CICS and the z/OS Health Checker (Checking the health of CICS)

Darren Beard
CICS TS Developer
IBM UK Labs, Hursley Park, UK

darren_beard@uk.ibm.com

Agenda

- What is the Health Checker
- How it is used by CICS
- Basic architecture
- A look at how it works
- Changes in CICS TS 6.1

What is health checker?

- IBM Health Checker for z/OS is a tool to help identify potential configuration problems before they impact availability or cause system outages
  - It programmatically checks the current active z/OS and sysplex settings and definitions for a system
  - It generates output with detailed messages to inform of any potential problems and suggested actions to take to resolve them.

- CICS is utilizing it as part of the simplification and conformance strategy

Health Checker

- The Health Checker for z/OS consists of two main parts
  - The framework
    - Manages registration and scheduling of the checks, deals with logging and reporting messages
  - Checks
    - Programs or routines which check or evaluate product specific parameters such as settings and definitions

Background – IBM Health Checker for z/OS

- IBM Health Checker for z/OS designed to encourage best practice
  - Report where not conforming with advice
  - Available since z/OS 1.4 via PTF (2002)
  - Part of base product since z/OS 1.7, but turned off (Sep 2005)
    - Turned on by default from z/OS 2.1 (Sep 2013)

- Health Check output
  - Visible as option CK in SDSF
  - Alternatively, use the HZSPRINT exec in a batch job to get output
  - Checks are associated with a product or subsystem
    - IBM provides over 250 health checker checks
  - Each check tests configuration or state information
    - Result in SUCCESS, WARNING or EXCEPTION message

Background – IBM Health Checker for z/OS health checks

- Can be run locally in the Health Checker address space
  - easier to write the tests, but they must be written in assembler
  - may have limited access to data from the remote address space
- Can run remotely in the address space of the subsystem
  - easier to obtain data concerning the caller
- Checks can be
  - Active or inactive
  - Run once only, on a defined interval, or on demand

CICS Health Checks

- Health Checker has two types of checks
  - Local (to Health Checker Address space)
  - Remote (Runs in any address space)
  - Checks usually defined as part of z/OS install
- Why are the CICS Health Checks not standard local or remote checks ?
  - Cannot be run as normal local checks
    - Health Checker does not have access to CICS configuration/state
  - Cannot be run as normal remote checks
    - Each region would have its own check
    - Would require one message per region
- Why are CICS Definitions not part of the z/OS install?
  - Can’t be part of the z/OS set (CICS SMP/E is separate)
  - Can’t be user defined since purpose is to check configuration

Supplied Checks for CICS TS 5.4

- Aimed at stopping job submission with region userid by unauthorised users.

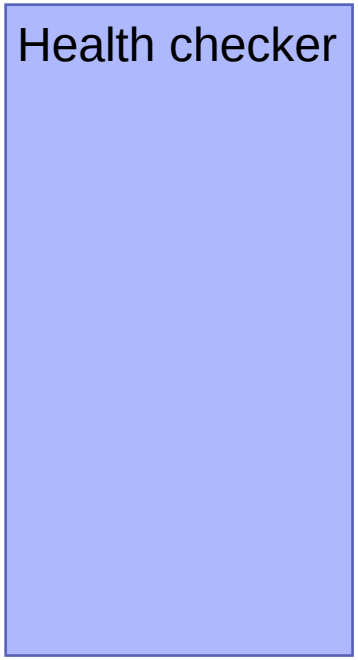
- CICS Checks Supplied
  - CICS_CEDA_ACCESS
    - CEDA available to default user
  - CICS_JOBSUB_SPOOL
    - SPOOL open
    - CECI available to default user
  - CICS_JOBSUB_TDQINTRDR
    - TD Queue to internal reader available to default user
    - CECI available to default user

Note: CICS support for the IBM Health Checker for z/OS is also available for releases prior to CICS TS 5.4. See APAR PI76963 for CICS TS 4.2 and APAR PI76965 for CICS TS V5 releases.

Timing of the CICS Health Checks

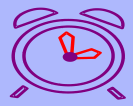
- CICS Health Checker tests are defined to z/OS by the first CICS region to start following an IPL
  - Defined as Active, Local and to be checked every 30 minutes
- Each CICS region has system task which runs health checks every 30 minutes.
  - Initially run 3-4 minutes after initialisation
  - Results of each check written to a buffer
- When Health Checker runs
  - Checks all regions which have run in the last 30 minutes
  - Reports on any regions which fail checks
  - Issues warning message if any fail or success message if none fail

Health Checker address space starts



First CICS after IPL starts

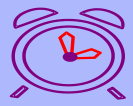
Health checker



CICS 1

Later, CICS region 9 starts

Health checker



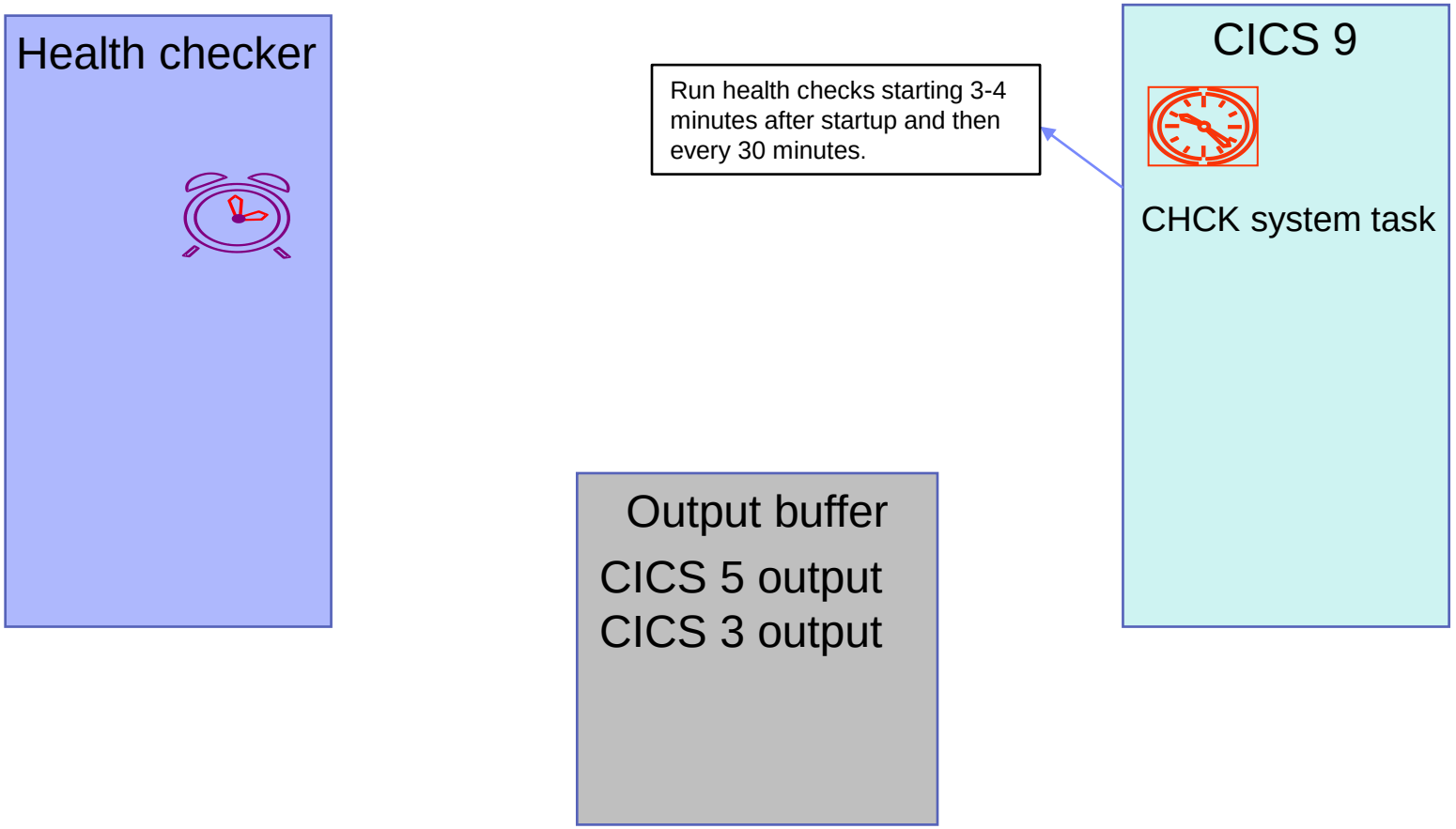
Output buffer

CICS 5 output

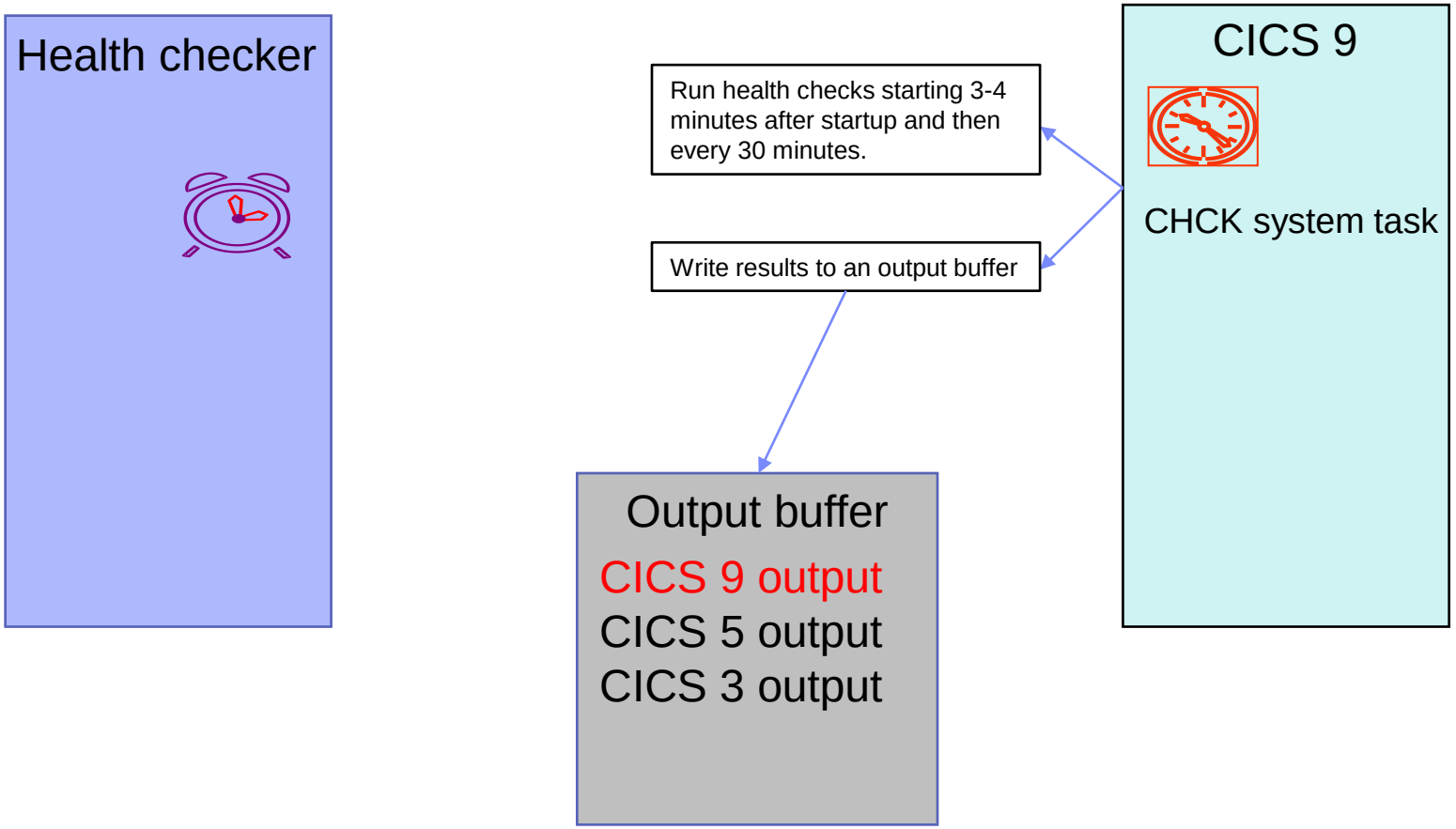
CICS 3 output

CICS 9

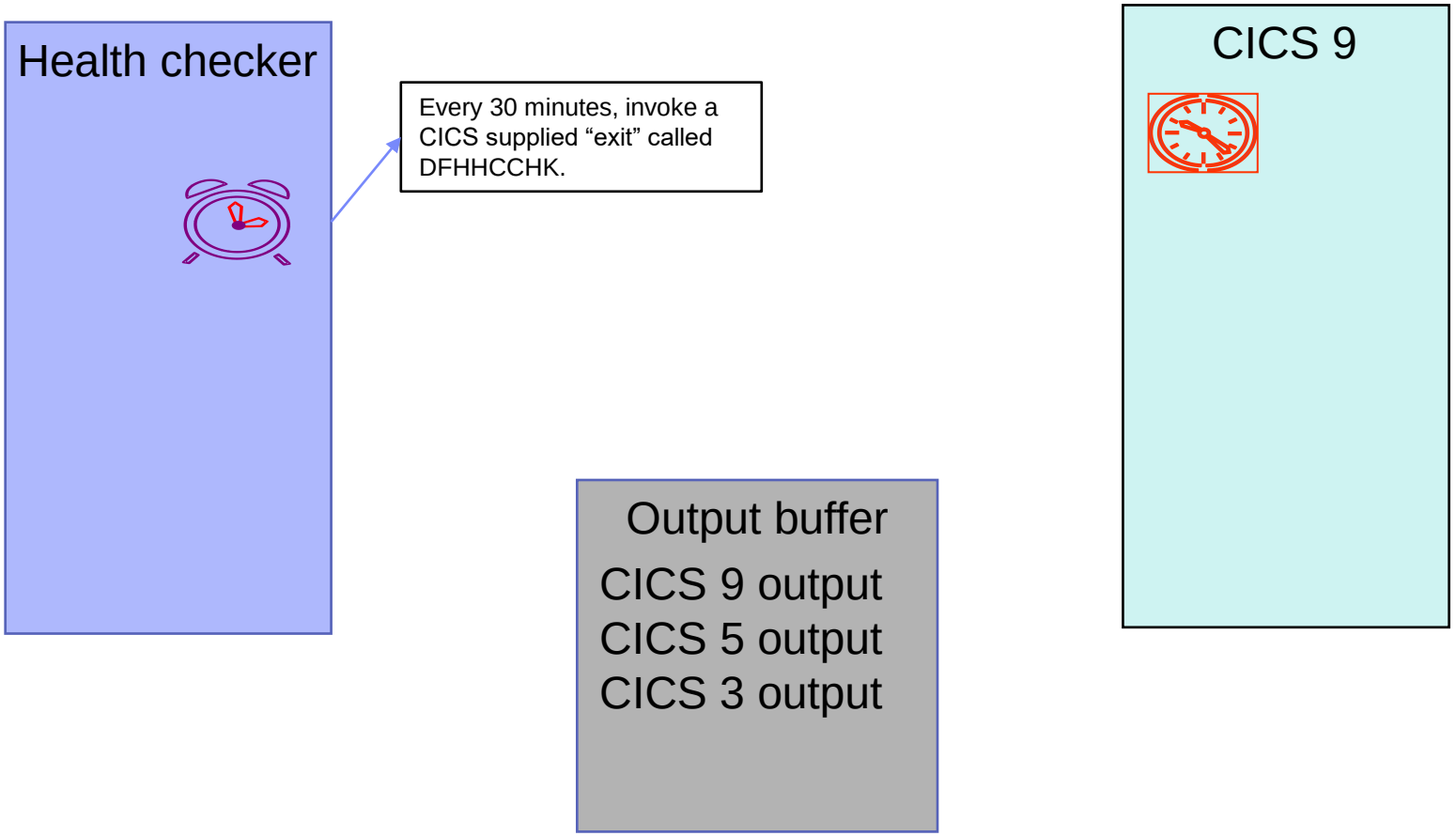
3 to 4 minutes later



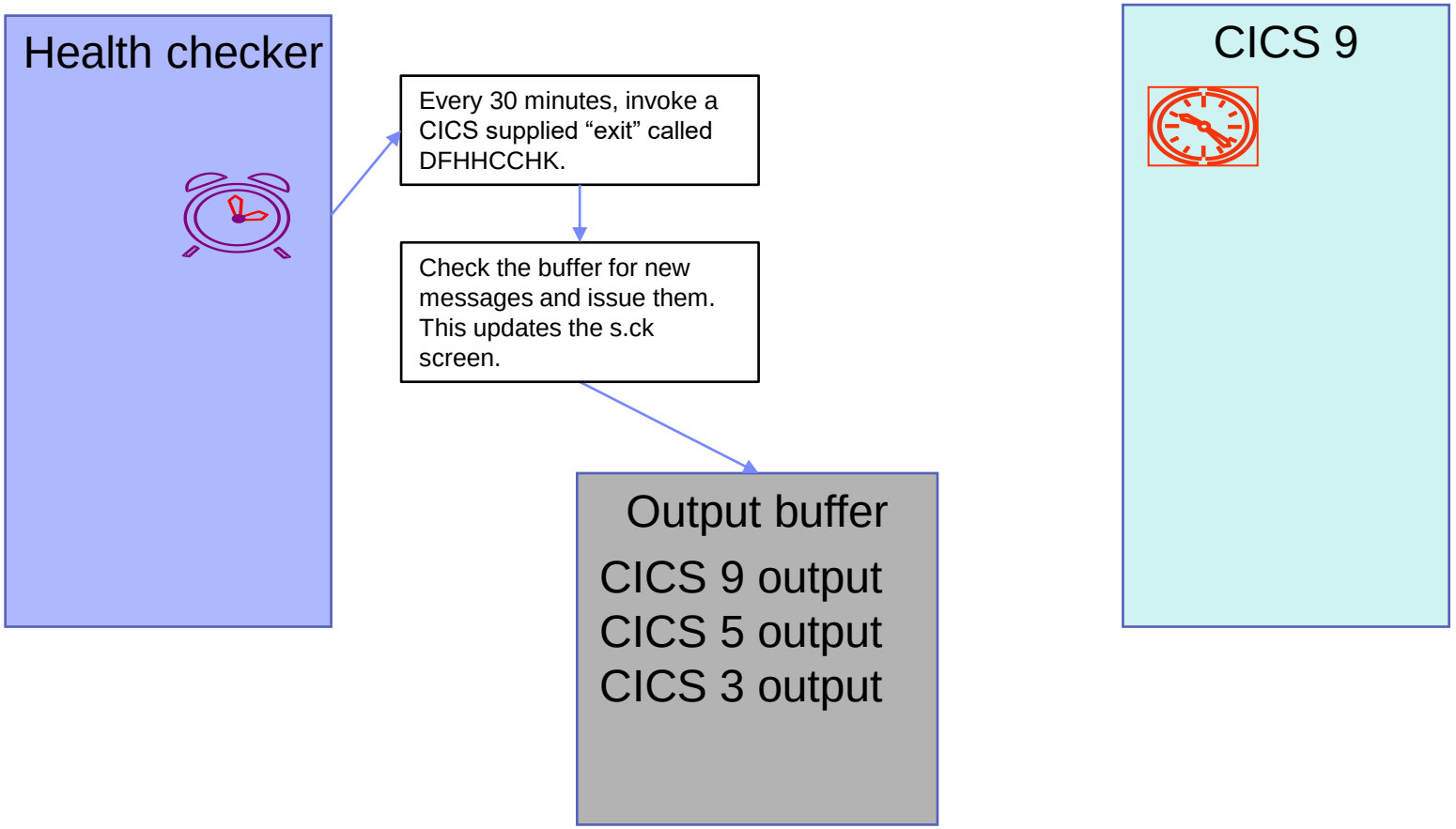
3 to 4 minutes later



Possibly several minutes later



Possibly several minutes later



Health Checker message updates

- Updates in S.CK output might not be as quick as you expect
- A change is made in CICS which will result in a health checker message
  - This change will be detected the next time CICS runs the health checks, which could be nearly 30 minutes later
- The health checker will not detect the change in CICS (via the message buffer) until it next checks for updates
  - This could be nearly 30 minutes after CICS has updated the message buffer

s.ck screen

```
SDSF HEALTH CHECKER DISPLAY  (ALL)                  LINE 1-22 (2028)
COMMAND INPUT ===>                                         SCROLL ===> CSR
NP   NAME                            CheckOwner       State              Status             Result Diag1    Diag2    DiagFrom Glob
     ALLOC_ALLC_OFFLN_POLICY         IBMALLOC         ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     ALLOC_SPEC_WAIT_POLICY          IBMALLOC         ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     ALLOC_TIOT_SIZE                 IBMALLOC         ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     ASM_LOCAL_SLOT_USAGE            IBMASM           ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     ASM_NUMBER_LOCAL_DATASETS       IBMASM           ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     ASM_PAGE_ADD                    IBMASM           ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     ASM_PLPA_COMMON_SIZE            IBMASM           ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     ASM_PLPA_COMMON_USAGE           IBMASM           ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     CATALOG_IMBED_REPLICATE         IBMCATALOG       ACTIVE(ENABLED)    EXCEPTION-LOW           4 00000000 00000000          NO
     CATALOG_RNLS                    IBMCATALOG       ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
 S   CICS_CEDA_ACCESS                IBMCICS          ACTIVE(ENABLED)    EXCEPTION-LOW           4 00000000 00000000          NO
     CICS_JOBSUB_SPOOL               IBMCICS          ACTIVE(ENABLED)    EXCEPTION-LOW           4 00000000 00000000          NO
     CICS_JOBSUB_TDQINTRDR           IBMCICS          ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     CNZ_AMRF_EVENTUAL_ACTION_MSGS   IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     CNZ_CONSOLE_MASTERAUTH_CMDSYS   IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     CNZ_CONSOLE_MSCOPE_AND_ROUTCODE IBMCNZ           ACTIVE(ENABLED)    EXCEPTION-LOW           4 00000000 00000000          NO
     CNZ_CONSOLE_OPERATING_MODE      IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          YES
     CNZ_CONSOLE_ROUTCODE_11         IBMCNZ           INACTIVE(ENABLED)  INACTIVE                0 00000000 00000000          NO
     CNZ_EMCS_HARDCOPY_MSCOPE        IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     CNZ_EMCS_INACTIVE_CONSOLES      IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          YES
     CNZ_OBSOLETE_MSGFLD_AUTOMATION  IBMCNZ           ACTIVE(ENABLED)    SUCCESSFUL              0 00000000 00000000          NO
     CNZ_SYSCONS_ALLOWCMD            IBMCNZ           INACTIVE(ENABLED)  INACTIVE                0 00000000 00000000          NO
```

```
********************************* TOP OF DATA ****************************
CHECK(IBMCICS,CICS_CEDA_ACCESS)
SYSPLEX:    PLEX2     SYSTEM: MV2C
START TIME: 05/10/2017 11:37:00.074340
CHECK DATE: 20170314  CHECK SEVERITY: LOW

* Low Severity Exception *

DFHH0001E The CEDA transaction is accessible to unauthenticated users.

  Explanation:  The IBM supplied transaction CEDA is accessible to the
    default user or CICS security is turned off.

    This means anyone who can connect to the IP address and port number
    of one of the CICS regions listed below can change the configuration
    of CICS.

    The regions listed below have a RC/RSN with more specific
    information about why the region failed the check:
    0801 CEDA installed and SEC=NO
    0802 CEDA installed and DFLTUSER can run it
    0807 The checking module can't be linked to
 System Action:  The system continues processing.

  Operator Response:  Report this error to the System Programmer.

  System Programmer Response:  Correct the security exposure which has
    been identified.

  Problem Determination:  n/a

  Source:  CICS Transaction Server

  Reference Documentation:  Look at the CICS Knowledge Centre for advice
    on best practice

  Automation:  n/a

  Check Reason:  CICS Health Check DFHHCC1
Check run (local time)     Jobname  ASID Applid   Regionid Ver  RcRn
05/10/2017 11:07:00.395096 CMASC111 007D C5407101 HODGINS  0710 0801 37
05/10/2017 11:07:20.515390 IYK3Z0VA 006A IYK3Z0VA LIQIN    0710 0801 7
05/10/2017 11:11:00.720032 CICSDEV1 007E IYK2Z3B1 WHARMBY  0710 0801 48
xxxxxxxxxxx 35 lines deleted xxxxxxxxxx

Number of regions checked  = 38

Number of regions reported = 38


END TIME: 05/10/2017 11:37:00.078608  STATUS: EXCEPTION-LOW
```

Time check run

CICS Region

CICS Release

Reason for reporting

Number times check run

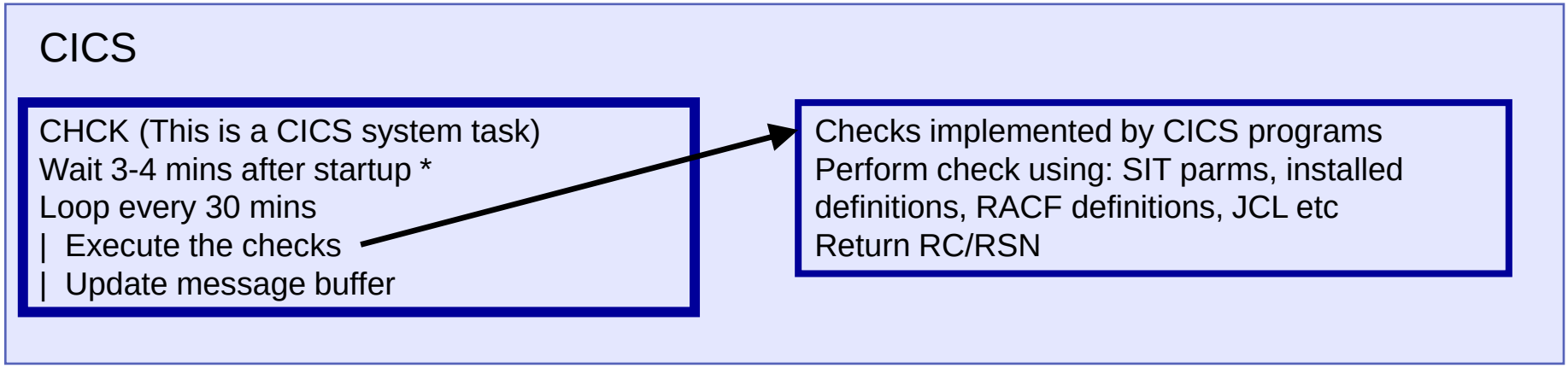
Summary details

Message output via batch job

- The output for a check (or checks) can be obtained by submitting a batch job
  - uses the HZSPRINT utility
  - useful if IPCS is not installed on your system
- Enables output to be checked programmatically
  - e.g. in automated testing

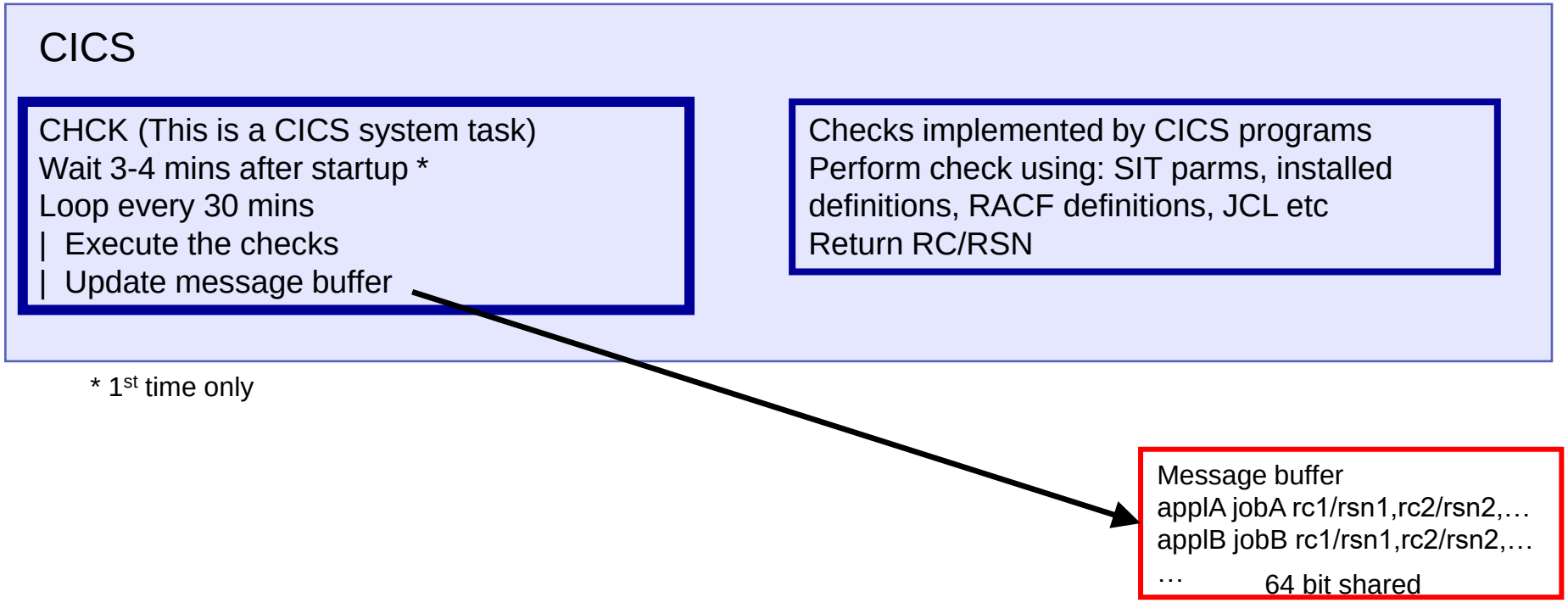
```
//HZSPRINT EXEC
PGM=HZSPRNT,TIME=1440,REGION=0M,PARMDD=SYSIN
//SYSIN DD *,DLM='@@'
CHECK(IBMCICS,CICS_REGION_CONFIGURATION)
,TIMERANGE(202303011200,20230301200)
@@
//SYSOUT DD SYSOUT=A,DCB=(LRECL=256)
```

Architecture



* 1st time only

Architecture



Architecture

CICS

CHCK (This is a CICS system task)
Wait 3-4 mins after startup *
Loop every 30 mins
| Execute the checks
| Update message buffer

Checks implemented by CICS programs
Perform check using: SIT parms, installed definitions, RACF definitions, JCL etc
Return RC/RSN

* 1st time only

Message buffer
applA jobA rc1/rsn1,rc2/rsn2,…
applB jobB rc1/rsn1,rc2/rsn2,…
…
64 bit shared

Health Checker

Read message buffer
Write health checker messages (to s.ck output)

Architecture

CICS

CHCK (This is a CICS system task)
Wait 3-4 mins after startup *
Loop every 30 mins
| Execute the checks
| Update message buffer

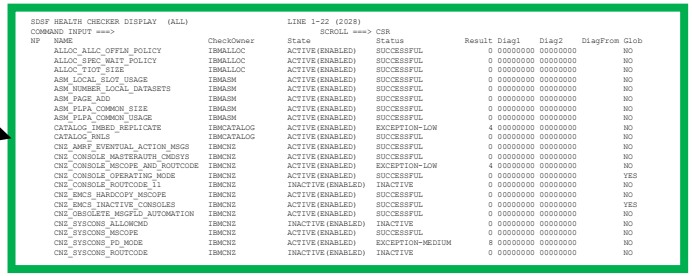
Checks implemented by CICS programs
Perform check using: SIT parms, installed definitions, RACF definitions, JCL etc
Return RC/RSN

* 1st time only

Message buffer
applA jobA rc1/rsn1,rc2/rsn2,…
applB jobB rc1/rsn1,rc2/rsn2,…
…
64 bit shared

Health Checker

Read message buffer
Write health checker messages (to s.ck output)



CHCK System Transaction Initialisation

- Defined and attached at CICS region startup (no CSD definition)
- Waits for 3-4 minutes (randomly)
  - Avoids IPL spike if many regions start at the same time
- Functions performed
  - Define storage to be used for the health checker messages
  - Define the CICS checks to z/OS
    - CSVDYNEX ADD to add exit DFHHCADD to health checker
    - HZSCHECK ADDNEW to tell health checker its definitions have changed

Heath Check Initialisation

- Health checker calls exit DFHHCADD
  - DFHHCADD is the CICS supplied HZSADDCHECK exit
    - Adds the checks to the set Health Checker knows about
    - Defines the check routine as DFHHCCHK
      - This module issues the required messages
    - Message table DFHHZMGT
- DFHHCADD, DFHHCCHK, DFHHZMGT are shipped in SDFHLINK
  - Must be in LINKLST to be available to health checker
  - Release independent

CHCK Health Checks

- System Transaction (CHCK)
- Runs checks every 30 minutes, why?
  - System may change
- Each check is a separate API enabled module
  - Commarea interface (2 bytes) to get rc,rsn
- Write single record with rc,rsn for all functions run

Health Checker Local Check - DFHHCCHK

- Same code used for all checks. Uses last record read
- Find last record written in message buffer
- For each record
  - If first error found, write error message DFHH000nE
  - Write report message DFHH0101I with address space
    - **Time of check (local time)**
    - **Jobname**
    - **ASID**
    - **Applid**
    - **Regionid**
    - **Version**
    - **RC/RSN**
    - **Number of times check run for this region**
- If no error write success message DFHH030nI

Health Checker Message Table

- Load Module DFHHZMGT in SDFHLINK
  - This contains message text (provided by CICS) for the z/OS health checker messages

- Three types of messages (these appear inside check)
  - Exception/OK messages
    - Describe check, whether it worked and action required to fix
  - Report
    - Used to list applids which fail
  - Debug
    - Activate by:
      F hzsproc,UPDATE,CHECK=(IBMCICS,CICS_CEDA_ACCESS),DEBUG
    - Output extra diagnostic information such as addresses. Only of use to IBM service team

Health Checker Commands

- Following are examples of common commands:
  - To run a check
    - f hzsproc,run,CHECK=(IBMCICS,CICS_JOBSUB_TDQINTRDR)
  - To turn off a check
    - f hzsproc,update,CHECK=(IBMCICS,CICS_JOBSUB_SPOOL),inactive
  - To turn on debug mode for a check
    - f hzsproc,update,CHECK=(IBMCICS,CICS_CEDA_ACCESS),debug

Running in debug mode

- Health checks can run in debug mode

```
f hzsproc,update,CHECK=(IBMCICS,CICS_CEDA_ACCESS),debug
```

  - results in extra “debug” messages being issued
    - Mainly of interest to IBM service team

- To turn off debug mode

```
f hzsproc,update,CHECK=(IBMCICS,CICS_CEDA_ACCESS),debug=OFF
```

Verbose mode

- Health checks can be run in “verbose” mode

```
f hzsproc,update,CHECK=(IBMCICS,CICS_CEDA_ACCESS),verbose=yes
```

  - this turns on more messages, for example RACF checks will add messages when in this mode. This will not always result in extra messages, it depends what the check actually does.

- Debug and Verbose are two independent options for a health check

Performance Impact

Table 1. Summary of CICS statistics for Health Checker CPU measurement

	CHCK disabled	CHCK enabled	Difference
Address space TCB time	0.003888	0.035986	0.032098
Address space SRB time	0.000349	0.000538	0.000189
Total CPU time	0.004237	0.036524	0.032287

Data from the CICS TS 5.4 performance report

- The table shows the results for 60 iterations of CHCK.
- One iteration of CHCK uses about 540 µs of CPU.
- In normal use, this means 1 ms of CPU per hour.
- An idle CICS with health checks disabled consumes approximately 50 ms per hour for CICS background processing
- The conclusion is that the health checks add no significant overhead to a CICS region

New in CICS TS 6.1

New z/OS Health Checker tests added in CICS TS 6.1

- 5 new tests appear on the s.ck output making 8 in total for CICS

```
SDSF HEALTH CHECKER DISPLAY  MV2C                          LINE 1-58 (253)
COMMAND INPUT ===> _                                         SCROLL ===> DATA
PREFIX=*   DEST=(ALL)   OWNER=DBEARD*   SYSNAME=
NP    NAME                                CheckOwner       State                Status
      ALLOC_ALLC_OFFLN_POLICY             IBMALLOC         ACTIVE(ENABLED)      SUCCESSFUL
      ALLOC_SMSHONOR_STATE                IBMALLOC         ACTIVE(ENABLED)      SUCCESSFUL
      ALLOC_SPEC_WAIT_POLICY              IBMALLOC         ACTIVE(ENABLED)      SUCCESSFUL
      ALLOC_TAPELIB_PREF                  IBMALLOC         ACTIVE(ENABLED)      EXCEPTION-LOW
      ALLOC_TIOT_SIZE                     IBMALLOC         ACTIVE(ENABLED)      SUCCESSFUL
      ASM_LOCAL_SLOT_USAGE                IBMASM           ACTIVE(ENABLED)      EXCEPTION-MEDIUM
      ASM_NUMBER_LOCAL_DATASETS           IBMASM           ACTIVE(ENABLED)      SUCCESSFUL
      ASM_PAGE_ADD                        IBMASM           ACTIVE(ENABLED)      SUCCESSFUL
      ASM_PLPA_COMMON_SIZE                IBMASM           ACTIVE(ENABLED)      SUCCESSFUL
      ASM_PLPA_COMMON_USAGE               IBMASM           ACTIVE(ENABLED)      SUCCESSFUL
      CATALOG_ATTRIBUTE_CHECK             IBMCATALOG       ACTIVE(ENABLED)      SUCCESSFUL
      CATALOG_IMBED_REPLICATE             IBMCATALOG       ACTIVE(ENABLED)      EXCEPTION-LOW
      CATALOG_RNLS                        IBMCATALOG       ACTIVE(ENABLED)      SUCCESSFUL
      CICS_CAT3_CONFIGURATION             IBMCICS          ACTIVE(ENABLED)      SUCCESSFUL
      CICS_CEDA_ACCESS                    IBMCICS          ACTIVE(ENABLED)      EXCEPTION-LOW
      CICS_JOBSUB_SPOOL                   IBMCICS          ACTIVE(ENABLED)      EXCEPTION-LOW
      CICS_JOBSUB_TDQINTRDR               IBMCICS          ACTIVE(ENABLED)      SUCCESSFUL
      CICS_REGION_CONFIGURATION           IBMCICS          ACTIVE(ENABLED)      EXCEPTION-LOW
      CICS_RESOURCE_CONFIGURATION         IBMCICS          ACTIVE(ENABLED)      EXCEPTION-LOW
      CICS_RESOURCE_SECURITY              IBMCICS          ACTIVE(ENABLED)      EXCEPTION-LOW
      CICS_USS_CONFIGURATION              IBMCICS          ACTIVE(ENABLED)      EXCEPTION-LOW
      CIM_HTTPS_ENABLED                   IBMCIM           ACTIVE(ENABLED)      SUCCESSFUL
      CNZ_AMRF_EVENTUAL_ACTION_MSGS       IBMCNZ           ACTIVE(ENABLED)      SUCCESSFUL
      CNZ_CONSOLE_MASTERAUTH_CMDSYS       IBMCNZ           ACTIVE(ENABLED)      SUCCESSFUL
      CNZ_CONSOLE_MSCOPE_AND_ROUTCODE     IBMCNZ           ACTIVE(ENABLED)      EXCEPTION-LOW
      CNZ_CONSOLE_ROUTCODE_11             IBMCNZ           INACTIVE(ENABLED)    INACTIVE
      CNZ_EMCS_HARDCOPY_MSCOPE            IBMCNZ           ACTIVE(ENABLED)      SUCCESSFUL
      CNZ_EMCS_INACTIVE_CONSOLES          IBMCNZ           ACTIVE(ENABLED)      SUCCESSFUL
      CNZ_OBSOLETE_MSGFLD_AUTOMATION      IBMCNZ           ACTIVE(ENABLED)      SUCCESSFUL
      CNZ_SYSCONS_ALLOWCMD                IBMCNZ           INACTIVE(ENABLED)    INACTIVE
      CNZ_SYSCONS_MSCOPE                  IBMCNZ           ACTIVE(ENABLED)      SUCCESSFUL
      CNZ_SYSCONS_PD_MODE                 IBMCNZ           ACTIVE(ENABLED)      EXCEPTION-MEDIUM
      CNZ_SYSCONS_ROUTCODE                IBMCNZ           INACTIVE(ENABLED)    INACTIVE
      CNZ_TASK_TABLE                      IBMCNZ           ACTIVE(ENABLED)      SUCCESSFUL
      CSAPP_FTPD_ANONYMOUS_JES            IBMCS            ACTIVE(ENABLED)      SUCCESSFUL
      CSAPP_MVRSHD_RHOSTS_DATA            IBMCS            ACTIVE(ENABLED)      SUCCESSFUL
      CSAPP_SNMPAGENT_PUBLIC_COMMUNITY    IBMCS            ACTIVE(ENABLED)      SUCCESSFUL
```

New tests

- Checks are grouped so as not to clutter the s.ck display
  - each new health check module can deal with up to 16 tests

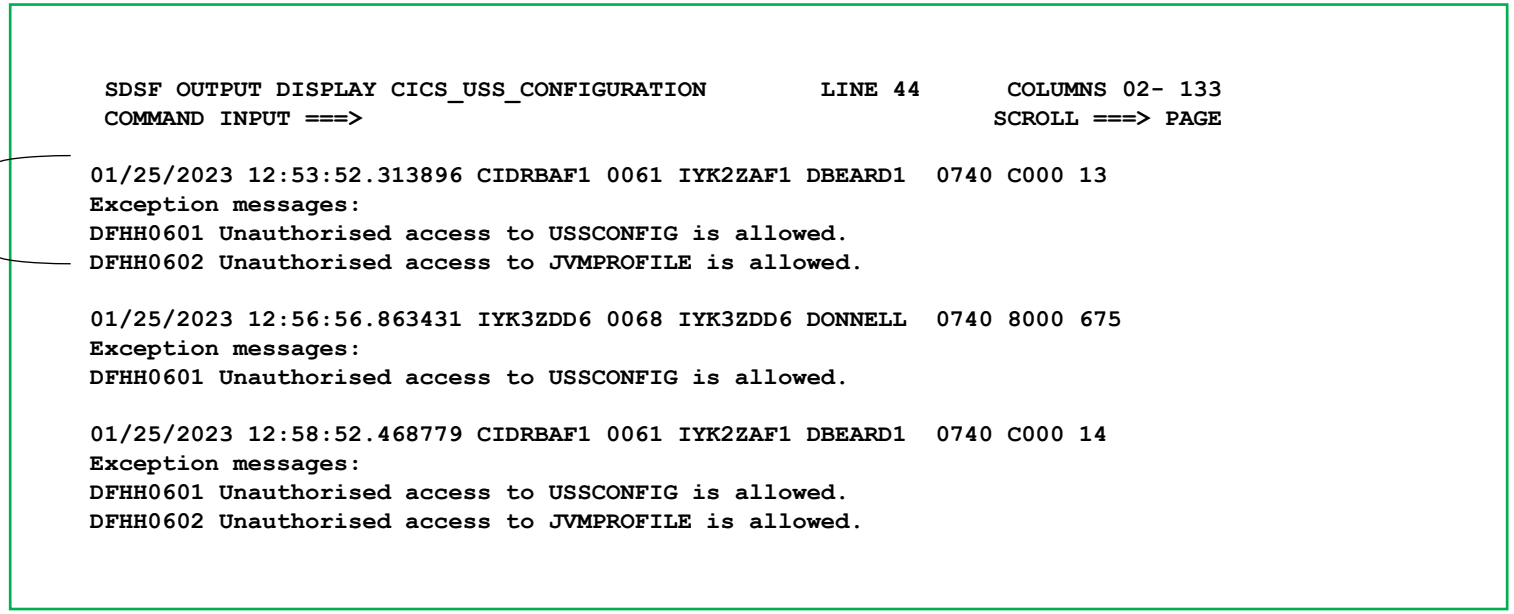
- Checks implemented
  - CICS_CAT3_CONFIGURATION (10 tests)
    - checks some transaction definitions
  - CICS_REGION_CONFIGURATION (14 tests)
    - predominantly SIT parameter checks
  - CICS_RESOURCE_CONFIGURATION (4 tests)
    - checks on some installed resources
  - CICS_RESOURCE_SECURITY (8 tests)
    - checks on a lot of the X… SIT parameters
  - CICS_USS_CONFIGURATION (3 tests)
    - checks on access to various USS files

New style messages

```
 SDSF OUTPUT DISPLAY CICS_USS_CONFIGURATION        LINE 44       COLUMNS 02- 133
 COMMAND INPUT ===>                                            SCROLL ===> PAGE

01/25/2023 12:53:52.313896 CIDRBAF1 0061 IYK2ZAF1 DBEARD1  0740 C000 13
Exception messages:
DFHH0601 Unauthorised access to USSCONFIG is allowed.
DFHH0602 Unauthorised access to JVMPROFILE is allowed.

01/25/2023 12:56:56.863431 IYK3ZDD6 0068 IYK3ZDD6 DONNELL  0740 8000 675
Exception messages:
DFHH0601 Unauthorised access to USSCONFIG is allowed.

01/25/2023 12:58:52.468779 CIDRBAF1 0061 IYK2ZAF1 DBEARD1  0740 C000 14
Exception messages:
DFHH0601 Unauthorised access to USSCONFIG is allowed.
DFHH0602 Unauthorised access to JVMPROFILE is allowed.
```

New style messages

Grouped by CICS region



```
 SDSF OUTPUT DISPLAY CICS_USS_CONFIGURATION        LINE 44       COLUMNS 02- 133
 COMMAND INPUT ===>                                            SCROLL ===> PAGE

01/25/2023 12:53:52.313896 CIDRBAF1 0061 IYK2ZAF1 DBEARD1  0740 C000 13
Exception messages:
DFHH0601 Unauthorised access to USSCONFIG is allowed.
DFHH0602 Unauthorised access to JVMPROFILE is allowed.

01/25/2023 12:56:56.863431 IYK3ZDD6 0068 IYK3ZDD6 DONNELL  0740 8000 675
Exception messages:
DFHH0601 Unauthorised access to USSCONFIG is allowed.

01/25/2023 12:58:52.468779 CIDRBAF1 0061 IYK2ZAF1 DBEARD1  0740 C000 14
Exception messages:
DFHH0601 Unauthorised access to USSCONFIG is allowed.
DFHH0602 Unauthorised access to JVMPROFILE is allowed.
```

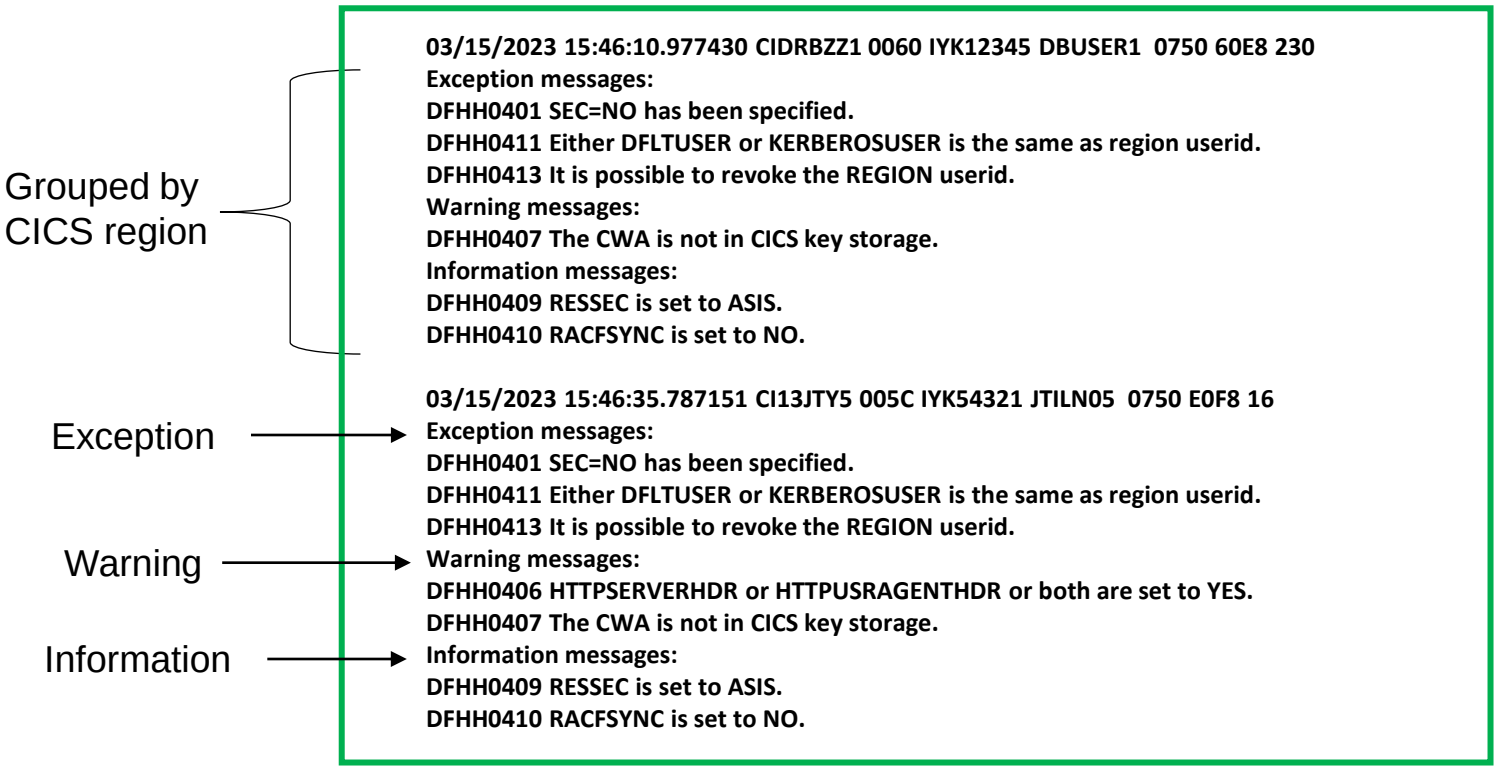
New style messages

Grouped by CICS region

```
 SDSF OUTPUT DISPLAY CICS_USS_CONFIGURATION        LINE 44       COLUMNS 02- 133
 COMMAND INPUT ===>                                             SCROLL ===> PAGE

01/25/2023 12:53:52.313896 CIDRBAF1 0061 IYK2ZAF1 DBEARD1  0740 C000 13
Exception messages:
DFHH0601 Unauthorised access to USSCONFIG is allowed.
DFHH0602 Unauthorised access to JVMPROFILE is allowed.

01/25/2023 12:56:56.863431 IYK3ZDD6 0068 IYK3ZDD6 DONNELL  0740 8000 675
Exception messages:
DFHH0601 Unauthorised access to USSCONFIG is allowed.

01/25/2023 12:58:52.468779 CIDRBAF1 0061 IYK2ZAF1 DBEARD1  0740 C000 14
Exception messages:
DFHH0601 Unauthorised access to USSCONFIG is allowed.
DFHH0602 Unauthorised access to JVMPROFILE is allowed.
```

Each error indicated by a unique message

New style messages



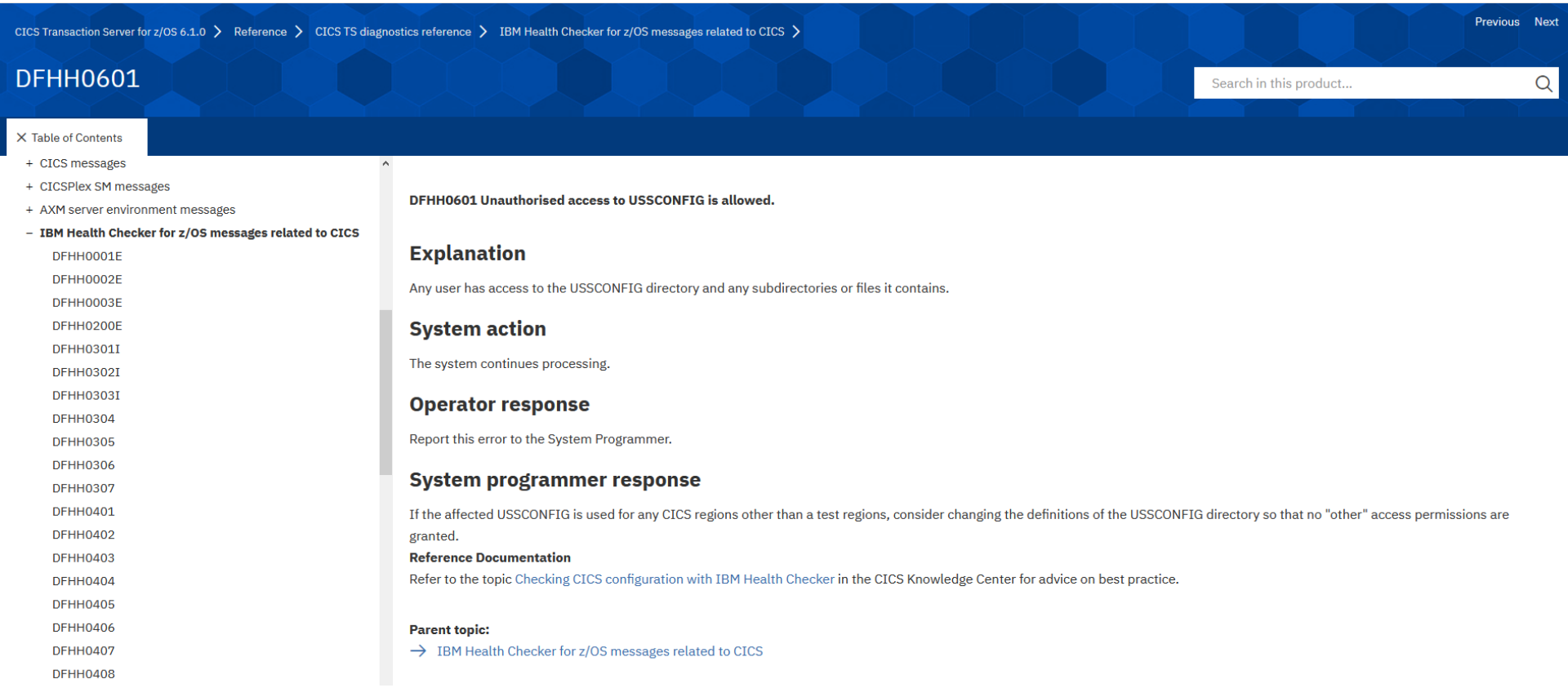
Messages are grouped as Exception, Warning or Information messages

New information

```
Number of regions checked  = 72

Number of regions reported = 69



Number of exception messages issued    = 105

Number of warning messages issued      = 139

Number of information messages issued  = 140
```

The number of exception, warning and information messages is given at the bottom of the s.ck output for a check.

Documentation



Every new message is documented along with details of what is wrong and what corrective measures should be considered.

Changes during initialization

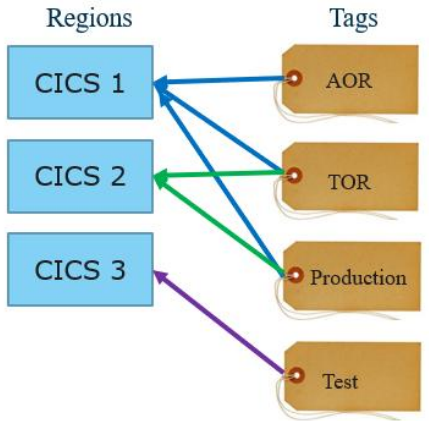
- A CICS TS 6.1 level of CICS needs to know that a CICS TS 6.1 or higher level system has defined the health checker checks
  - so that there are 8 and not only 3 defined
- A CICS TS 6.1 system will check to see which release level has performed the setup
  - if CICS TS 6.1 or higher, do nothing
  - if back level region, re-initialize the checks and buffers

Controlling the new tests

- It might be required to control which z/OS Health Checker checks are to be run (or not) against CICS regions
  - 38 new checks have been added in CICS TS 6.1 and it might not be necessary to run all of these
    - You might not want to conform to all of the recommendations implicit in the checks for some in-house reason
    - It might be nice to turn off one or more of the checks
  - Tagging files enable this control

Tagging file

- A file which allows tags (or labels) to be associated with CICS regions
- Enable individual health checks to be turned off
- Enable CICS regions to be excluded from all health checks



Exclude specific health checks for all regions

```
# It is possible to exclude some specific health checks for all CICS regions which use this tagging file.
# A check to be excluded is identified by the message number which the check would issue if it failed.
# The Check text string must begin with the message number. Other text is optional.
# To exclude a check, uncomment the cicsts-ExcludedHealthChecks label and the relevant line or lines below.
#
- cicsts-ExcludedHealthChecks:
#   - Check: "DFHH0401 SEC=NO has been specified."
#   - Check: "DFHH0402 XTRAN=NO has been specified."
#   - Check: "DFHH0403 XUSER=NO has been specified."
#   - Check: "DFHH0404 CONFDATA=SHOW has been specified."
    - Check: "DFHH0405 Do not run this one MINTLSLEVEL lower than 1.2 has been specified."
#   - Check: "DFHH0406 HTTPSERVERHDR or HTTPUSRAGENTHDR or both are set to YES."
#   - Check: "DFHH0407 The CWA is not in CICS key storage."
#   - Check: "DFHH0409 RESSEC is set to ASIS."
#   - Check: "DFHH0410 RACFSYNC is set to NO."
#   - Check: "DFHH0411 Either DFLTUSER or KERBEROSUSER is the same as region userid."
    - Check: "DFHH0412 GMTRAN allows users access to the default screen."
#   - Check: "DFHH0413 It is possible to revoke the REGION userid."
#   - Check: "DFHH0414 XCMD=NO has been specified."
#   - Check: "DFHH0501 Default user can execute sensitive transactions"
#   - Check: "DFHH0502 Default URM DFHISAIP is being used in IPIC TCPIPSERVICEs."
```

Specific checks can be excluded for all CICS regions but others will be run.

What you will see in CICS

DFHPA2206 I IYK12345
31/01/2023 09:29:41 IYK12345 The region will not run z/OS health check DFHH0405 due to specifications in the cicstags.yaml file.

DFHPA2206 I IYK12345
31/01/2023 09:29:41 IYK12345 The region will not run z/OS health check DFHH0412 due to specifications in the cicstags.yaml file.

The joblog will show which health checks have been excluded.

The health checks are identified by the message number of the message which would be issued if the test were to fail.

DFHH or DFHHC

- Do not confuse messages
  - DFHHxxxx
    - Issued by Health Checker
    - Cannot be suppressed by a CICS message exit (XMEOUT)
    - Most can be prevented by using a tagging file setting

      DFHH0401 SEC=NO has been specified.

  - DFHHCxxxx
    - Issued by CICS message domain
    - Can be suppressed by using XMEOUT
    - Tagging file settings will have no effect on these messages

      DFHHC0101I IYK12345 CICS has registered successfully to the z/OS Health Checker.

Turning off all health checks

- It may be required to exclude some CICS regions from health checks completely
  - For example, any region tagged as development or test does not need to be subject to checking

- Tagging files will let you do this

Exclude CICS regions from health checks

```
# HealthCheckExcludedRegions specifies the region usage types of regions which are to be completely
# excluded from health checks. The ExcludeUsage types to be excluded must have been defined in the
# RegionUsage section of this file.
#
- cicsts-HealthCheckExcludedRegions:
  - ExcludeUsage: "Development"
  - ExcludeUsage: "Test_regions"
#
```

Regions can be completely excluded from z/OS Health Checker checks based on usage. For example, all development or test regions.

What you will see in CICS

DFHPA2205 I IYK12345
31/01/2023 16:01:01 IYK12345 This region is excluded from z/OS Health Checker checks due to specifications in the cicstags.yaml file.

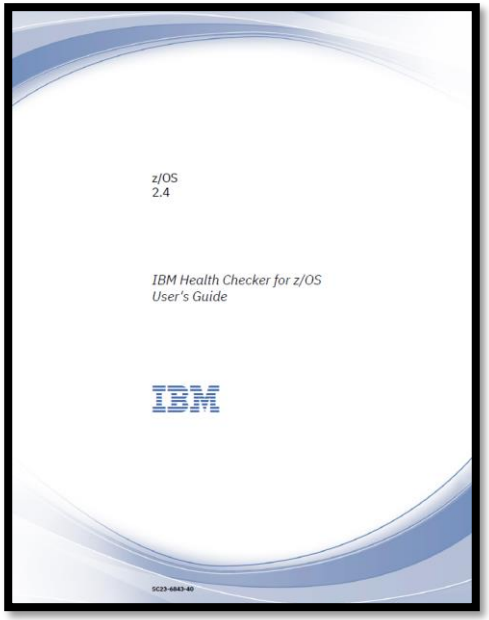
The joblog will show that the region has been completely excluded from health checks.

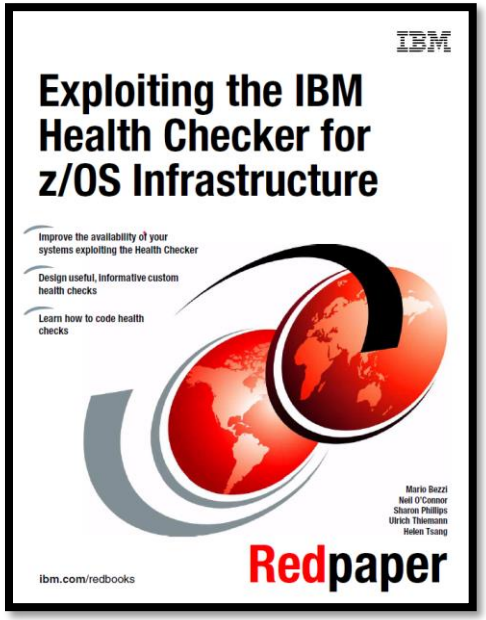
CEMT INQUIRE SYSTEM

```
I SYS
  STATUS:  RESULTS - OVERTYPE TO MODIFY
    System
    Aging( 00500 )
    Aidcount(00000)
    Akp( 04000 )
    Cicstslevel(060100)
    Cmdprotect(Cmdprot)
    Db2conn()
    Debugtool( Debug )
    Dfltuser(CICSUSER)
    Dsalimit( 05242880 )
    Dsrtprogram( NONE     )
    Dtrprogram( DFHDYP    )
    Dumping( Sysdump )
    Edsalimit( 0629145600 )
    Forceqr( Noforce )
    Healthcheck(Nocheck)
    Lastcoldtime(                )
  +  Lastemertime(                )

                                                                SYSID=CICS APPLID=IYK12345
   RESPONSE: NORMAL                                  TIME: 16.05.23  DATE: 28/02/23
 PF 1 HELP              3 END              5 VAR              7 SBH    8 SFH    9 MSG    10 SB    11 SF
```

Further information

- For further details see IBM Documentation
  - CICS TS 5.4 or later
    - Support for IBM Health Checker for z/OS
  - IBM Health Checker for z/OS User's Guide (SC23-6843-40)
  - Exploiting the Health Checker for z/OS infrastructure (REDP-4590)





Summary

- No configuration is required
  - CICS will set up the checks and z/OS health checker is on by default
- The checks test for recommended best practise
  - SIT parameters
  - Some resource definitions
  - Some region definitions
- Failure of checks only results in messages
  - No actions are taken automatically should tests fail

Any questions