TD) DFHINTRA



Some Examples of Traditional Messages

- ◆ Storage
 - ◆ Three types of common messages (G/E) DSA
 - ◆ % of DSA usage
 - ◆ % of EDSA usage
 - ◆ % of GD SA usage
 - ◆ Approaching SOS or at SOS
 - ◆ SOS in OSCOR (MVS) type storage (New CICSTS56)

Some Examples of Traditional Messages

- ◆ Maximum Tasks (MXT)
 - ◆ % MXT
 - ◆ Approaching MXT
- ◆ Transaction Class (TCLASS)
 - ◆ % Maximum Task Limit for a Class
 - ◆ Approaching Maximum Task Limit for a Class
- ◆ Files
 - ◆ CI/CA Split Activity
 - ◆ ENQ Waits (e.g., Exclusive Control Waits)
- ◆ TS/TD CIs allocated and used



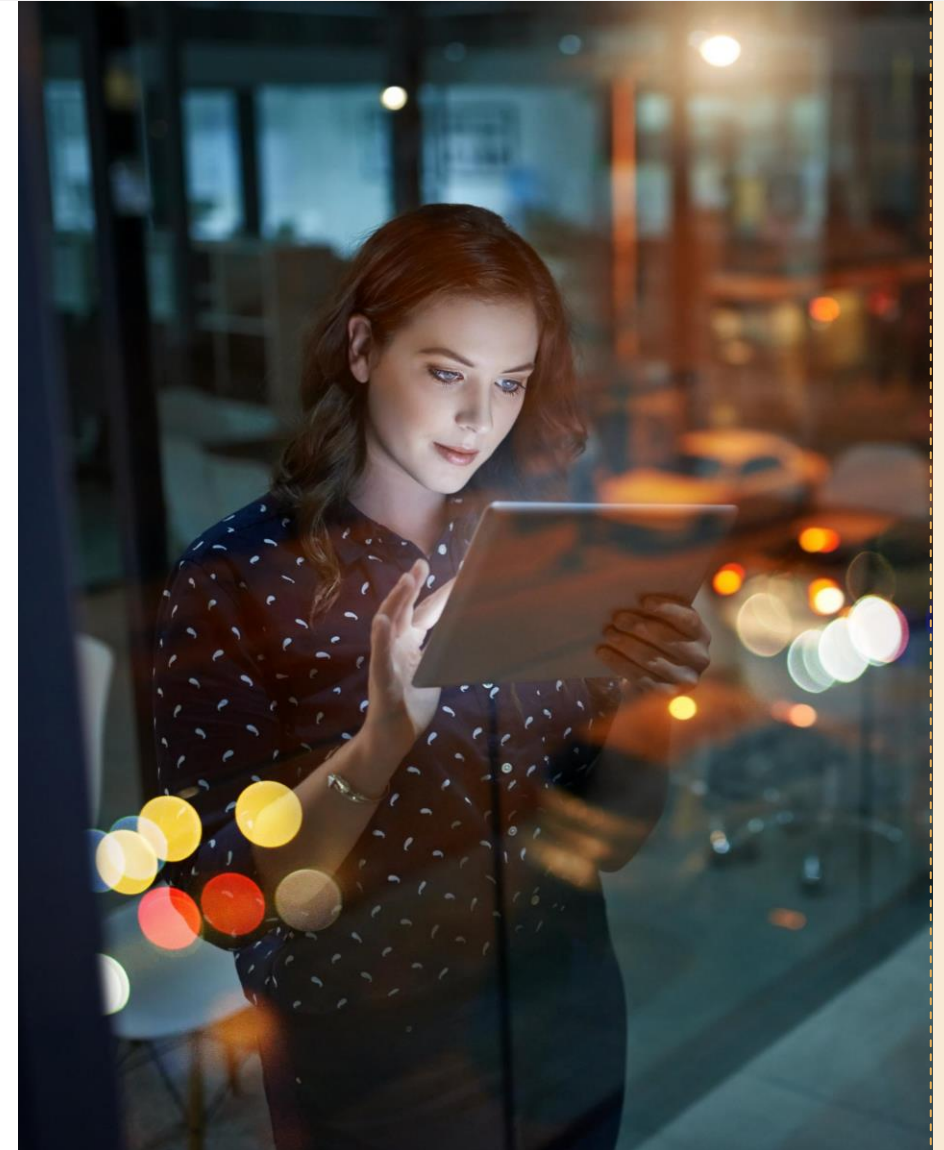


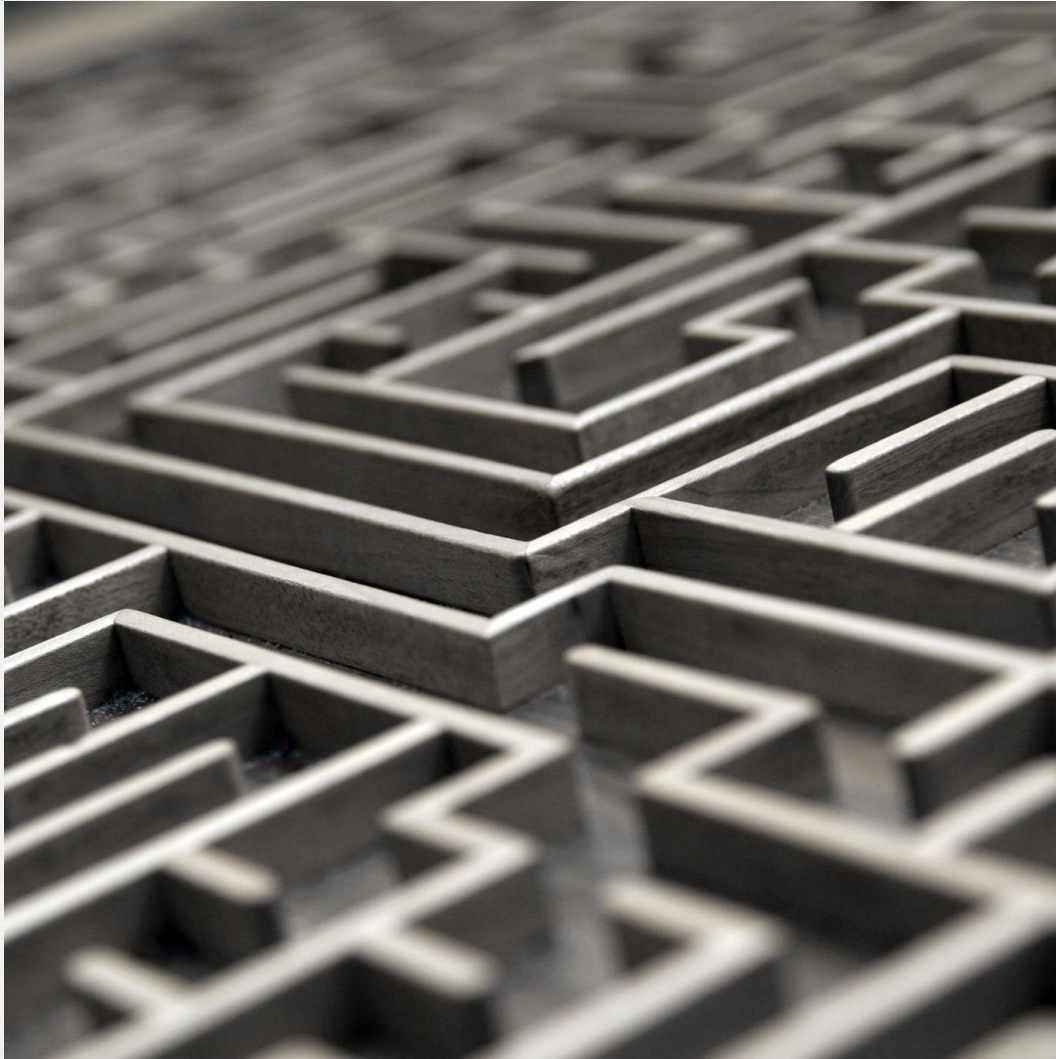
Attention/Warning Messages

- ♦ Some installations have added warning alerts to be informed when exception conditions occur by issuing a message
 - ♦ Reaching MXT or a TCLASS Limit
 - ♦ Reaching SOS
 - ♦ Excessive CPU usage by a task
 - ♦ Excessive task residency time
 - ♦ Excessive storage being used by a task
 - ♦ Excessive task I/O
 - ♦ Excessive CPU utilization by the region

Major Considerations

- ♦ Most monitors will present the information regarding the issues mentioned on the previous slide.
 - ♦ The issues presented are an important part of identifying how efficient the resources are being used by CICS.
 - ♦ They provide good information regarding exception conditions such as SOS, wait on resources such as strings, MXT etc.





Major Considerations

- ◆ However, there are many things degrade resources that are hidden.
 - ◆ For example, you get an SOS condition below the line
 - ◆ You may see many tasks running with storage allocated below the line
 - ◆ Do you raise the DSALIM (if storage is available) or find out why there are so many tasks are using storage below the line?
 - ◆ Upon investigation you may find that programs are running above the line but hidden to you are programs compiled with a COBOL CBL that had DATA(24) specified

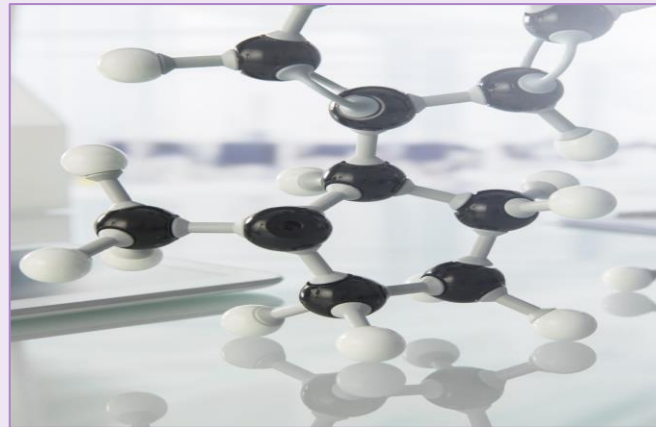
CICS Health Check

So, the idea of a CICS Health Checker should include not only those things that are occurring in the system but also those hidden things that can potentially cause a situation if the proper conditions occur

The Health Checker will also help in problem determination and performance tuning

The Health Checker can also be used as a proactive tool to identify possible potential problems and areas that need attention

The Health Checker should continually monitor the CICS system for potential problem and performance issues that may exist in the system



Possibly, even providing a grading system that can be used to identify which CICS systems require immediate attention



CICS Health Check

- ◆ In addition to the traditional messages, there are over 300 other things that can affect the overall health of the CICS system.
- ◆ The areas include not only resources that are reaching limits such as MXT, but also poor use of CICS resources that could be allocated for other purposes (e.g., more LSR buffers to reduce physical I/O).

- ◆ These areas include but not limited to
 - ◆ Over allocated resources such as MXT or TCLASS definitions
 - ◆ Over allocated VSAM resources such as DASD space
 - ◆ Under allocated VSAM resources such as free space
 - ◆ Incorrect VSAM CISZ for data and index
 - ◆ Unused resources such as disabled or never used resources
 - ◆ Potentially dangerous settings such as SHUTDOWN in transaction/system dumps
 - ◆ Low thread REUSE in DB2
 - ◆ Low look-aside hit ratio in LSR

CICS Health Check





Classification of Messages

- ♦ There are several Health Check classification of messages that can be produced as you analyze your CICS system
 - ♦ System Informational Messages – for example, represent information regarding the overall status of the system such as CPU utilization or paging
 - ♦ CICS Informational Messages – for example, represent incorrect settings of parameter definitions such as some SIT parameters that could be specified in a better way such as a high ICVR
 - ♦ Warning Messages – for example, that may indicate a potential situation that may need correction such as having too many index levels in a KSDS file
 - ♦ Potentially Severe Messages – for example, that may indicate a potential problem such as SOS or MXT or a poor LSR look-aside hit ratio that increases physical I/O activity and adversely affects task response times

CICS Health Check Categories



- ♦ These are some of the major categories that should be addressed with a CICS Health Check:
 - ♦ CICS Related (SIT or Resource Definitions)
 - ♦ Enqueue
 - ♦ VSAM Data Sets
 - ♦ LSR Pool
 - ♦ Data Tables
 - ♦ DB2 Definitions
 - ♦ MQ
 - ♦ JAVA
 - ♦ IMS
 - ♦ Logger
 - ♦ Storage Management (Real/Virtual)
 - ♦ Operating System
 - ♦ Communications
 - ♦ Security
 - ♦ System Informational
- ♦ There are over 300 different checks over the traditional ones.



CICS Related

- ◆ These are some SIT parameters that can help in more efficient use of resources
 - ◆ ICV/ICVR/ICVTSD – in certain circumstances can help reduce CPU utilization or ineffective settings
 - ◆ ICVR = 0
 - ◆ SUBTSKS=1 on a uniprocessor
 - ◆ BMS use of Device Dependent Suffix (DDS)
 - ◆ Non-use of LE options
 - ◆ Internal/ Auxiliary Trace
- ◆ Dump requests (SYD/TRD)
 - ◆ SHUTDOWN Option specified
- ◆ TCTUA/TWA Usage that can be a source for storage violations or non-use which wastes storage
- ◆ Unnecessary use of LOAD HOLD
- ◆ Unused Resource Definitions (TXD, PPT, FCT etc.)

VSAM Files/LSR Related



- ◆ LSR
 - ◆ Use of dynamic allocations for pools
 - ◆ Poor look-aside on a buffer size
 - ◆ Over allocation of strings and key length
 - ◆ Buffer fragmentation
 - ◆ No Index Pool defined
- ◆ VSAM
 - ◆ Poor free space allocation
 - ◆ None provided with files reflecting inserts
 - ◆ Provided and files do not reflect any inserts
 - ◆ Incorrect space allocation
 - ◆ Too small can lead to unnecessary extent processing
 - ◆ Too large can lead to wasted, unused disk space
 - ◆ Excessive CI/CA splits

z/OS and System Informational

- ◆ Although the z/OS (MVS) area is not a direct responsibility of the CICS system programmer, there are areas that you should review because they could influence or impact the CICS region
 - ◆ (E) CSA free space availability
 - ◆ Amount of MVS region storage available in the region
 - ◆ Available Frame Queue (AFQ)
 - ◆ Unreferenced Interval Count (UIC)
- ◆ System Informational
 - ◆ Transactions per second versus estimated capacity for the CICS region
 - ◆ CICS region priority versus other work (WLM)



CICS Health Check



- ♦ To get a proper perspective about the health of your CICS system, you need to analyze not only the traditional areas but also those areas not generally covered by system monitors
- ♦ Develop the different checks that apply to your installation
- ♦ Separate the conditions found by category
- ♦ Too many potentially severe messages may be an indication that the system is severely underperforming
- ♦ Some of these situations are another department's responsibility to fix (e.g., VSAM file definitions)
- ♦ Determine how often you should run the Health Checker
 - ♦ Some conditions don't change
 - ♦ Others can appear suddenly



Introduction to the IBM Health Checker

- ♦ What is the z/OS Health Checker?
 - ♦ It is a software package that gathers information about the system environment and parameters to help identify potential configuration problems that could affect the availability and reliability of the system and may possibly result in system outages.
 - ♦ It is a z/OS component that helps simplify and automate the identification of potential configuration problems before they impact system availability or cause outages.
 - ♦ CICS Transaction Server supports Health Checker rules that define the best practices for CICS system configuration.

Source: IBM Health Checker for z/OS

Introduction to the IBM Health Checker

- ◆ Each CICS TS region runs a system transaction CHCK every 30 minutes to check and report on compliance to the best practices
- ◆ Every 30 minutes the IBM Health Checker for z/OS provides address space reports on the previous 30-minutes activity
 - ◆ If any region is non-compliant with the best practices, a warning is issued with the details of the non-compliance
- ◆ There are three CICS health checks supplied for the Health Checker
 - ◆ See PI76965
 - ◆ See PI76963
 - ◆ Operating system PTFs for previous z/OS releases
 - ◆ For z/OS V2.1 → UA91584
 - ◆ For z/OS V2.2 → UA91583

Source: IBM Health Checker for z/OS



z/OS Health Checker



- ♦ The z/OS Health Checker has a minimum check for CICS health issues
- ♦ The z/OS Health Checker rules run every 30 minutes in the z/OS Health Checker address space
 - ♦ Transaction – CHCK (Priority 254)
 - ♦ Program – DFHHCHK
- ♦ Any conditions result in a warning message being issued
- ♦ When the system comes up and WLMHEALTH is active, the system will register with the z/OS Health Checker and send message DFHHC0101I and system transaction CHCK is activated

Health Checker Registered Message

• **+DFHHC0101I CICSTS56 CICS has registered successfully to the z/OS Health Checker.**

• D T

• IEE136I LOCAL: TIME=17.57.55 DATE=2020.180 UTC: TIME=21.57.55 DATE=2020.180

• D A,L

• CNZ4105I 17.57.55 DISPLAY ACTIVITY 489

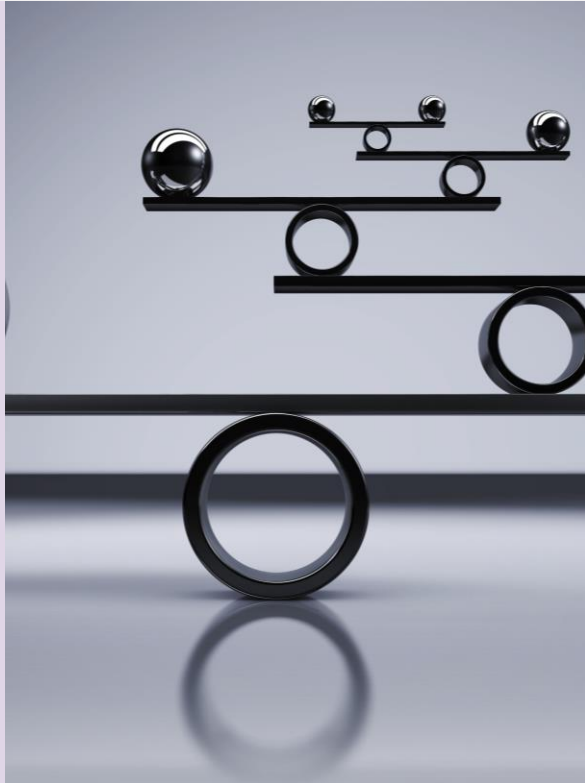
(Console Messages)

JOBS	M/S	TS	USERS	SYSAS	INITS	ACTIVE/MAX	VTAM	OAS
00007	00027	00001	00036	00016	00001/00100	00026		
LLA	LLA	LLA	NSW S	HZR	HZR	IEFPROC	NSW S	

- 33 2A CHCK QR 36 DFHHCHK SUSP SUSP NOT RUNNING SY NO (System Task)

1CAD4710 1CAD4030 CHCK TASKWAIT 05 08 07072020 11:07:01 11:37:01 DF0466AA (ICE)

CICS Workload Manager Health API



- ♦ A recent SIT parameter (WLMHEALTH) can be used to control the workflow into the CICS TS region
 - ♦ Can be used to have warm-up process after CICS system initialization to ensure CICS is ready to receive work
 - ♦ For example, MQ Adapter
 - ♦ CICSplex SM WLM
 - ♦ TCP/IP Sysplex Distributer and Port Sharing (works with TCP/IP Listeners, CICS Web Support, EZA Sockets, Liberty, JVMSERVERS,...)

CICS Workload Manager Health API

- ◆ **Format SIT Parameter**
 - ◆ `WLMHEALTH {(20 | interval[,25 | number]) | OFF}`
 - ◆ **interval** – Specifies the amount of time, in seconds, between calls that CICS makes to the z/OS Workload Manager Health API (IWM4HLTH). The value specified must be in the range 0 - 600. The default value is 20
 - ◆ **number** – Specifies the health adjustment value that CICS provides to the z/OS Workload Manager Health API (IWM4HLTH) in each call at the specified interval. The value must be in the range 1 - 100. The default value is 25
- ◆ **Takes effect after message: DFHSI1517 applid Control is being given to CICS**



z/OS Health Checker

- ◆ The z/OS Health Checker reviews potential security issues in CICS
 - ◆ CICS_CEDA_ACCESS – checks whether transaction CEDA is accessible to unauthorized users that may compromise the LPAR and/or CICS region security
 - ◆ SIT parameter SEC=NO
 - ◆ CEDA accessible to the CICS default user
 - ◆ CICS_JOBSUB_SPOOL – checks if the system spool is accessible to unauthorized users that may compromise the security of the LPAR and/or CICS region security
 - ◆ SIT parameter SPOOL=YES
 - ◆ SIT parameter SEC=NO
 - ◆ CECI accessible to the CICS default user





z/OS Health Checker

- ◆ **CICS_JOB SUB_TDQINTRDR** – checks to see whether an unauthorized user has access to the internal reader via a TD queue that may compromise the security of the LPAR and/or CICS region security
 - ◆ SIT parameter SEC=NO
 - ◆ CECI accessible to the CICS default user
 - ◆ Any TD data queue (EXTRAPARTITION with DD name that has SYSOUT referencing INTRDR) writes to an internal reader by the CICS default user

z/OS Health Checker

- ◆ One additional test is available regarding VSAM RLS files
 - ◆ VSAMRLS_QUIESCE_STATUS
 - The check looks for unresponsive CICS® regions for QUIESCE and UNQUIESCE, which could indicate a problem. IBM recommends monitoring QUIESCE and UNQUIESCE events for unresponsive CICS regions
 - ◆ Check message IGWRH0402E



z/OS Health Checker

- ◆ Possible messages:
 - ◆ DFHH0001E – CEDA transaction is available to unauthorized users
 - ◆ DFHH0002E – Spool is accessible to unauthorized user
 - ◆ DFHH0003E – A TDQ defined to internal reader is accessible to an unauthorized user
 - ◆ DFHH0200E – CICS Health Checker cannot run due to error
 - ◆ DFHH0301I – All CEDA transactions are protected from unauthorized users
 - ◆ DFHH0302I – The spool is protected from unauthorized users
 - ◆ DFHH0303I – No TDQs defined to internal reader that can be written to by unauthorized users



- ◆ Installations that have expanded the coverage of traditional checks to include additional checks have seen improvements
 - ◆ A reduction in the CICS CPU utilization
 - ◆ Improvement in the overall CPU utilization
 - ◆ Recovery of wasted disk space
 - ◆ Reduced physical I/O
 - ◆ Recovery of virtual/real storage that can be used for I/O buffers
 - ◆ Improved response times
- ◆ As a result, some installations have been able to defer upgrades
- ◆ Need to identify which areas you would like to analyze and develop the codes/actions to check for the conditions identified

