# Upgrading to CICS Transaction Server for z/OS, Version 4.2

This topic describes our experiences during the upgrade from CICS® Transaction Server for z/OS® (CICS TS) Version 4.1 to Version 4.2 in our Parallel Sysplex® environment. It is not intended to be a step-by-step procedure because each migration is unique due to the CICS configuration and features installed.

## Applicable CICS TS documentation

During the migration to CICS TS 4.2, we used the following documentation:

- *Program Directory for CICS Transaction Server for z/OS V4.2.0*, GI13-0565
- *CICS Transaction Server for z/OS V4.2 Upgrading from CICS Transaction Server 4.1*, GC34-7190
- *CICS Transaction Server for z/OS V4.2 Installation Guide*, GC34-7171
- *CICS Transaction Server for z/OS V4.2 Messages and Codes VOL 1*, GC34-7175
- *CICS Transaction Server for z/OS V4.2 Messages and Codes VOL 2*, GC34-7176
- *CICS Transaction Server for z/OS V4.2 CICS Operations and Utilities Guide*, SC34-7213
- *CICS Transaction Server for z/OS V4.2 CICSPlex SM Messages and Codes*, GC34-7201

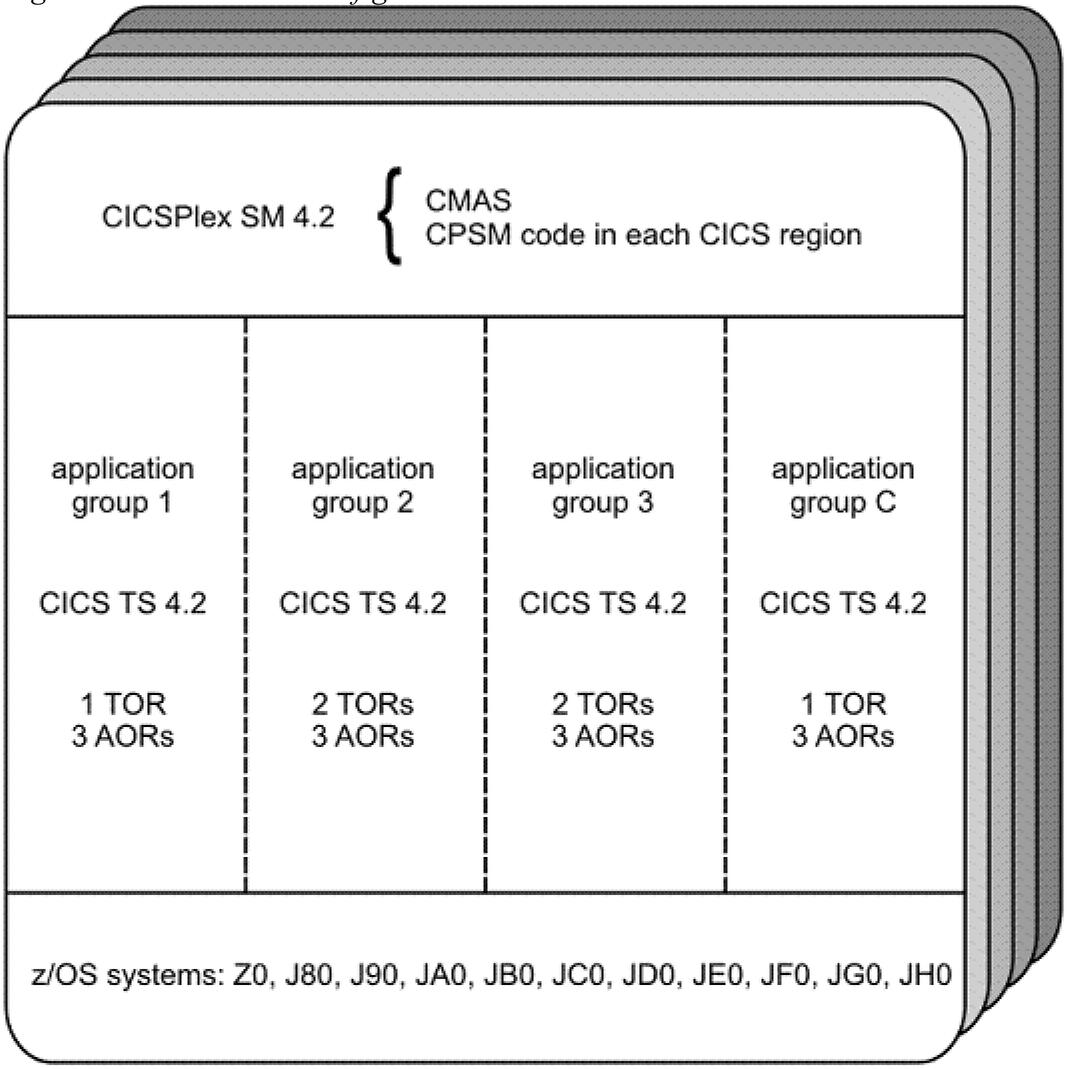
The CICS documentation is also available in information center format at publib.boulder.ibm.com/infocenter/cicsts/v4r2/index.jsp.

## Overview of upgrading to CICS TS 4.2

Our goal with all of our migrations and upgrades is to follow the path of a typical customer. We upgraded slowly across our test sysplex and within the workloads on that sysplex. This created a mix of releases within a system as well as across the CICSPlex®. We did this to test as many coexistence and operational combinations as possible. After we completed the upgrade steps for the CICSPlex on our test sysplex, we followed a similar upgrade strategy for the CICSPlex across our pseudo-production sysplex.

Figure 1 illustrates the four main application groups in our Parallel Sysplex.

*Figure 1. Our CICS TS configuration*



The application groups process the following types of work:

- Application group 1: IMS™/DBCTL
- Application group 2: CICS/VSAM (both RLS and non-RLS)
- Application group 3: DB2® and WebSphere® MQ
- Application group C: cryptographic functions, Java, C++, and others

# Preparing to upgrade to CICS TS 4.2

Before we began the actual upgrade process, we did some preparatory work in the following areas:

- **Backing up our data**

Even though we created new files and data sets, as a precaution, we first took backups of all of the CSDs and data repositories.

- **Defining aliases**

  We defined new catalog aliases for CICS TS 4.2.

- **Loading the CICS TS 4.2 product libraries**

  We set up and ran the SMP/E jobs to load the CICS TS 4.2 product libraries. We then ran a copy job to bring the build libraries over to our production systems.

- **Allocating supporting data sets**

  We created copies of all of our supporting libraries (JCL, SYSIN, TABLEs, and so on). We reviewed these and updated accordingly with all the necessary CICS TS 4.2 changes.

- **Customizing the CICS region data sets**

  We customized the jobs in `*hlq*.SDFHINST(DFHDEFDS)` and `*hlq*.SEYUINST(EYUDEFDS)` to define all of the region data sets and submitted them a number of times. Depending on your environment, you might need to alter the default file sizes. We increased the file sizes for the `DFHGCD`, `DFHINTRA`, and `DFHTEMP` data sets. Being a test organization, we also increased the sizes of our auxtrace data sets.

- **Reviewing and reassembling tables**

  We reviewed and reassembled any tables that we had modified.

- **Updating SYS1.PARMLIB and APF-authorizing program libraries**

  In `SYS1.PARMLIB`, we updated the LINK list and LPA list. We APF-authorized the following program libraries:

  - *hlq*.SDFHAUTH
  - *hlq*.SDFHLINK
  - *hlq*.SDFJAUTH
  - *hlq*.SEYUAUTH
  - *hlq*.SEYULINK

  In `SYS1.PROCLIB`, we reviewed and updated all our procs for the CICS TS 4.2 changes.

# Upgrading CICSPlex SM

Starting with CICS TS 3.2, the CICSPlex SM (CPSM) installation is integrated with the CICS installation. Previously, it was a separate process. You can now modify the DFHISTAR job to change both the CICS and CICSPlex SM installation parameters.

If you are implementing CICSPlex SM for the first time, CPSM consists of the following parts on each system:

- CMAS (CICS-managed address space)
- CPSM code running in each CICS region (MAS), which communicates with the CMAS, sometimes referred to as the *agent* code.

In order for a CMAS and a MAS to communicate, they must be running the same version of CPSM.

You can run CPSM at different release levels across a CICSPlex, as long you follow the rules documented in *CICS Transaction Server for z/OS V4.2 Upgrading from CICS Transaction Server 4.1,* under the section "Conditions for running CICSPlex SM Version 4.2 and earlier releases concurrently." By adhering to these rules, we ran the following combinations at the same time across our CICSPlex/sysplex during our upgrade:

- 4.2 CMAS with 4.2 MAS
- 4.2 CMAS with 4.1 MAS
- 4.1 CMAS with 4.1 MAS

**Note:** DFHIRP must be at the highest level of the code in a system image, and the version of DFHIRP for CICS TS 4.2 can only be used on z/OS 1.7 and higher.

# Upgrading the CMASs

We did the following to upgrade the CMASs:

1. Defined a new CSD
2. Upgraded the CSD with CPSM 4.2 level resource definitions and the CICS startup group list. We did this by running the DFHCSDUP utility with the `UPGRADE` command, as documented in *CICS Transaction Server for z/OS V4.2 CICS Operations and Utilities Guide*.
3. Reviewed our CICS resource definition tables, which we had updated earlier
4. Converted the CPSM data repository to the CPSM 4.2 level by running the EYU9XDUT utility, as documented in *CICS Transaction Server for z/OS V4.2 Installation Guide*

**Note**: Be sure you have the fix for APAR PM53250 installed which fixes a problem where this utility took a very long time to run.

5. Reviewed the JCL in the EYUCMAS member for any changes to the CMAS startup procedure. We updated the data set names to use our new high-level qualifiers.
6. Updated the MAS startup procedures to point to the new CPSM data sets in order to identify the new CPSM code to the MAS regions

Our CMASs were then ready to start. Remember that the maintenance point CMAS must be the first CMAS to be upgraded to the new release.

# Upgrading the MASs

We reviewed the steps documented in *CICS Transaction Server for z/OS V4.2 Upgrading from CICS Transaction Server 4.1* to upgrade the MASs. Many of the steps are similar to the steps we followed to upgrade the CMASs.

We did the following to upgrade the MASs:

1. Defined a new CSD and copied our application groups from the old CSD
2. Upgraded the CSD for CICS TS 4.2. We also removed groups for previous releases of CPSM from the group list.

   **Note:** If you are also changing z/OS release levels, the LE definitions (on CSD) may need to be updated, too.

3. Reviewed our CICS resource definition tables, which were updated earlier
4. Copied the JCL for our MAS startup procedures and changed the library names to use our new high-level qualifiers
5. Reviewed the LE libraries we had in the RPL concatenation

The method for connecting to WebSphere MQ changed in CICS TS 4.1. Instead of defining default settings for the CICS - WebSphere MQ connection in the **`DFHMQPRM`** operand of an **`INITPARM`** system initialization parameter, you must now use the new MQCONN resource definition. You can use the MQCONN resource definition to specify a queue-sharing group, or you can use it to specify the name of a single queue manager. We set up our group 3 regions to specify our WebSphere MQ queue-sharing group and a shared INITQ (e.g. MQGT QSG for our test sysplex and MQGP QSG for our pseudo-production sysplex). We then start additional instances of CKTI using the **`CKQC STARTCKTI`** command at region startup.

Note the following points about upgrading to CICS TS 4.2:

1. With CICS TS 4.2, all JVMs run in a 64-bit environment. You must use the IBM 64-bit SDK for z/OS, Java Technology Edition, Version 6.01, to run Java workloads. See *CICS Transaction Server for z/OS V4.2 Upgrading from CICS Transaction Server 4.1*, in the

section titled "Key changes to CICS support for Java applications," for details about memory requirements and other information.

2. CICS TS 4.1 and higher no longer supports Java 1.4.2 or 1.5. If you are running an older version of Java you will need to upgrade your CICS Java environment to Java 6.
3. CICS TS 4.1 and higher no longer supports the DCT macro as a means of defining transient data queues. These must be defined in the CSD using TDQUEUE resource definitions.
4. Before CICS Transaction Server version 2.3, JVM profiles were stored in a PDS member. In CICS TS 2.3 and later, they are stored in a file system directory pointed to by the **`JVMPROFILEDIR`** system initialization parameter. If you are migrating from a release prior to CICS Transaction Server 2.3, you will need to make the appropriate Java changes. We keep our JVM profiles outside the file system shipped with CICS TS 4.2, so that they are not overridden with a CICS TS 4.2 file system at maintenance time.

Our MASs were then ready to start.

# Upgrading the CICSPlex SM Web User Interface

As documented in *CICS Transaction Server for z/OS V4.2 Upgrading from CICS Transaction Server 4.1*, both the CICSPlex SM Web User Interface (WUI) and the CMAS to which it connects must be at the highest level of CICSPlex SM within the CICSplex. This means that both must be at the same level as the maintenance point CMAS.

Because the CICS system that acts as the WUI is just another MAS, we used the same steps as for upgrading a MAS:

1. Upgraded the MAS that acts as the WUI
2. Upgraded the WUI CSD, as above
3. Upgraded the contents of the WUI server repository (EYUWREP), as documented in *CICS Transaction Server for z/OS V4.2 Upgrading from CICS Transaction Server 4.1*

# Experiences with upgrading to CICS TS 4.2

With the exception of some typos and unrelated sysplex issues, this upgrade went well. We like the new enhancements to some of the views in the WUI.