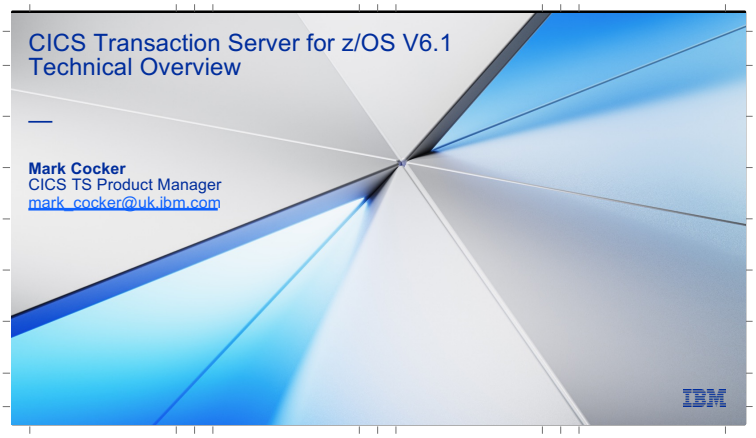
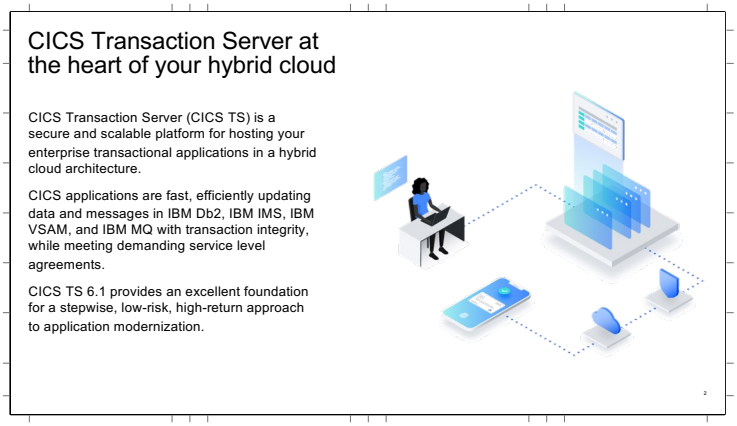


This is the Technical overview for CICS Transaction Server which was announced on April 5th 2022 and GA from June 17th 2022.



So lets start with a summary of how we position CICS within the enterprise.

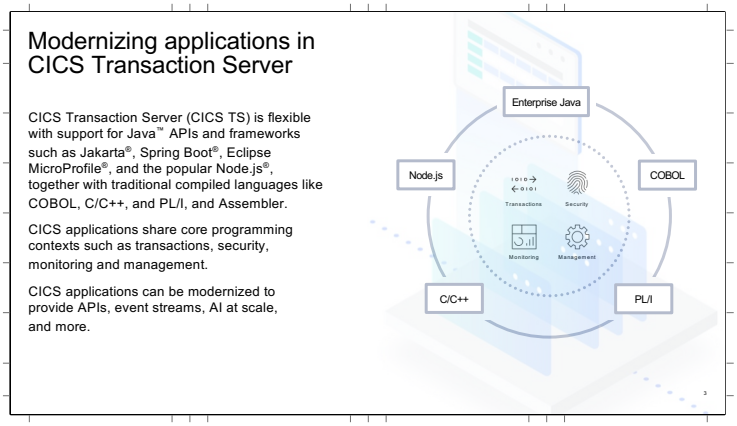
It is a world-class mixed language application server and has been running customer core services reliably for many years.

CICS harnesses the power of IBM Z, z/OS, and facilities like the Sysplex, to provide a secure and scalable platform for transactional workloads and allows integration in a hybrid cloud architecture.

CICS provides developers with programming interfaces to efficiently update data and messages whilst prioritizing large workloads to meet demanding service level agreements.

CICS TS 6.1, together with broader initiatives on the zSystems platform, provides an excellent foundation to modernize your applications in a stepwise manner that is low risk. It enables you and your teams to focus on meeting new business requirements and delighting customers.

So what does modernisng an application look like ?



CICS is very flexible and provides a choice of familiar languages and frameworks to integrate and modernize your applications and data.

This includes Java, Node.js, and the latest IBM Enterprise compliers for COBOL, C/C++, PL/I, and Assembler.

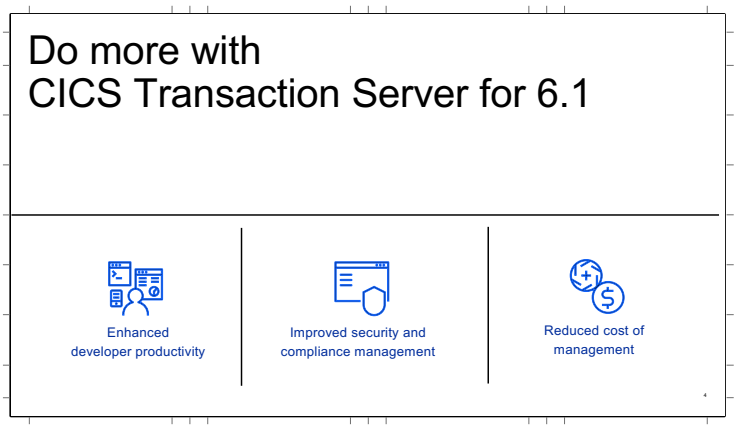
Each of these have been optimized to take advantage of new instructions on your IBM Z machine.

Because CICS applications share core programming contexts and can easily link between languages, this means developers are able to modernize part of an application and use languages, frameworks, IDEs, SCMs, tools, and pipelines that they are familiar with.

For example, a developer can replace a COBOL program with one written using Java Spring Boot framework that interacts with microservices running in the cloud.

CICS application can also be modernized with API enablement, interact with event streams like Kafka, and call new AI inferencing engine in the latest IBM z16 hardware.

So let’s now turn our focus to what is new in the latest CICS TS release.



CICS TS 6.1 was announced on 5th April 2022 and is available from 17th June 2022 onwards.

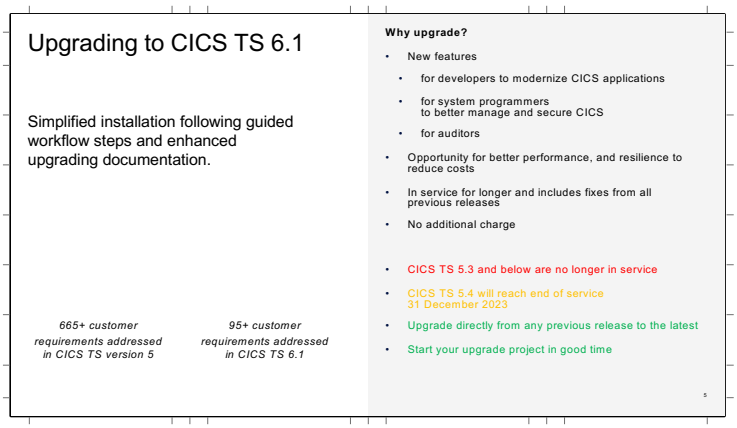
Its new features and interactions with other products are grouped under the following themes:

Firstly, enhanced developer productivity to provide the latest enterprise APIs to give you a solid foundation for Java components. It also includes a new tool to move your CICS resource definitions to be updated and managed alongside your source code.

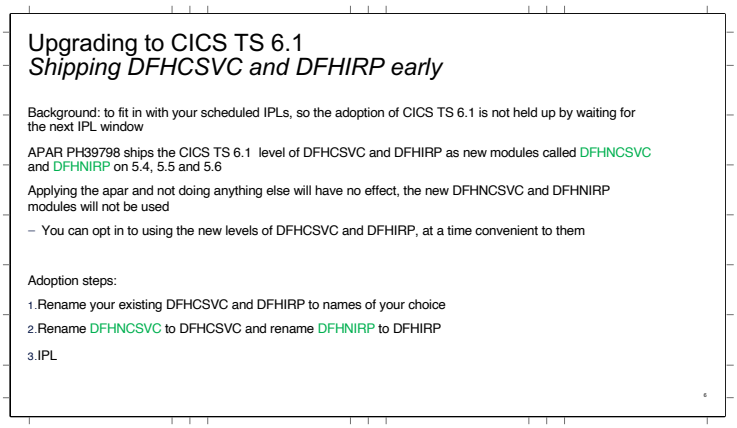
Second, Improved security and compliance management to strengthen network security, solve sysplex-wide security issues, and automate gathering evidence for compliance audits.

Thirdly, reduced cost of management. Lots of new Foundational capabilities including new policies, resource overrides, resilience items and monitoring and diagnostics to better manage CICS regions.

Let’s start with the first of these.



- New features for developers to modernize CICS applications – example using Java and its popular frameworks like Jakarta, Spring Boot, Eclipse Microprofile
- New features for system programmers to better manage and secure CICS – for example with CICS policies, TLS 1.3 and multi-factor authentication
- Opportunity for better performance, and resilience to reduce costs – for example with better statistics and diagnostics
- Upgrade directly from any release to the latest
- Includes service fixes from previous releases



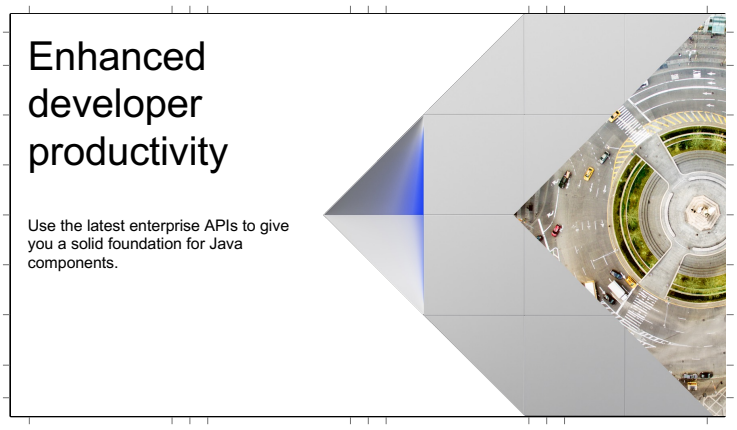
After apply the APAR to earlier CICS release, say 5.5, follow the adoption steps. Then in the end you are ready for running 6.1 when needed. So you can continue run with 5.5 library, or can run with 6.1 library.

Upgrading to CICS TS 6.1
Shipping DFHCSVC and DFHIRP early

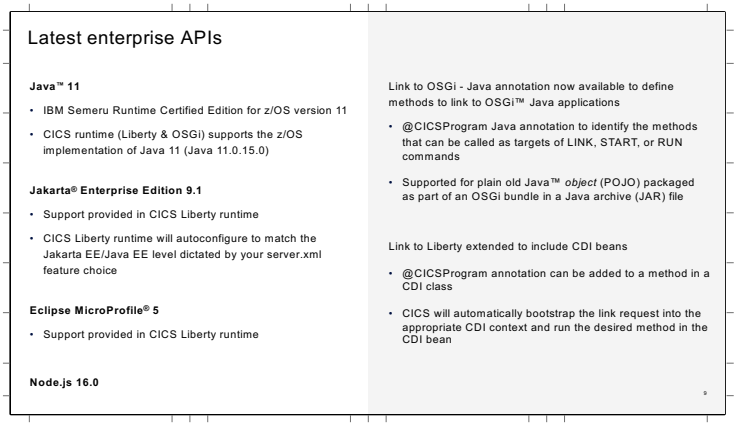
When CICS TS 6.1 is installed you can add the SDFHLPA library for the new release to the LPA concatenation as normal but an immediate IPL is not required

- The type3 SVC and IRP applied by this maintenance procedure is sufficient until the next scheduled IPL, at which time you can revert to using the GA-supplied modules
- The other CICS modules that are contained in SDFHLPA are not release specific and are upward and downward compatible, so earlier release modules operate correctly with the new GA release

- Note: In CICS TS 6.1, DFHNCSVC and DFHNIRP are shipped as dummy modules
  - They contain skeleton code, enough to assemble. They immediately return to their caller.
  - The modules will be replaced when the next release beyond CICS TS 6.1 ships its SVC and IRP code to lower releases as maintenance.



So for the first theme of Enhanced developer productivity we will first look at how we can enhances the experience and productivity of developers by adding support for familiar languages and APIs, and build tools.



In CICS TS 6.1, developers are now able to use the full breadth of features in Java 11, Jakarta 9.1, and Eclipse MicroProfile 5 to write CICS applications.

These versions are now open source projects with large communities and support behind them.

The CICS support for Java 11 requires IBM Semeru Runtime Certified Edition for z/OS version 11 that is available for free. IBM Semeru Runtime Certified Edition for z/OS, Version 11, formerly known as IBM 64-bit SDK for z/OS, Java Technology Edition, is certified with the Java Compatibility Kit as a fully compliant Java product. This rebranding aligns the naming for Java on the z/OS platform with that on other platforms where OpenJDK class libraries are similarly powered by the Eclipse OpenJ9 JVM technology.

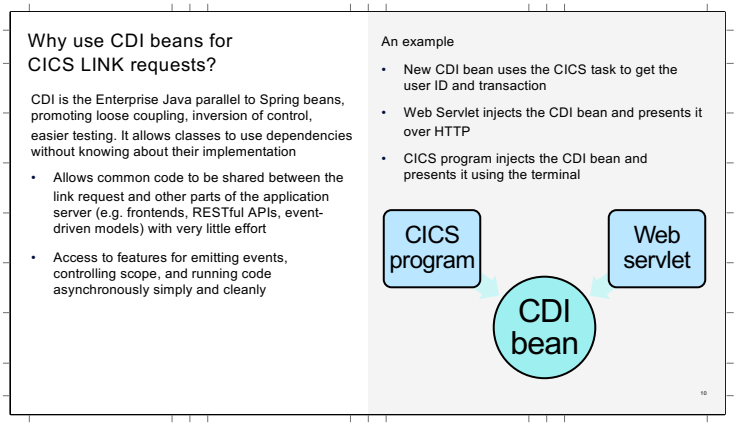
(Semeru, or Mount Semeru is an active volcano in East Java, Indonesia)

CICS support for Jakarta 9.1 and MicroProfile 5 are provided by Liberty that is included with the CICS TS installation.

For Java developers using the OSGi framework, CICS now supports **Java annotations** to expose a Java method to be called via an EXEC CICS LINK command as was previously available for Liberty applications. As the OSGi application is installed, CICS will automatically search for these annotations and automatically install a PROGRAM resource for it.

This makes it is easier to deploy and test these applications and move them to different JVM servers in the future.

In addition, the Link to Liberty support is further extended to allow linking to CDI beans, via similar annotation to the method in the CDI class.

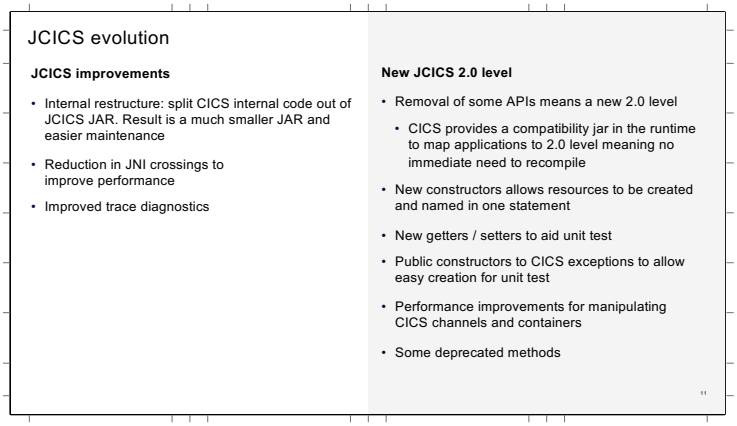


Context and Dependency Injection (CDI) provides a common mechanism to inject component such as Enterprise JavaBeans (EJBs) or managed beans into other components such as JavaServer Pages (JSPs) or other EJBs.

CDI support is provided by Liberty, and is configured in the Liberty server configuration files (server.xml and included files). CDI beans can be packaged in any of the following archive types: JARs, EJB JARs, or WARs. Any location on the application class path can contain CDI beans.

You can link to a Java™ EE, Spring Boot or CDI application that runs in a Liberty JVM server either as the initial program of a CICS® transaction, or by using the LINK, START, or START CHANNEL commands from any CICS program. To be linked to by a CICS program, a Java EE application needs to be a plain Java object (POJO) packaged in a Web ARchive (WAR) or Enterprise Application Archive (EAR).

To configure your Liberty JVM server to support linking to Java EE or Spring Boot applications, add the cicsts:link-1.0 feature to server.xml. To support linking to CDI programs, the cdi-1.2, cdi-2.0, or cdi-3.0 features must be added to server.xml (Linking to CDI v1.0 applications is not supported).



The JCICS API jar (**com.ibm.cics.server.jar**) has been re-architected to separate native code from the Java implementation resulting in the following benefits:

- Reduced chance that applications which (inadvertently) embed the JAR, get out of sync with the CICS runtime
- Reduced footprint and lower impact on development environments
- com.ibm.cics.server.jar shrunk from 873K to 448K
- Maintenance and bug fixing of **internals** no longer drives JAR versioning
- Less churn for repositories which publish the JCICS API (i.e. Maven Central)

As far as External changes are concerned:
Public constructors on exceptions

– So customers can unit test their code and throw mock exceptions if required

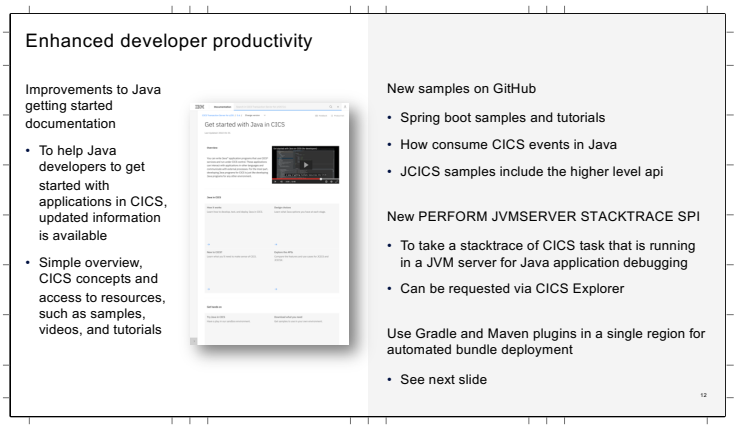
- Getter and setter methods provide better ways to do unit test

When providing a JCICS 2.0 level, there is a danger that existing OSGi based applications would be affected if they explicitly exclude version 2.0.0 of the ‘server package’ (JCICS) from the Import-Package header. The CICS bundle would fail to install, and the ‘unfulfilled’ dependency reported as an exception.
To provide compatibility we were able to define a new ‘proxy’ fragment we are able to attach to and enhance the original bundle by re-exporting the v2.0.0 API bundle as v1.900.0.

A fragment bundle is simply an extension of a host bundle. Using a fragment bundle means:

•JCICS bundle can be kept ‘pure’ and specific to the latest runtime API

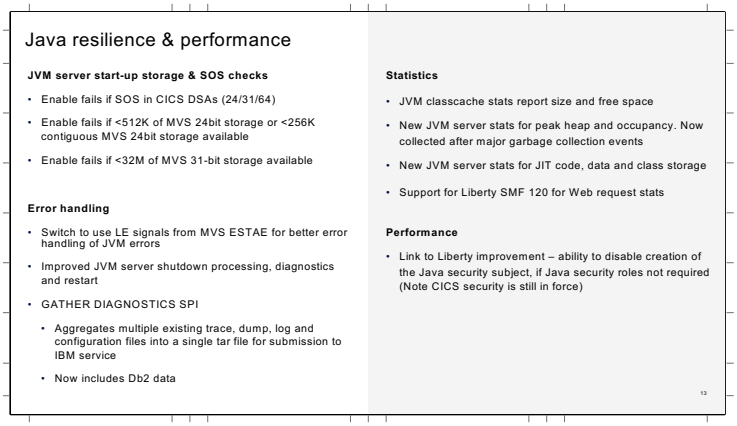
•customers cannot accidentally compile against the lower-level compatibility API because the compensating fragment is not distributed beyond our runtime



The CICS documentation has a dedicated section to help Java developers to get started and familiarise themselves with key CICS concepts with tutorials and videos.

On GitHub there are new samples and tutorials showing use of Spring Boot and how to consume CICS events in Java. The JCICS samples have been updated to include the new JCICS 2.0 level. Over time there will be more examples added to show use of the new JCICS 2.0 capabilities.

A new PERFORM JVMSERVER STACKTRACE command provides a method of generating a stack trace of the Java thread which is running a CICS task. The stack trace gives a report of the Java call stack at the current point in time during execution of the Java thread for the given CICS task. It can be used to understand in what method a Java application is suspended, and how that method was invoked.When the command is issued, the thread running the task will be located. A stack trace will be generated and written to the JVM log file (dfhjvmlog). The stacktrace can be activated via CICS Explorer, right clicking on the relevant JVMSERVER and selecting ‘stacktrace of running task’ option.



In the area of Resilience and performance for Java there are new capabilities in CICS TS 6.1.

Before starting a JVM server, CICS will check for short on storage conditions for CICS managed storage and will check on the amount of 24-bit and 31-bit storage MVS storage available (ie the amount of storage available in the region outside of the CICS managed DSAs.

CICS Java has two types of threads:

- CICS-enabled threads. Run on a CICS T8 TCB
- Plain Java threads, created and managed by the JVM

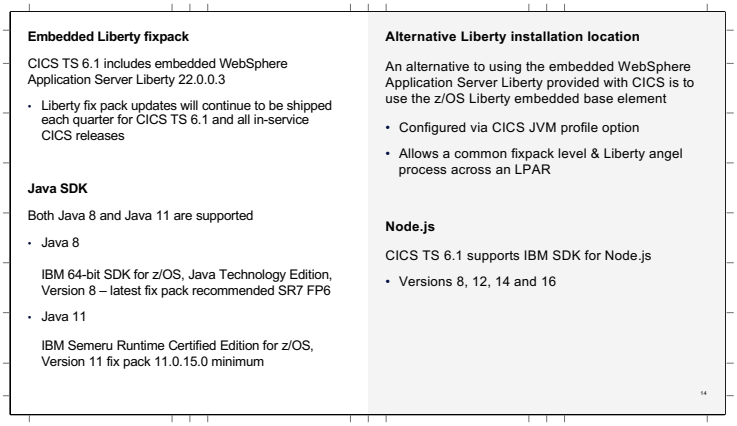
CICS-enabled threads stack the CICS ESTAE exit. Normal threads just have the LE ESTAE exit. This causes some differences in behaviour when errors occur. The implementation has changed to incorporate add a UNIX signal handler. This provides nicer handling of aborts in the Java VM, better diagnostics for CICS-enabled threads and additional diagnostics for common signals.

The GATHER DIAGNOSTICS SPI introduced in CICS TS 5.6 has been enhanced in CICS TS 6.1 to include collection of DB2 trace data when an application is using the DB2 Universal JDBC driver.

Java 8 Service Refresh 5 changed the JVM heap from a single consolidated heap to four smaller heaps. This meant there were misleading stats if there was more than a single java heap. The solution was to UtiliseGarbageCollectorMXBeans to track peak heap and occupancy after MAJOR garbage collection only.
In addition we also utilize MemoryPoolMXBeans to extract the data on the allocated and used memory for: JIT code cache, JIT data cache and Class storage.

Liberty produces SMF 120 subtype 11 (when HTTP requests are made to Liberty) and subtype 12 (Java batch). Changes to CICS code were required in order to drive the Liberty code, which produces the SMF records.

Link-to-Liberty is initiated from a CICS transaction where CICS security applies. By default the JCA adapter and WorkManager that process the CICS L2L requests create a Java subject with security credentials based on the CICS userid used from the parent task. This processing has been shown to be expensive in Liberty.
Not all customers will want or care about a Java Subject bring created. A JVM system property has been added instructing CICS to bypass the Java Subject creation (and Liberty registry checking) overheads.



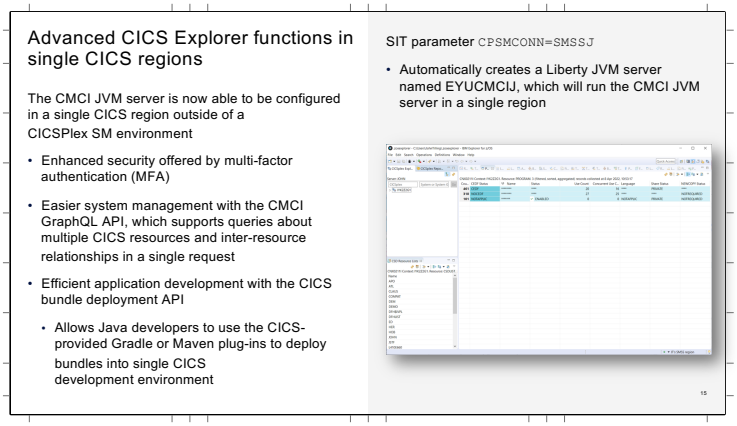
The slide summaries the level of support provided for Java and Node.

For Node we now support Node 16.

For Java 8 we recommend staying current with the latest fixpack.

CICS now supports the recently released Java 11 capability on z/OS via a minimum of fixpack 11.0.15.0. Expect further updates in CICS and further fixpacks from z/OS as the Java 11 capability matures.

For Liberty we continue to rollout support for the quarterly fixpacks on all inservice CICS releases. CICS TS 6.1 is shipped with support for Liberty 22.0.0.3 and that will roll forward via maintenance. As an alternative, you can choose to specify a new WLP_INSTALL_DIR setting in your JVM profile to use an alternative version of Liberty - one that is not supplied with CICS. This is for customers who want to have a common fixpack level and Liberty angel process across all Liberty users on an LPAR.



We understand that some CICS developers are using regions that are not configured to be part of a CICSplex.

This means developers are missing out on the full capabilities of CICS Explorer, including bundle deployment, multi-factor authentication, and aggregation.

However, with CICS TS 6.1 you can now easily setup the CMCI JVM server in a single region with a few changes to the CICS startup configuration.

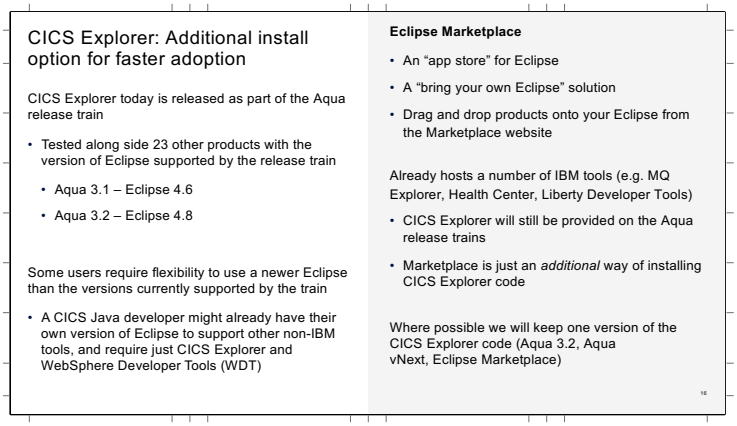
You can then connect to the region using CICS Explorer and use these advanced features.

The CICS® management client interface (CMCI) supports deploying CICS bundles into a region through the CICS bundle deployment API.

- Java developers can redeploy a bundle into a CICS region within seconds, without the need for a zFS connection or to disable, discard, and reinstall the bundle manually.
- Java developers can integrate CICS bundle build and deployment into their toolchain, saving lots of manual work.
- The API ensures controlled access both to the CICS system definition data set (CSD) for BUNDLE definition installation and to the bundle directory on zFS, so that system programmers can allow Java developers to deploy bundles without granting additional access.

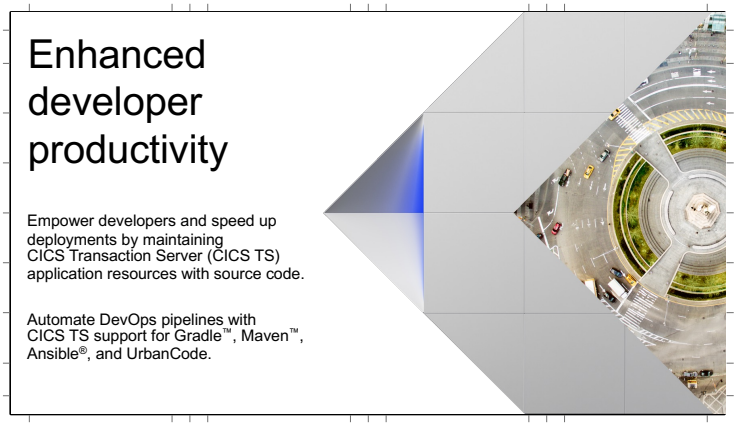
The CICS bundle deployment API enables Java developers to deploy bundles whilst the system programmer retains control. This is achieved by removing the need for developers to write bundles to zFS through FTP, or to install bundles from CSD. These actions are taken by the API, by using a functional ID or another user ID with sufficient access.

CICS provides a Gradle and a Maven plug-in that uses this API to publish bundles to CICS. Java developers can use it for bundle deployment at development time.



IBM CICS Explorer is released as part of the Aqua release train which is tied to a specific Eclipse release. It is tested alongside 23 other products.

An additional release channel for CICS Explorer is being provided, namely Eclipse Marketplace. It can be though of as a “app store” for Eclipse. It allows a drag and drop into an existing Eclipse. It allows use of CICS Explorer in Eclipse environments different from those currently supported by the Aqua release train. It is already used by other IBM tools.



We have also been working on a solution to enable developers to speed up application deployments by maintaining CICS application resources with their source code.

With CICS TS 6.1, we are introducing the new CICS TS resource builder. It is a tool to facilitate configuration-as-code for application resources.

A few years ago we introduced tools to help you automate DevOps pipelines with CICS TS and these now all support CICS TS 6.1.

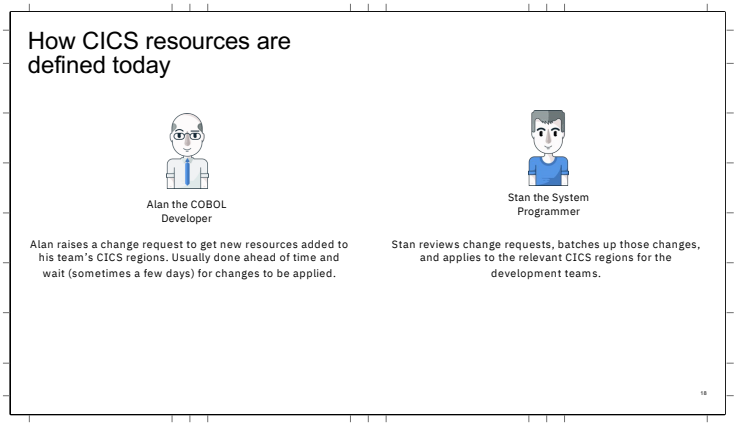
This includes plugins for Gradle, Maven and UrbanCode Deploy.

And, last year we extended this list to include Ansible.

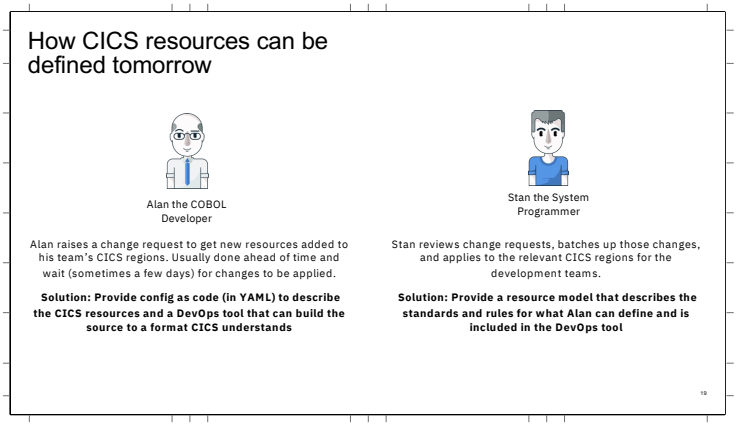
DETAILS:

Announcement - https://www.ibm.com/common/ssi/cgi-bin/ssialias?infotype=an&subtype=ca&appname=gpateam&supplier=897&letternum=ENUS222-092

What's new? - https://www.ibm.com/docs/en/cics-ts/6.1?topic=whats-new

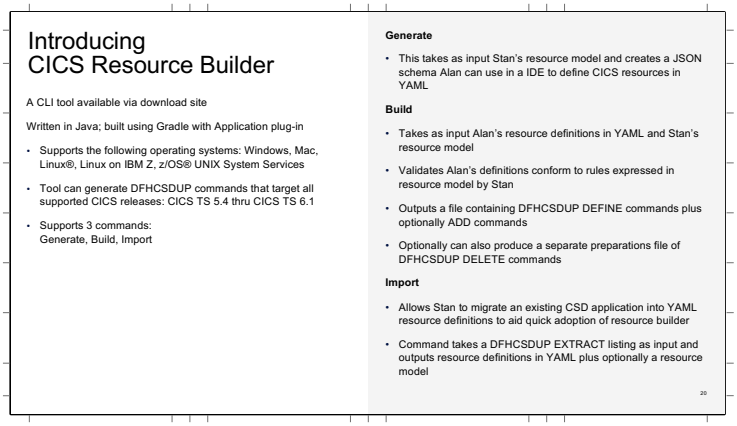


Shows a typical process that happens today in CICS shops when new CICS resources are required.



Shows a solution to these problems, by using configuration as code.

YAML, is a human-readable data-serialization language. It is commonly used for configuration files elsewhere.

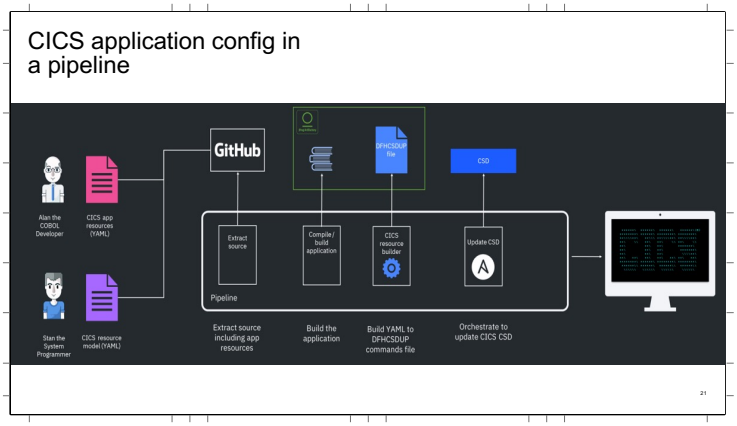


With CICS TS 6.1, we are introducing the new CICS resource builder. It is a tool to facilitate configuration-as-code for application resources.

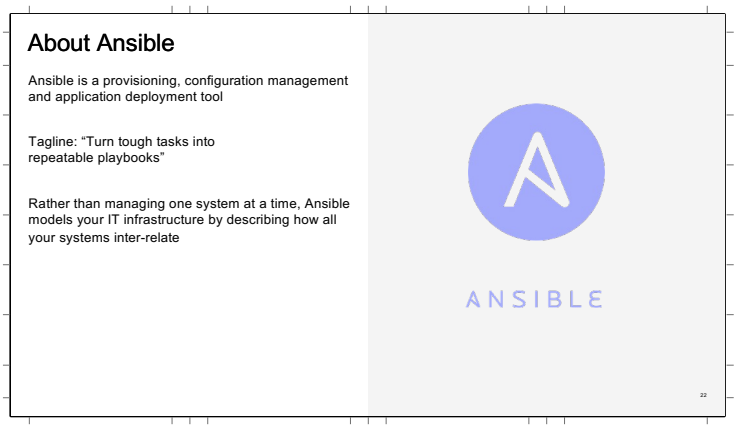
It enables developers to manage resource definitions in a readable source format using a modern source control management (SCM) and modern pipeline.

It is designed to run alongside the application build and provides output to be used in later steps in the pipeline to deploy resources alongside the application.

Developers are now able to change application code and resources together in a single pull request, giving a consistent and efficient approval and audit process, enabling applications to be deployed with confidence in minutes rather than hours. However it still provides control to the system programmer to enforce standards and naming conventions.



The diagram shows how the resource builder can be incorporated into a CD/CI pipeline



Ansible is extremely popular across many enterprises (non-Z) but is now starting to become popular on Z as well. It’s a general purpose automation tool.

Common uses of Ansible include:

- System provisioning
- Installing applications
- Managing users
- Updating certificates
- Continuous delivery
- Configuration management

You can also use to automate very small things, eg individual command on a single machine

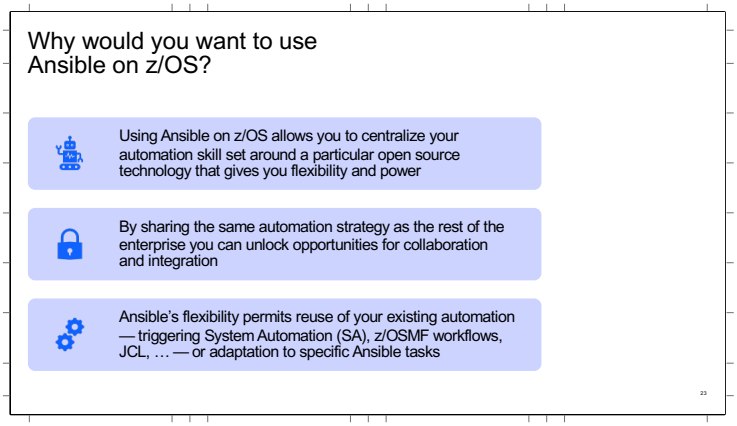
It’s now the top cloud configuration tool and is heavily used on-prem too.

Some of the reasons for that include:

- It normalizes tooling across a multitude of platforms
- It centralizes your enterprise automation strategy
- You can achieve configuration as code

It has a big eco-system, extensions to Ansible are called modules.

There are over 3000 modules for all the things you might need it to do.

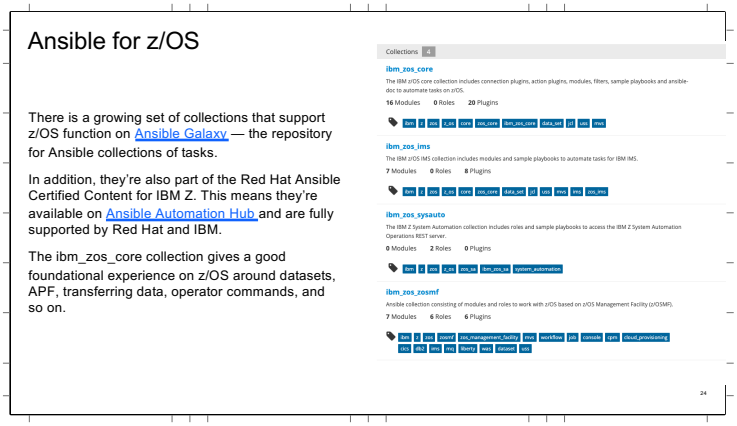


So why would you want to use it on Z ?

Being able to the skillset for automation on other platforms and use it on Z.

Build more sophisticated solutions that automate changes to multiple environments, deploy back end on Z, front end on a distributed platform and make it all part of one process.

Because its general purpose, you can reuse all the existing automation assets you have.



On the right hand side you can see a screen shot of Ansible Galaxy which is the registry of repositories where you find Ansible collections of tasks.

There is a growing set of collections for z/os environments. The z/OS core collection is the most extensive at this point in time.

Recently added (February 2022) to that list is a collection for CICS.

Galaxy is the registry of collections available as open source (use on your own terms) typically with no service, other than that provided by the open source community.

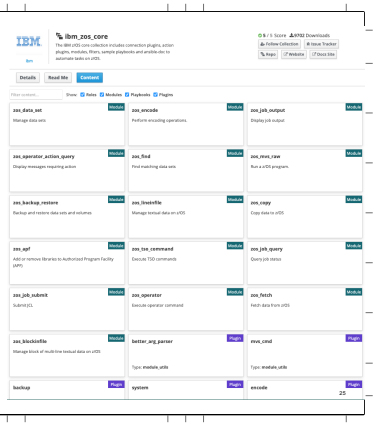
There is also a paid subscription service supported by Red Hat and IBM called Ansible Automation Hub and our collection is published in there as well.

The z/OS core collection

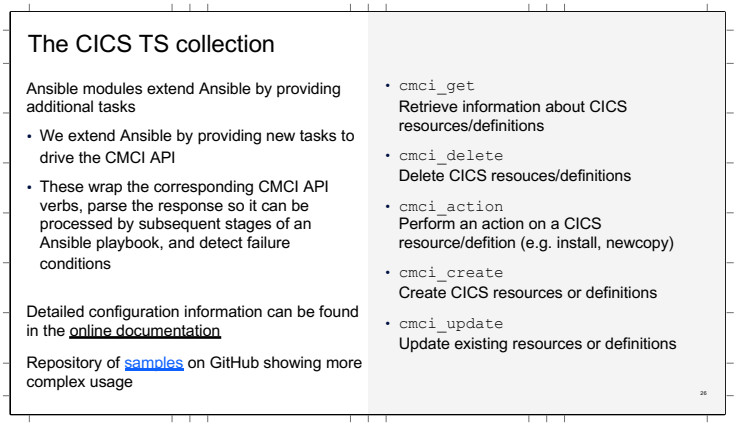
The ibm_zos_core collection provides easy to use z/OS building blocks necessary for performing basic operations in z/OS including:

- Job related tasks
- Data set tasks
- Issuing commands

Combining these modules together in Ansible playbooks enables an infinite number of use cases to be automated against



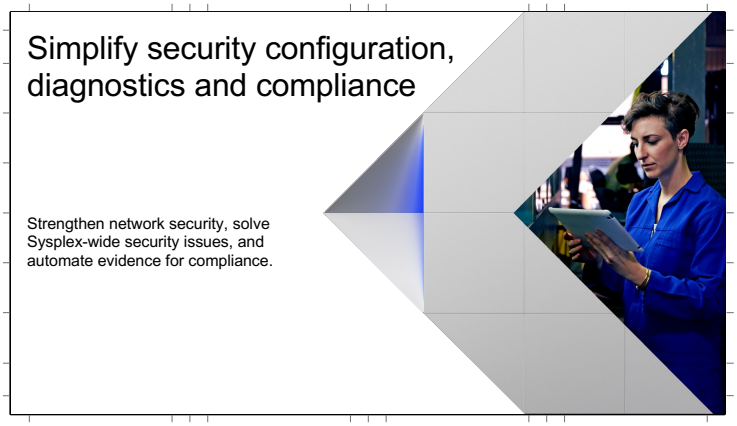
The z/OS core collection has a lot ansible tasks that give you building blocks to build on, interacting with z/os datasets, submitting jobs, issuing commands etc



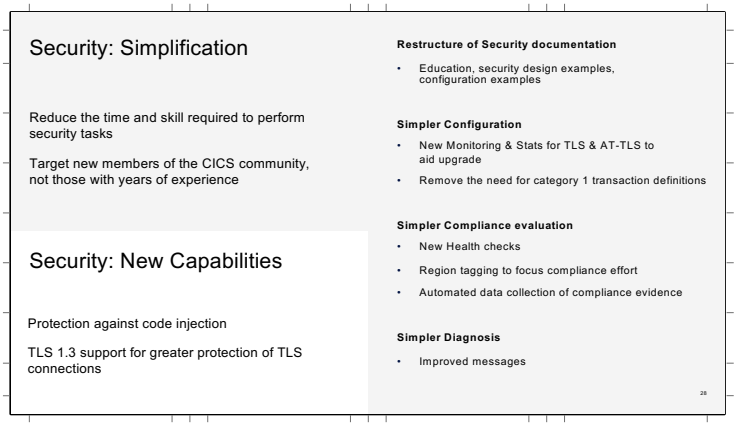
In the new CICS collection, we have published 5 ansible modules that wrap the CMCI API verbs and integrate the response into your playbook.

So it lets you do anything you can do with the CMCI api, so for example query resources in a CICS region, updating resources, or definitions in CSD or CPSM BAS.

In GitHub we have published examples of using these tasks. For example there is a simple sample of querying information about CICS regions and then writing the output to a csv file. You can do this today manually via CICS Explorer, but this provides an automated, repeatable way to getting that information into a speadsheet.



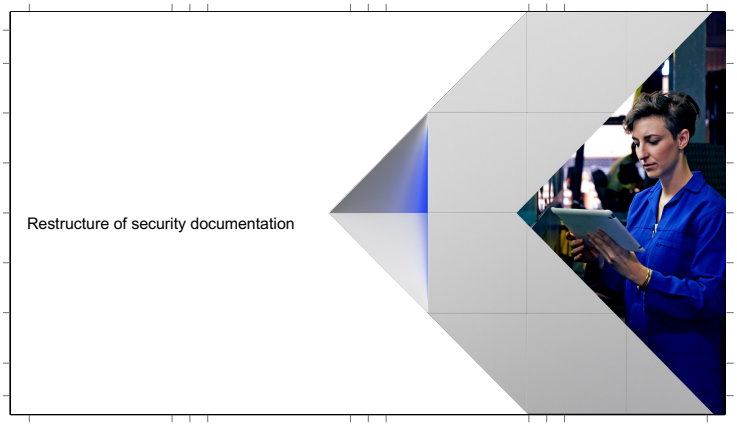
CICS TS 6.1 provides improved security and now gathers evidence for compliance. This includes support for TLS 1.3 and new documentation and diagrams based on scenarios to help secure systems and applications. CICS TS now collects configuration evidence required by compliance authorities and writes it to SMF. This evidence is collected by IBM Z Security and Compliance Center to provide detailed reports to enable executives, administrators, and auditors to understand compliance metrics with ease, and to track compliance drift over time using dashboard visualizations.



The big aim of the Security focus in CICS TS 6.1 is simplification. We want to reduce the time and skill required to perform security tasks. This is aimed at our new members of the CICS community, not those with 20 or 30 years worth of experience.

We are going to look at simplification in various guises, firstly through simplified and restructured documentation. Next we will look at changes providing simpler configuration when upgrading. After that we will look at simpler ways to ensure compliance and collect evidence of compliance. Last but not least we will look at simpler diagnosis through greatly improved security messages.

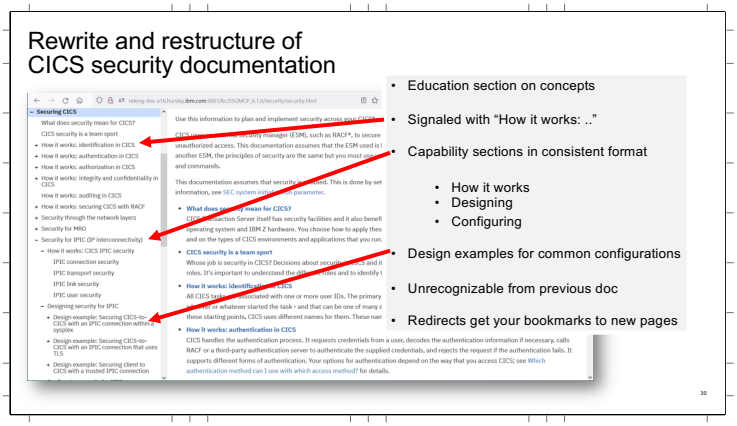
Alongside simplification, there are new Security capabilities introduced in CICS TS 6.1, so we will look at protection against code injection and support for the latest TLS level, TLS 1.3



The restructure of the security documentation has been one of main items for the Security enhancements made in CICS TS 6.1.

There is an increase in younger system programmers, as the older generation retire. The 500 pages of CICS security documentation has been built up over the last 50 years and whilst is a good reference for those who have grown up with it, it is not easily consumed by the new generation. The language of computing and security has changed, making it difficult for new joiners to understand some of the terminology used.

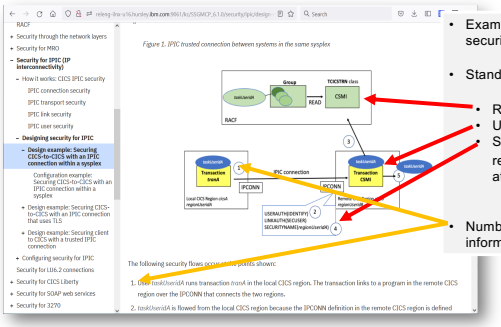
The restructured documentation includes education and has a scenario based structure with clear advice on best practice and recommendations.



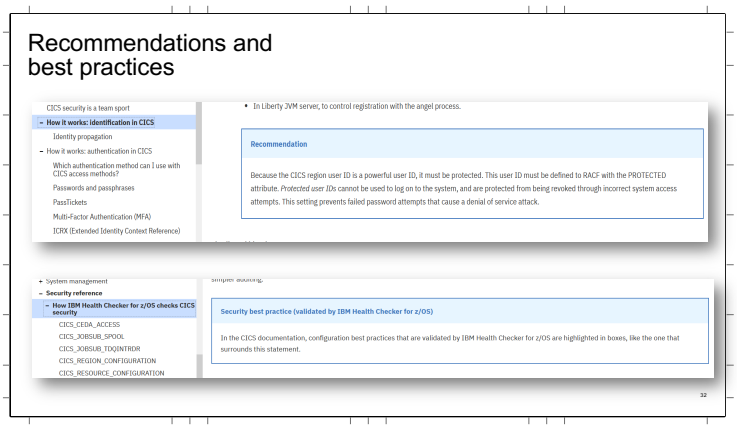
The new doc has an education section, with a series of how it works topics to describe the various concepts of security, such as identification, authentication, authorisation etc, and how they apply to CICS.

These are followed by a series of sections relating to capabilities within CICS, for example SOAP, IPIC, Liberty and so on. For each of these capabilities , first there is how it works topic giving an overview of the security concepts for that capability. Next comes one or examples on how to design security for that scenario (useful to application architects) and finally the configuration options that come into play with task based workflow for system programmers.

Design example diagrams



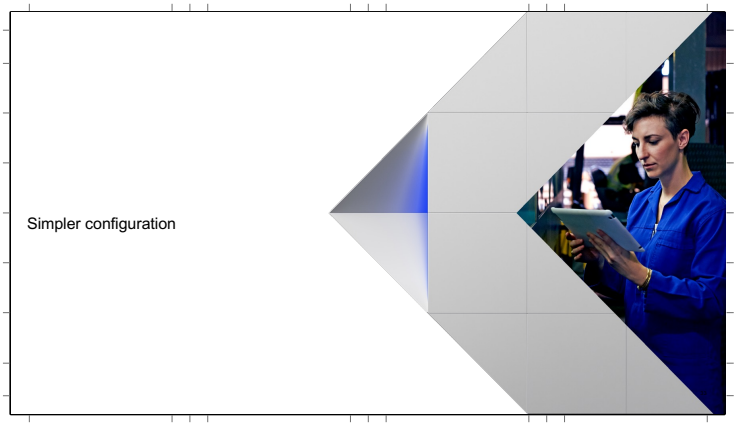
- Examples with diagrams of security information
- Standard diagrams with
  - RACF profile and permissions
  - UserIDs in flow
  - System management resources and security attributes
- Numbered flow of security information

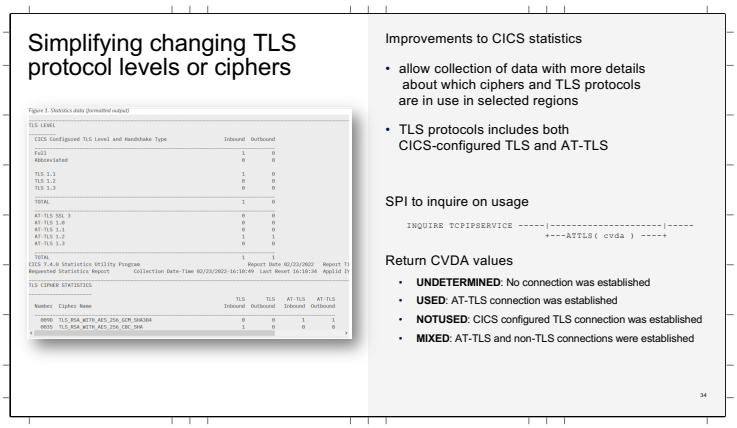


In various places in the documentation you will find recommendations and best practices based on our experience of reviewing customer installations. These will both give advice where there are choices and, importantly, explain why the advice is given.

The difference between best practice and recommendation is that best practice is also backed up by a check in the health checker that the advice is being followed.

More information on the health checker is coming in later slides.


Simpler configuration



If a TLS protocol is compromised, you need to switch to a higher minimum TLS level.

If a cipher is compromised, you need to remove the cipher from all TLS connections.

To safely make a change, you need to identify if the compromised protocol or ciphers is in use. When you know it is in use, you need to identify who is using it so that the client or server that CICS is communicating with can also be upgraded.

CICS statistics and monitoring have been enhanced to help evaluate the TLS protocol levels and ciphers in use and identify the impact of a change to you and your users. o

TLS protocols includes both CICS-configured TLS and AT-TLS.

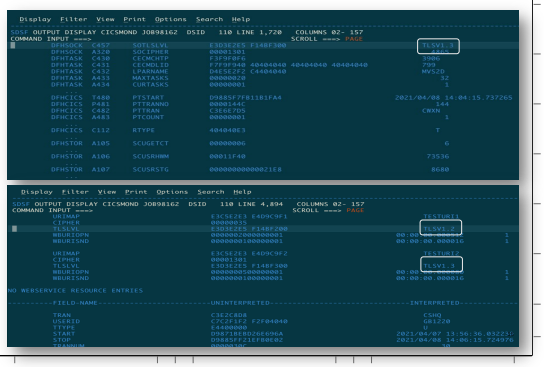
A new ATTLS keyword on the inquire TCPIPSERVICE command.

New cipher resource statistics provide details of the ciphers that are used in inbound and outbound connections. TLS protocol level usages are now available for review when you use the TCP/IP global statistics.

Simplifying changing TLS protocol levels or ciphers

Improvements to CICS & monitoring

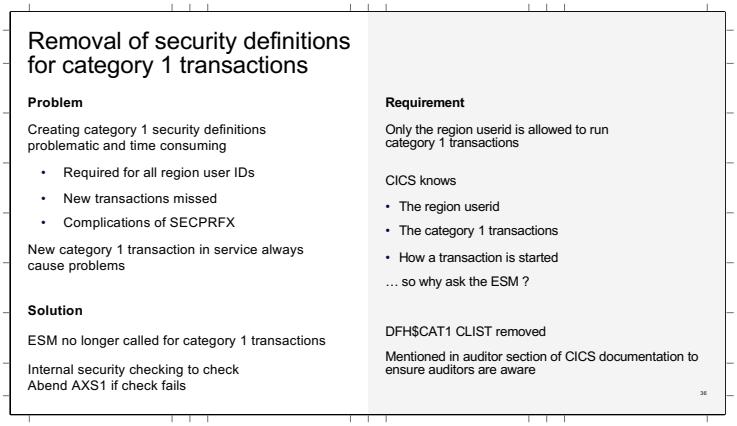
- For inbound connections, the TLS level is reported in the performance record
- Transaction resource class monitoring records report TLS level for outbound connections



Information for inbound connections regarding the TLS level is available in the Performance class records.

For outbound connections, the TLS level is reported in the transaction resource class monitoring records.

The slide shows examples from a DFH£MOLS report showing the TLS level reported.

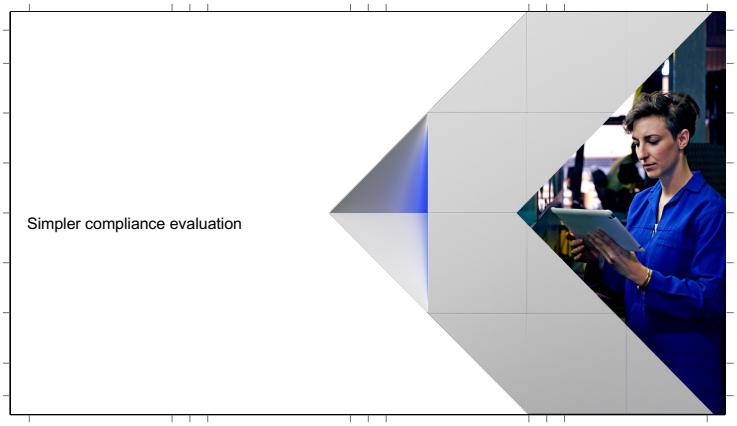


Category 1 transactions are for CICS internal use only and must not be started from a user terminal. Because these transactions are part of CICS itself, they run under the CICS region user ID.

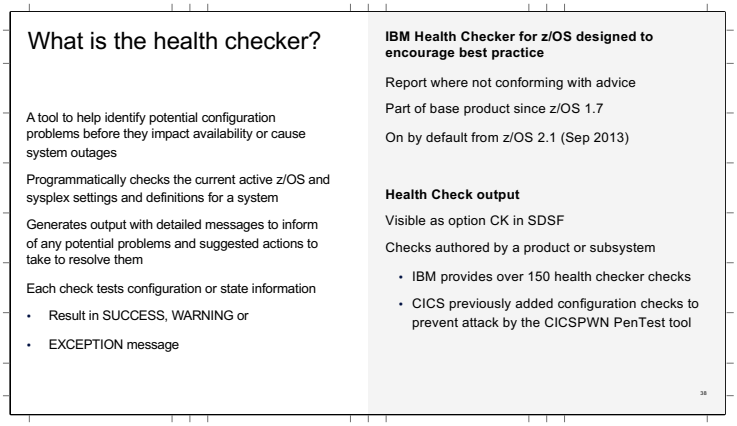
Previously when starting a CICS TS Category 1 transaction, a call to RACF® validated that the configuration was correct.

RACF is no longer checked when starting a CICS Category 1 transaction. The DFH$CAT1 CLIST has been removed. As CICS knows the region userid, knows which transactions are Category 1, knows how the transaction was started, it can make the check itself without involving an ESM.

This change improves security as only CICS determines that a Category 1 transaction can run. This change also simplifies configuration and upgrades because there is no need to define the Category 1 transactions to RACF, which might create misconfiguration. You will need to define the CICS region user ID to RACF to confirm the ID that is used for running CICS Category 1 transactions. Surrogacy definition is still required as documented in Surrogate security.



So lets move on and talk about compliance evaluation. The first thing to talk about is the Health Checker.



So first a quick introduction to the Health checker for those who are not familiar with it.

IBM Health Checker for z/OS (Health Checker) is a z/OS component that helps simplify and automate the identification of potential configuration problems before they impact availability or cause outages. As well as messages there is a simple to use interface to use in SDSF

CICS TS supports Health Checker rules that define preferred practices for CICS system configuration. CICS TS introduced rules In CICS TS V5 to check for security misconfiguration and this has been expanded upon in CICS TS 6.1

Each CICS region providing support for Health Checker executes the system transaction CHCK as a long-running task. This task wakes up every 30 minutes to check and report on compliance to preferred practices. Our performance team previously measured the impact of having the Health checker active with CICS TS 5.4 and concluded it has zero impact, ie consuming 1ms of cpu per hour for an idle region.

New health checks

Based on best practice reviews of customers

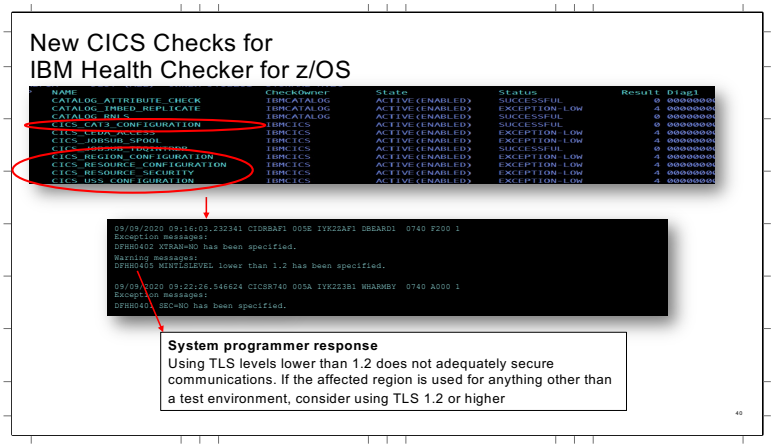
Covers security configuration of

- Region's definitions
- CICS resources
- CICS zFS security

Best practice advice is aimed at production or production-like regions

Examples of checks

- `SEC=YES`
- `XTRAN=YES|class`
- `XUSER=YES`
- Default user can access sensitive transactions
- Universal USSCONFIG access
- Universal JVMPROFILE access



There are five new sets of checks in CICS TS 6.1, adding to the set of three checks added in CICS TS V5.

CICS_CAT3_CONFIGURATION checks that configuration of the CICS® category 3 transactions are as expected.

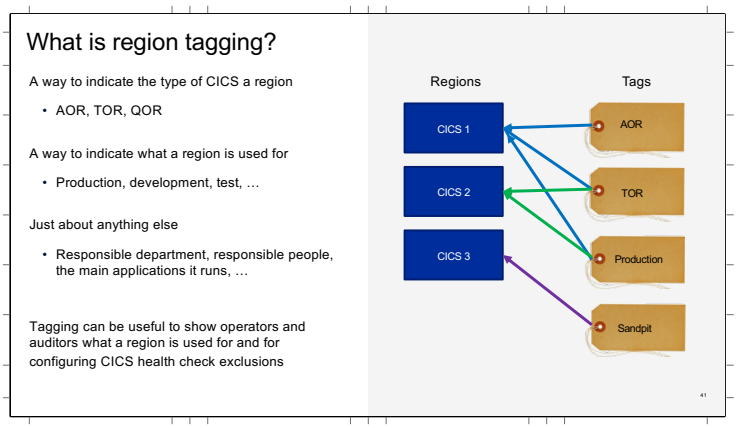
CICS_REGION_CONFIGURATION checks that key production region configuration parameters align to best practice recommendations

CICS_RESOURCE_CONFIGURATION checks that configuration of the CICS resources aligns to best practice recommendations for production regions.

CICS_RESOURCE_SECURITY checks a range of SIT parameters that align to best practice recommendations for production regions.

CICS_USS_CONFIGURATION checks that access to key USS (UNIX System Services) files and directories align to best practice recommendations for production regions.

When to select a check you get more detailed information giving a report of all regions that produced messages. All messages come with advice on how to fix the exception in the IBM documentation.



CICS region tagging offers you the ability to classify regions based on the assigned APPLID, region user ID, and job name for the region. You can define region usage based on naming conventions and pattern matching elements of the assigned APPLID, region user ID, and job name. You can then use these definitions to aid with auditing, running CICS Health Checks, or generally providing categorization information to operators. Region tagging is optional for use in CICS.

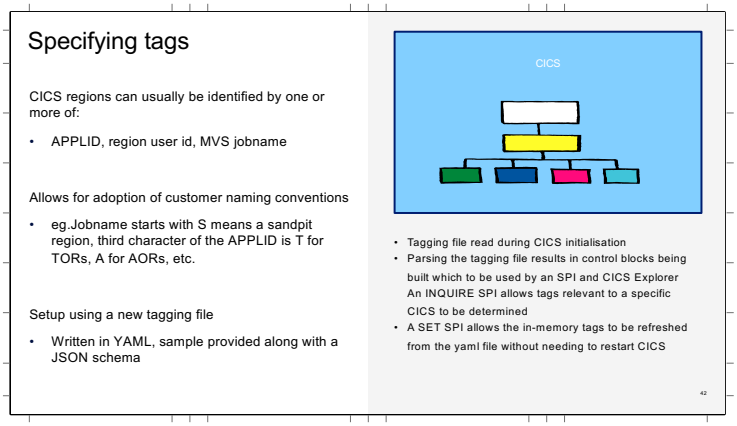
Why use region tagging?

You have three main reasons to consider as to why you would use region tagging.

- Display information for operators.
  Operators or other users with valid access can use INQUIRE TAG to identify the region tags in use, which you configure based on your existing naming conventions.

- Auditing
  Auditors can more easily determine the region usage.

- Configuring CICS health check exclusions
  You can define in the region tagging file: Which regions are excluded from running any health checks. For example, you might exclude regions that are defined as development or sandpit.
  Which specific health checks are excluded from being run. For example, you might want to disable the health check DFHH0701, which detects whether XPCT=NO is specified in your SIT parameters.



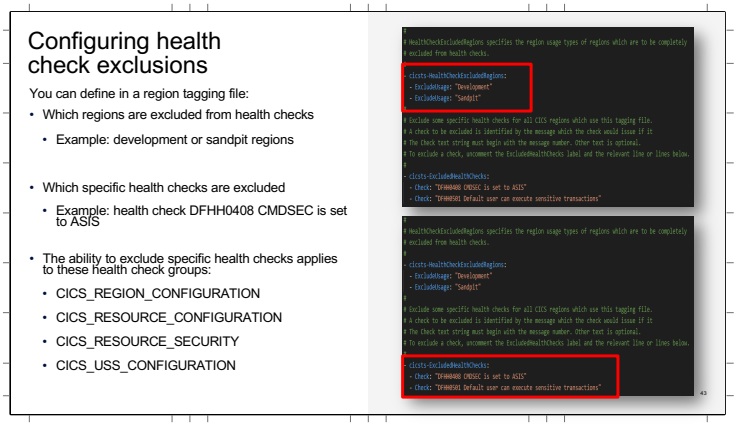
A region tagging file is defined in YAML, which is a human-readable data-serialization language. It is commonly used for configuration files elsewhere. Its defining a set a key-value pairs. The region tagging file is formatted according to the YAML 1.2 standard. YAML files must conform to specific formatting to ensure correct processing.

To aid users in formatting the region tagging YAML file correctly, a JSON schema is provided. In many schema-aware editors, editing a file named cicstags.yaml will automatically use the correct schema by downloading it from SchemaStore. Alternatively, you can manually apply a schema to the relevant files. The schema provided with CICS is called cicstags-1.0.0.json and is located in USSHOME/schemas.

The CICS region tagging file must be called cicstags.yaml and must be saved in the root directory of USSCONFIG. A sample version of the file is located in USSHOME/samples/cicstags.

CICS defines 5 tag keys and users can add their own, as shown in the example cicstags.yaml.

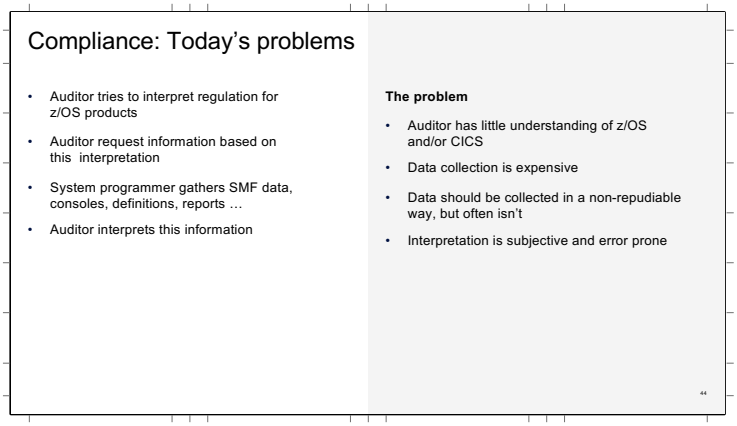
- The region type is defined by the tag key cicsts-RegionTypes. For example, TOR or AOR.
- The region usage is defined by the tag key cicsts-RegionUsage. For example, Production, Development, or Sandpit.
- The applications that are used in a CICS region are defined by the tag key cicsts-Applications. For example, Mortgage, Banking, or Travel.
- The usage types of regions that are excluded from CICS Health Checks are defined by the tag cicsts-HealthCheckExcludedRegions. For example, Production, Development, or Sandpit.
- Specific CICS Health Checks that are to be excluded are defined by the tag keycicsts-ExcludedHealthChecks. For example, DFHH0402 or DFHH0409.



You can use tagging to exclude types of regions from health checks, for example development or sandpit regions.

You can also exclude specific checks from all regions.

The screenshots on the right hand side show the sample YAML file and how you set up such exclusions.

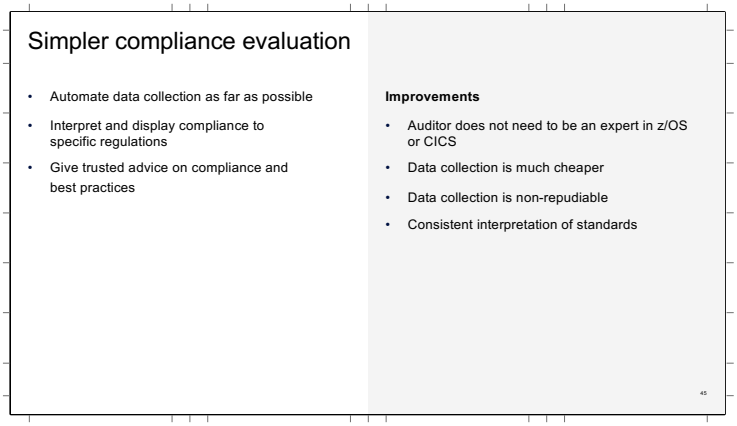


Let us now look at how CICS is audited today for compliance with industry and government regulations.

The situation at the moment is that auditors have a hard time trying to audit CICS. They have various regulations that are generally written for non Z platforms and they have to try and interpret these regulations.

Based on their interpretation the auditor then requests information to be gathered, which can be quite a time consuming procedure. Many customers have multiple people, whose sole job it is to gather this information. The auditor then has to interpret the information collected to come to a conclusion as to whether or not the system is in compliance with the required regulation.

The problem the auditors have is their knowledge of Z and especially CICS is limited, apart from a few specialist auditors. In addition the data collection is expensive because many customers have 10s, 100s or even 1000s of CICS regions. Collecting that amount of data is very difficult, especially as it needs to be collected in a way that can't be tampered with. Lastly, there is a problem that an auditor's interpretation can be subjective, and sometimes can be error prone.



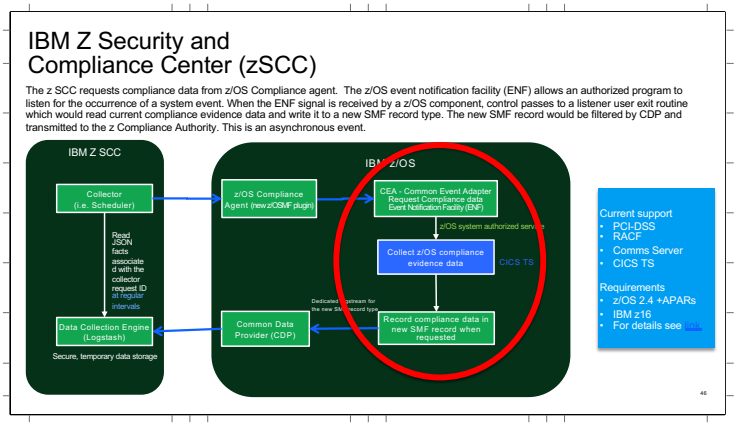
In CICS TS 6.1 we addressed this problem by coming up with a solution that automates the data collection as far as is possible. In addition, with the correct tooling in place, we have something that can interpret and display compliance with particular regulations in a simple way to an auditor to say whether a customer passes or fails.

For this first release of the solution we have concentrated on the PCI DSS (Payment Card Industry Data Security Standard), but where this overlaps with the DISA STIGs Defense Information Systems Agency (DISA) Security Technical Implementation Guides (STIGs) we also support those.

Its not just about a pass or a fail, the solution can give advice on what needs to be done to comply.

The solution means that the auditor does not need to be so much of an expert on the z/OS or CICS product.

Other advantages is we've made collection of the data much cheaper because it doesn't require collection of data from multiple regions and secondly the data is written in a way that you can't tamper with it.

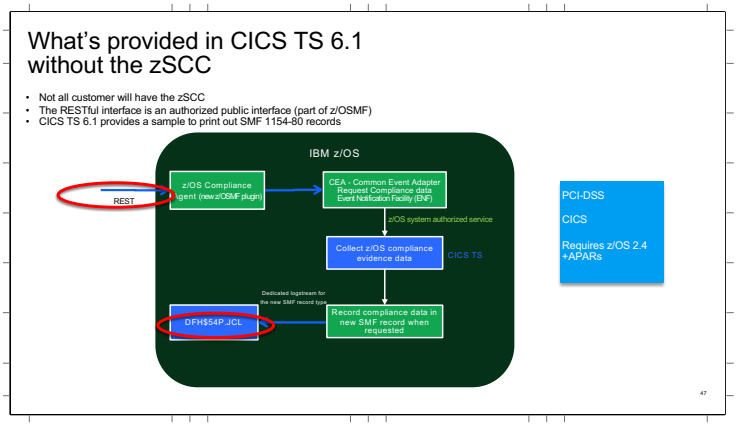


The CICS solution is part of a wider solution released in conjunction with the new Z16 mainframe.

There is a new product called the Z Security and Compliance Center (zSCC). This is tooling which will allow auditors to request information from many products running on Z systems. That data is then collected automatically and interpreted by the zSCC. Results are displayed on a dashboard to indicate compliance or not via a simple traffic light system which you can drill down into.

Where this affects CICS is the zSCC issues a Restful service to an agent running on z/OS which is part of z/OSMF. This agent issues an ENF event and distributed to all the systems that are listening for it, one of those being CICS. CICS will then go and collect data and write it out to an SMF record. The SMF record is then collected by a new component called the common data provider, interpreted in a common way and the results sent in JSON format to the zSCC which interprets it and displays it.

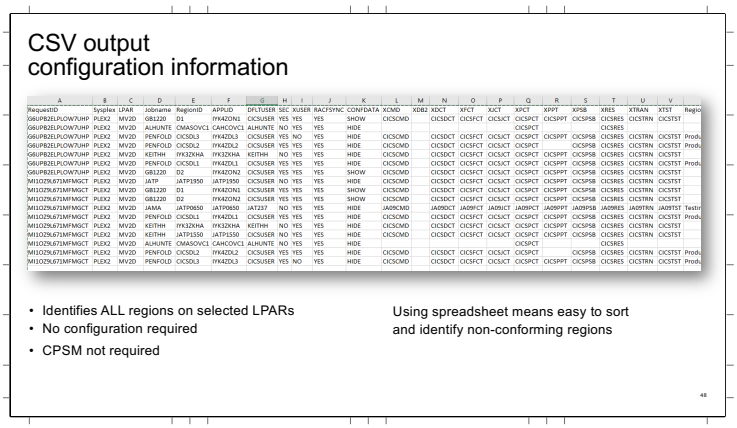
The solution which z/OS 2.4 or higher with maintenance and it also requires a Z16.



zSCC is not part of the base operating system. We realise that not all customers will purchase the zSCC. For those that don't, they can still use the functionality in CICS by means of a RESTful interface which is published as part of zOSMF.

This will do the same as before, ie generate an ENF event which is broadcast to all CICS regions. The CICS regions will collect the data and write an SMF record. When all the SMF records have been written, that information can be displayed by a sample program that CICS provides.

This solution requires z/OS 2.4 or higher with the necessary z/OS maintenance applied which provides the new ENF support.

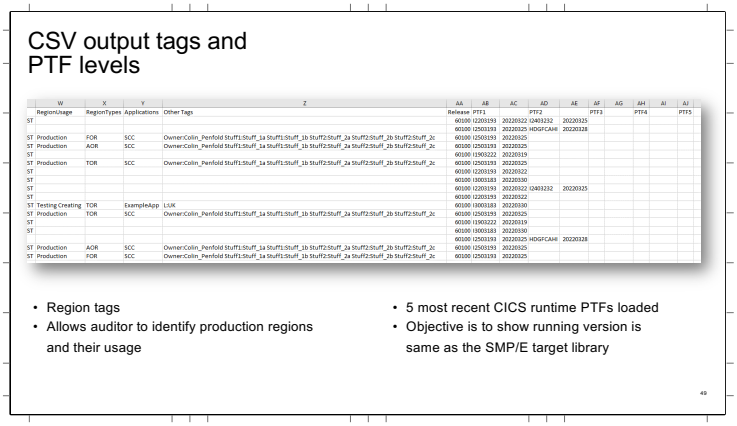


CICS provides sample JCL and a sample REXX program to read the SMF data and output in various formats, one being a CSV file for easy display in a spreadsheet.

The slide shows an example of the output CSV file. The data is shown in a nice easy to read form and you can see whether regions are compliant using sort as appropriate.

For each region you see the basic information such as jobname, applid and job userid, followed by the SIT configuration. On the right hand side you've got information about the RACF classes that are used, or not used by that region.

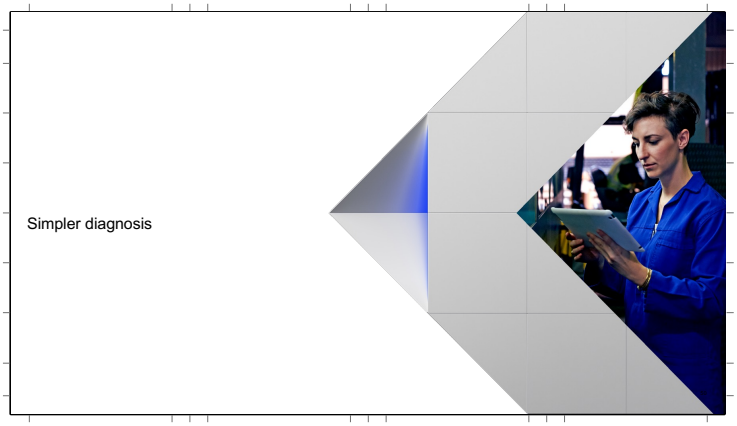
Note: The information is collected independent of whether you are using CPSM.

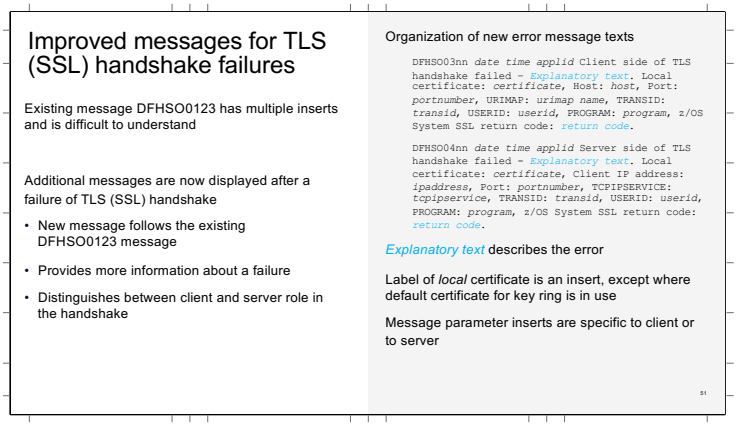


On the right hand side of the spreadsheet is information regarding Regions tags, both our prefdefined tags, and any user tags defined.

So for example whether or not a region is a production region, or development or test. The data can then be sorted for an auditor to concentrate on production regions for example.

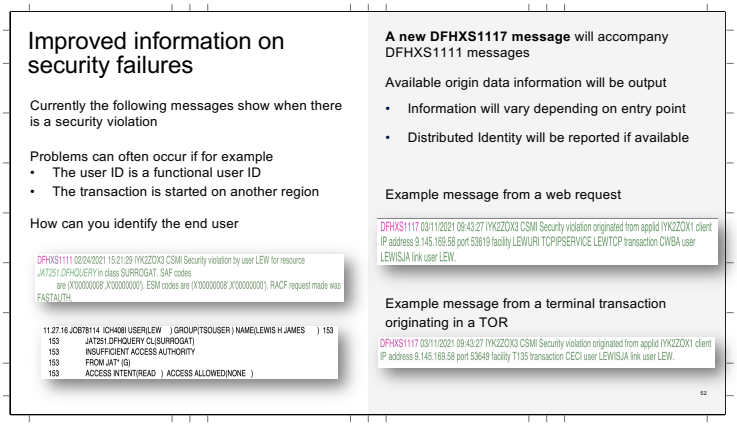
On the far right hand side is information about the release level of CICS and also the five most recent PTFs on the system and their dates. In the screen shot, these are pre GA development regions, so we don't have PTF numbers, instead we have our internal delivery levels. The purpose of this information is to be able to use it in conjunction with the levels shown in the SMP/E datasets, to show that CICS is running at the required service levels as installed via SMP/E.


Simpler diagnosis



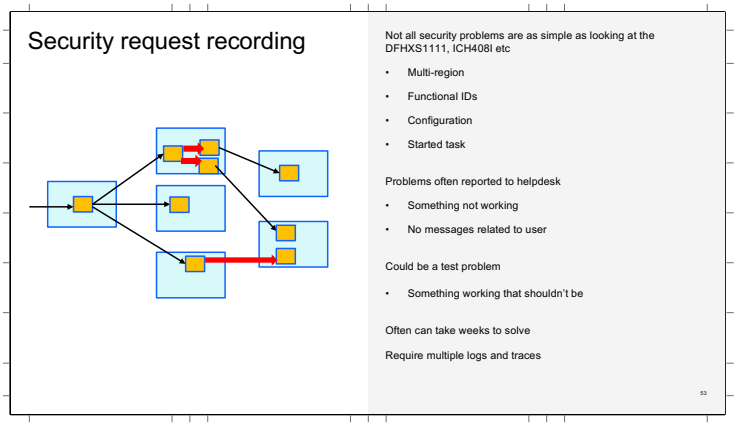
When diagnosing TLS handshake failures, the existing DFHSO0123 message is difficult to understand and interpret because it has so many different inserts. The message has been kept for compatibility reason, but in addition a new set of messages have been produced. There is a different set beginning DFHSO03xx are for client role, DFHSO04xx for server roles.

There are 7 or 8 different messages each for client and server, narrowing down the error, with specific relevant inserts and explanatory text.



A more general security diagnosis problem comes in being able to identify the end user. Sometimes there are functional user ids involved or the transaction was actually started in a different region.

A new DFHXS117 message will now accompany the existing DFHXS1111 message. Available origin data will be displayed.



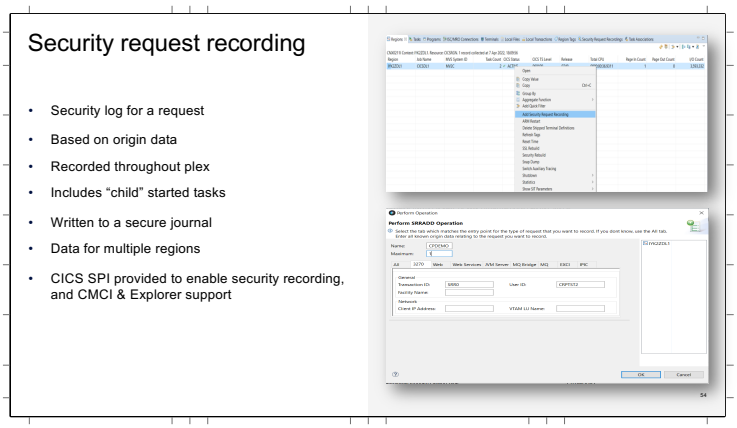
CICS security can be very complex. CICS security has lots of configuration options, therefore there is a lot that can go wrong, a lot of places to look.

Almost certainly there are multiple CICS regions involved (eg TOR/WOR, AORs, FORs etc). You may have dynamic routing which for example could affect whether something works or not, if the regions have inadvertantly been set up differently. You have programs that are linked to, tasks that are started etc.

Individual transaction definitions can have security parameters set on/off, for example whether resource security is active, or command security.

The links between systems also have a lot of security information, for example LINK user ids, and whether of not the task userid is flowed across the link.

RACF classes can be quite complicated.

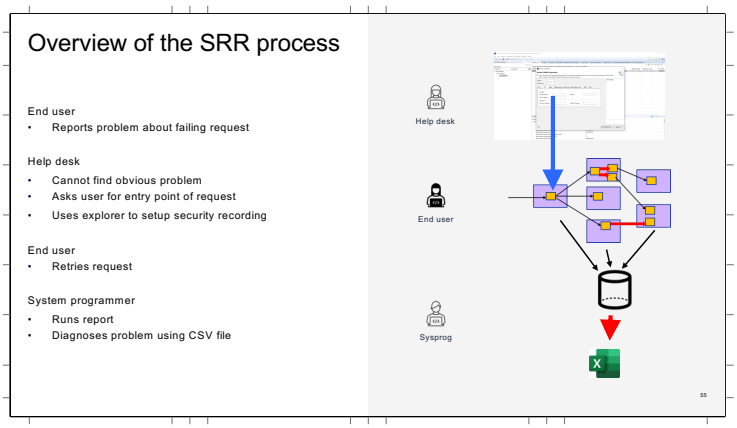


The idea behind Security Request Recording is to have a mechanism to allow a help desk operator to easily obtain all of the data first time, to pass on to a system programmer to diagnose a problem.

It needs to be lightweight to allow it to be used in a production region. It has to be able to work in a sysplex environment, work with all ESMs and be secure.

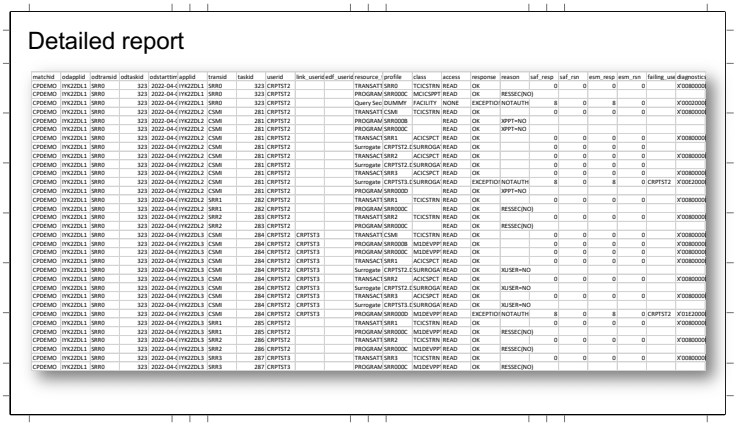
There needs to be an easy was to turn it on and off and to look at the output.

So security request recording provides a way to capture a security log for a request. Its based on origin data, its recorded throughout the plex, wherever the request goes. It includes child started tasks. The data is written to a secure journal. A CICS SPI is provided which is exploited by CICS Explorer via the CMCI.



So the scenario is that a problem is reported to the help desk about a failing request. The help desk cannot find the problem. The help desk asks to the point of entry for the request and activates SRR. The request is rerun.

A system programmer can run a report to interpret the data written to the SMF log and uses the resultant csv file to diagnose the problem.

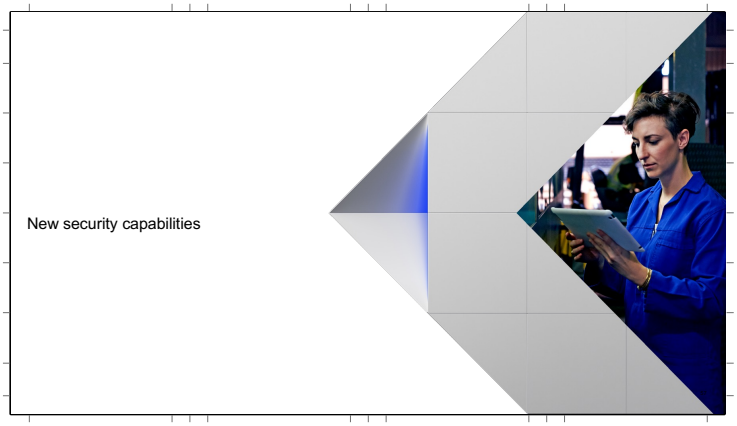


matchId	odaapplid	odtransid	odtaskid	odstarttim	applid	transid	taskid	userid	link_userid	edf_userid	resource	profile	class	access	response	reason	saf_resp	saf_rsn	esm_resp	esm_rsn	failing_use	diagnostics
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL1	SRR0	323	CRPTST2			TRANSAT1	SRR0	TCICSTRN	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL1	SRR0	323	CRPTST2			PROGRAM	SRR000C	MCICSPPT	READ	OK	RESSEC(NO)						
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL1	SRR0	323	CRPTST2			Query Sec	DUMMY	FACILITY	NONE	EXCEPTIO	NOTAUTH	8	0	8	0		X'00020000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	CSMI	281	CRPTST2			TRANSAT1	CSMI	TCICSTRN	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	CSMI	281	CRPTST2			PROGRAM	SRR000B		READ	OK	XPPT=NO						
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	CSMI	281	CRPTST2			PROGRAM	SRR000C		READ	OK	XPPT=NO						
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	CSMI	281	CRPTST2			TRANSAC1	SRR1	ACICSPCT	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	CSMI	281	CRPTST2			Surrogate	CRPTST2.C	SURROGA	READ	OK		0	0	0	0		
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	CSMI	281	CRPTST2			TRANSAC1	SRR2	ACICSPCT	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	CSMI	281	CRPTST2			Surrogate	CRPTST2.C	SURROGA	READ	OK		0	0	0	0		
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	CSMI	281	CRPTST2			TRANSAC1	SRR3	ACICSPCT	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	CSMI	281	CRPTST2			Surrogate	CRPTST3.C	SURROGA	READ	EXCEPTIO	NOTAUTH	8	0	8	0	CRPTST2	X'00E20000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	CSMI	281	CRPTST2			PROGRAM	SRR000D		READ	OK	XPPT=NO						
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	SRR1	282	CRPTST2			TRANSAT1	SRR1	TCICSTRN	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	SRR1	282	CRPTST2			PROGRAM	SRR000C		READ	OK	RESSEC(NO)						
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	SRR2	283	CRPTST2			TRANSAT1	SRR2	TCICSTRN	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL2	SRR2	283	CRPTST2			PROGRAM	SRR000C		READ	OK	RESSEC(NO)						
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	CSMI	284	CRPTST2	CRPTST3		TRANSAT1	CSMI	TCICSTRN	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	CSMI	284	CRPTST2	CRPTST3		PROGRAM	SRR000B	M1DEVPP	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	CSMI	284	CRPTST2	CRPTST3		PROGRAM	SRR000C	M1DEVPP	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	CSMI	284	CRPTST2	CRPTST3		TRANSAC1	SRR1	ACICSPCT	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	CSMI	284	CRPTST2	CRPTST3		Surrogate	CRPTST2.C	SURROGA	READ	OK	XUSER=NO						
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	CSMI	284	CRPTST2	CRPTST3		TRANSAC1	SRR2	ACICSPCT	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	CSMI	284	CRPTST2	CRPTST3		Surrogate	CRPTST2.C	SURROGA	READ	OK	XUSER=NO						
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	CSMI	284	CRPTST2	CRPTST3		TRANSAC1	SRR3	ACICSPCT	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	CSMI	284	CRPTST2	CRPTST3		Surrogate	CRPTST3.C	SURROGA	READ	OK	XUSER=NO						
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	CSMI	284	CRPTST2	CRPTST3		PROGRAM	SRR000D	M1DEVPP	READ	EXCEPTIO	NOTAUTH	8	0	8	0	CRPTST2	X'01E20000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	SRR1	285	CRPTST2			TRANSAT1	SRR1	TCICSTRN	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	SRR1	285	CRPTST2			PROGRAM	SRR000C	M1DEVPP	READ	OK	RESSEC(NO)						
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	SRR2	286	CRPTST2			TRANSAT1	SRR2	TCICSTRN	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	SRR2	286	CRPTST2			PROGRAM	SRR000C	M1DEVPP	READ	OK	RESSEC(NO)						
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	SRR3	287	CRPTST3			TRANSAT1	SRR3	TCICSTRN	READ	OK		0	0	0	0		X'00800000
CPDEMO	IYK2ZDL1	SRR0	323	2022-04-(	IYK2ZDL3	SRR3	287	CRPTST3			PROGRAM	SRR000C	M1DEVPP	READ	OK	RESSEC(NO)						

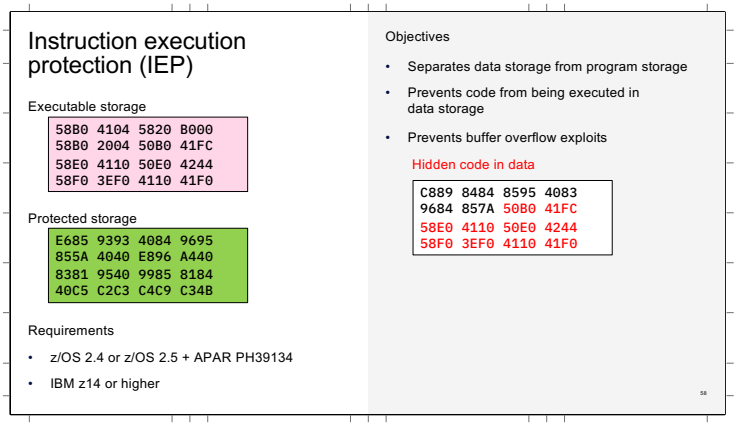
The slide shows the csv file produced by the CICS supplied utility to read the secure SMF log. In this example with have information for a specific request named CPDEMO

We can see the data pertaining to where the request originated, its region, the userids involved. As we scan across we can see all the regions where the request flowed, to, the tasks involved in those regions, and the relevant userids.

Further right we see the security requests that were issued in those regions and what the responses were for those requests.



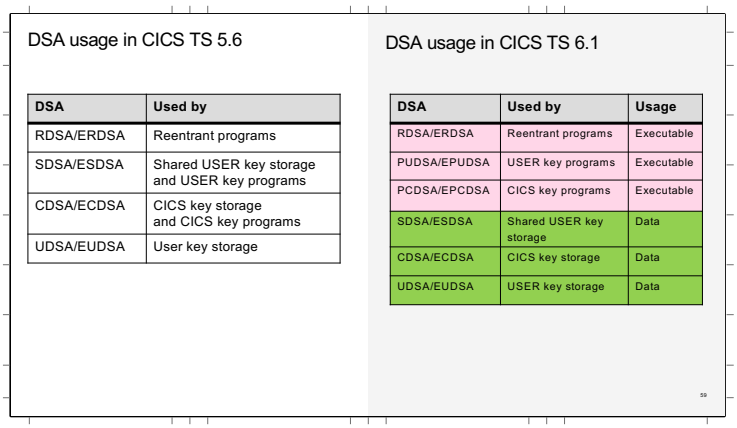
The final section regarding security will talk about new capabilities introduced in CICS TS 6.1 in the area of security.



The first of the new capabilities is something called Instruction Execution Protection (IEP).

IEP provides the ability to separate program storage (executable storage) from data storage. Code can only run on executable storage. This prevents a buffer overflow type vulnerabilty whereby code is hidden in a data area and then some bad actor calls this code.

IEP is supported on z/OS 2.4 and higher together with LE apar PH39134. As far as hardware is concerned, a z14 or higher is required.



In order to support IEP and separate program storage from data storage, CICS has introduced four new DSAs.

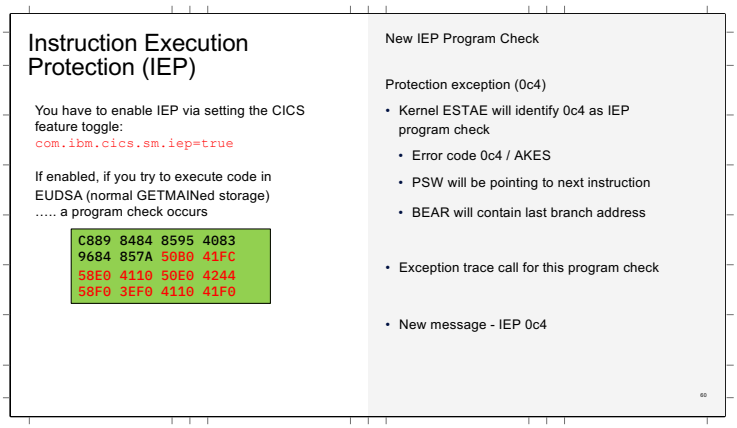
When IEP is active:

The PUDSA, EPUDSA are now used for program storage for user key programs. This is executable storage.

The PCDSA and EPCDSA are now used for program storage for CICS key programs. This is executable storage.

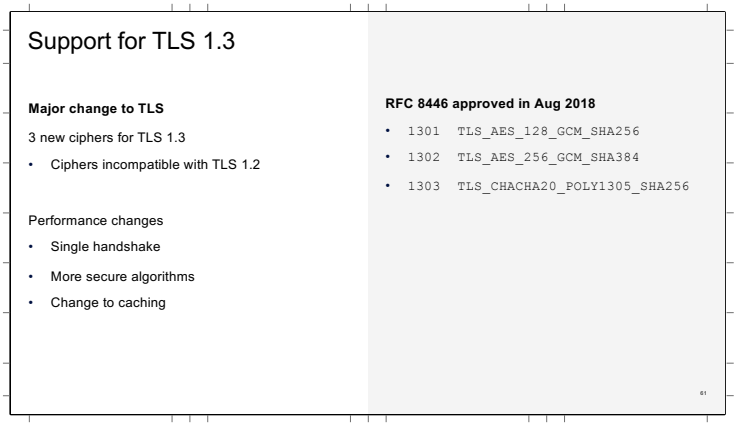
The RDSA and ERDSA remain unchanged and are for reentrant programs.

Data is separated from program storage, so the existing SDSA/ESDSA, CDSA/EXCDSA and UDSA/EUDSA contain only data, no programs.



To activate IEP you activate it via the feature toggle shown on the slide.

If IEP is active and for example you try to execute code in the EUDSA (which is where storage returned as a RESULT of an EXEC CICS GETMAIN typically resides), then a protection exception 0C4 occurs. The CICS kernel will identify this as an IEP program check and will cause an new AKES transaction abend to occur. There are new exception traces written and new DFHSMxxxx messages.

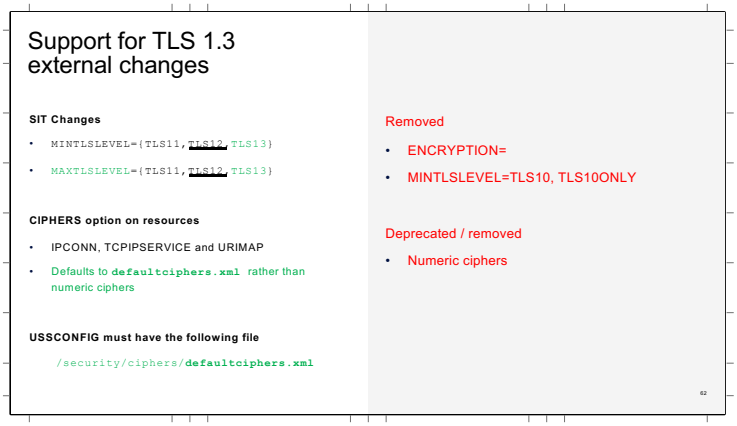


The second new piece of security functionality in CICS TS 6.1 is support for TLS 1.3 protocol.

TLS 1.3 is the first major change to TLS since 2008. The ciphers used are simplified, reduced from 37 used in TLS 1.2 down to 6 in TLS 1.3 or which 3 are supported by z/OS.

There are no common ciphers shared between TLS 1.2 and TLS 1.3

There are performance and of course security changes involved, in particular there is a single flow in TLS 1.3 for the handshake. Which encrypts more of the information involved in the handshake.



CICS was very early adopter of SSL and so has some legacy features which provide a challenge when upgrading to TLS 1.3. CICS endeavours to be backward compatible whenever possible, but in this case there are features for which its no longer possible to support, and indeed no longer desirable to support.

The original implementation used a SIT option of ENCRYPTION with values of STRONG, MEDIUM and WEAK. STRONG allowed TLS 1.0. This mechanism was deprecated several years ago and replaced by the MINTLSLEVEL SIT option.

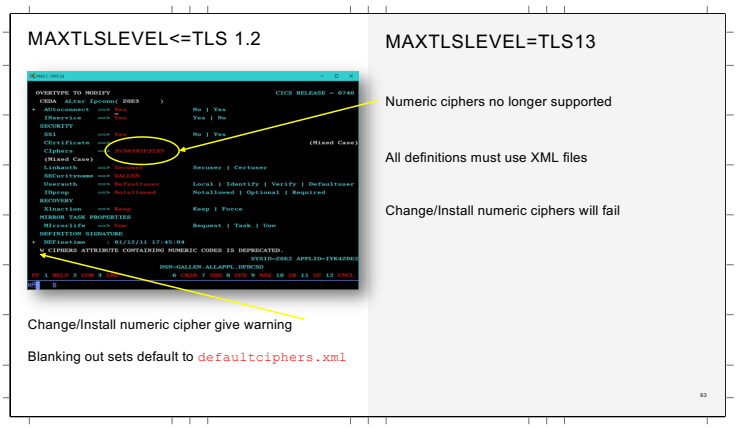
The ENCRYPTION option has now been removed as has values of TLS10 and TLS10ONLY for MINTLSLEVEL., as support for TLS 1.0 is withdrawn in CICS TS 6.1. This protocol is regarded as too weak.

CICS TS 6.1 introduces the MAXTLSLEVEL. So now in simple way you can control the minimum and maximum levels of TLS protocol supported on a CICS region.

The main configuration change is the removal of the ability to specify numeric ciphers on resource definitions. The region for this is that they only allowed specification of two digit ciphers whereas modern ciphers are four digits. Instead the user should use the xml ciphers files introduced in CICS TS 5.2

As a consequence of this, the default of two digit ciphers is no longer useful. Instead the new default ciphers are specified in a new xml file called `defaultciphers.xml`

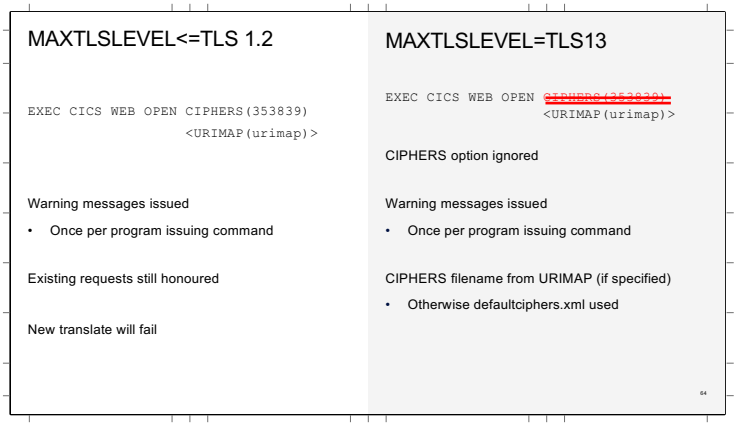
A sample of this file is provided and must be copied to the USSCONFIG directory and reviewed to ensure it contains ciphers you are happy with.



So to use TLS 1.3, there are upgrading tasks that need to be performed.if you haven't already converted to using cipher files.

Note the new MAXTLSLEVEL defaults to TLS12 to allow a clean initial upgrade. With TLS 1.2 the existing numeric ciphers are still supported. You will be warned if you install or modify a definition containing numeric ciphers. If you blank out te ciphers, the new defaultciphers xml file will be used.

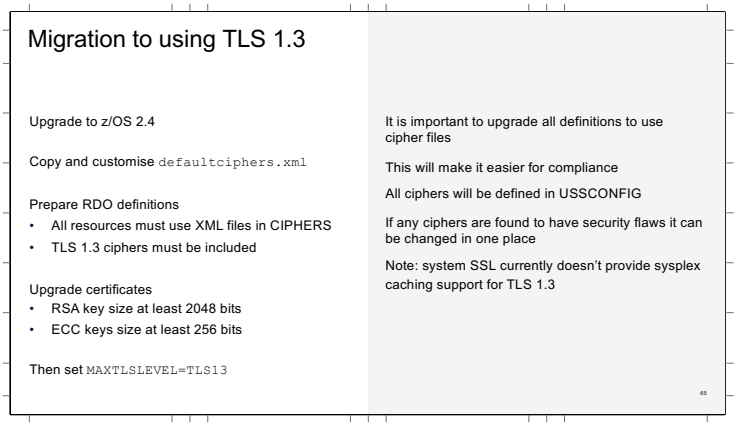
Migrating to TLS 1.3 will involve a number of steps, the last of which will be to set MAXTLSLEVEL to TLS13. After this change, all inbound and outbound connections will use TLS 1.3 if the connected system supports it. Hence the TLS 1.3 ciphers need to be in the cipher file for these connections and its no longer possible to work with numeric ciphers on resource definitions. Error messages are produced if resource definitions containing numeric ciphers try to be defined, or try to be installed.



The WEB OPEN CIPHERS command also presents a problem.

With MAXTLSLEVEL of TLS12 we will continue to support existing options. A warning msg (once per program issuing the command) are issued warning that numeric ciphers are deprecated. New or changed programs will fail to translate if they use the option.

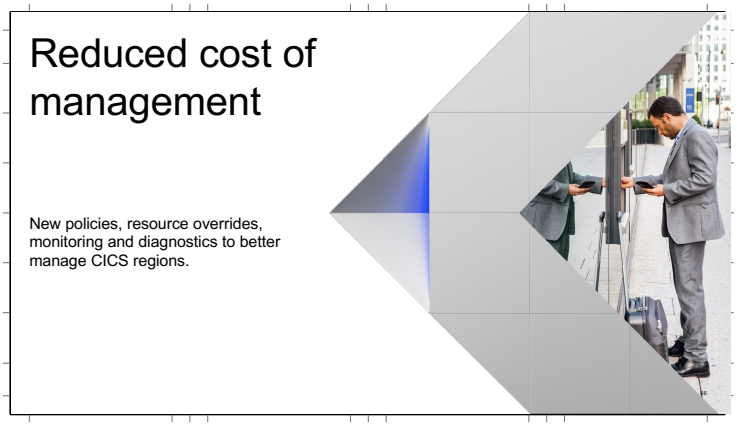
With MAXTLSLEVEL of TLS13, the ciphers option is ignored. Again warning messages are issued. The name of the ciphers file from the URIMAP if used, else defaultciphers.xml is used.



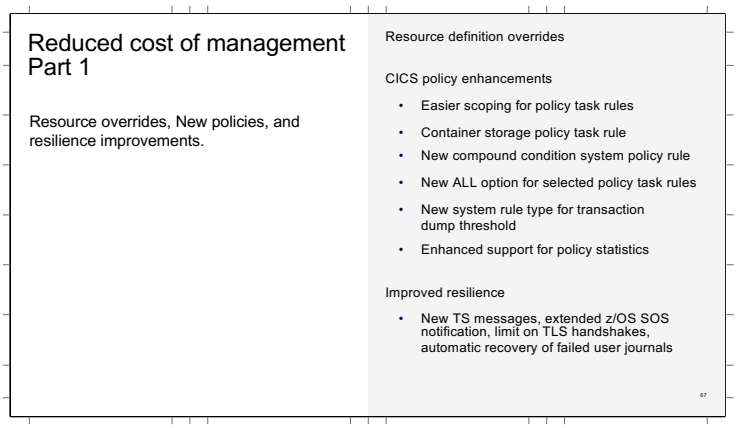
So in summary, the upgrade to TLS 1.3 is not as easy as upgrading to previous protocol levels. The incompatibility of TLS 1.3 ciphers with TLS 1.2 ciphers means that upgrade actions are required.

The CICS documentation gives a clear step by step migration path to hopefully provide a smooth upgrade to TLS 1.3

The benefit of upgrading is the greater security provided by TLS 1.3. By using xml files, future upgrades will be more controllable and auditable..



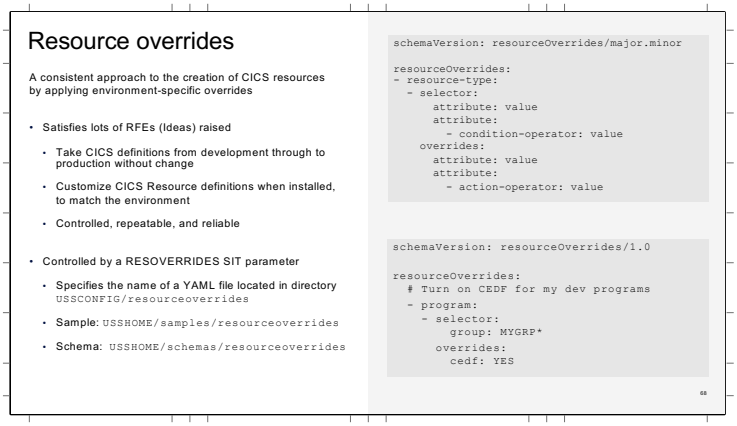
The third a final theme of CICS TS 6.1 is reduced cost of management, which includes a whole host of Foundational enhancements.



Resource Overrides satisfies a large number of RFEs as well as positioning CICS for the future as we move towards support for containerisation.

Policies provide an easier way for customers to add to their overall resilience.

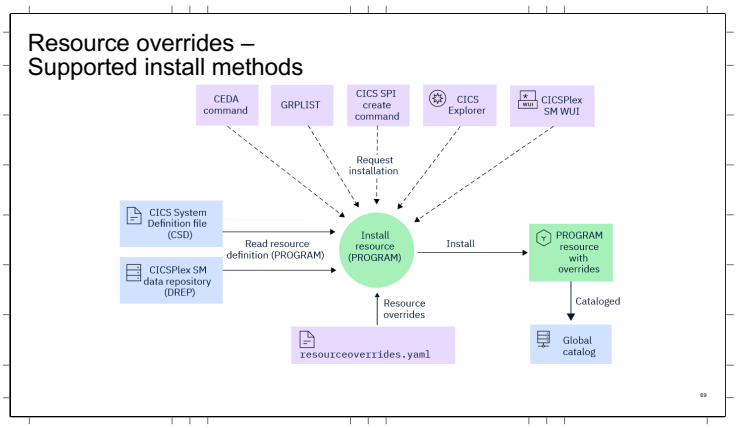
There have been a number of Foundational items to build in more CICS resilience.



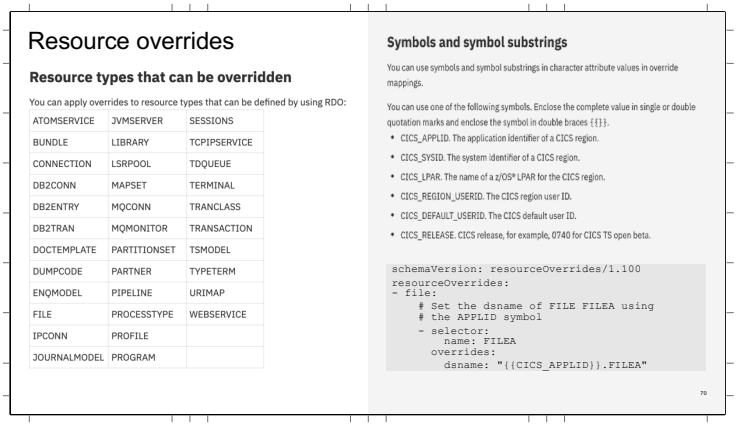
You can override existing resource definitions at the point that those resources are installed. This allows you to tailor resources or apply standards for a specific CICS environment, such as test or development. You can specify and apply resource overrides to any resource that can be defined by using RDO (CSD or BAS).

Overriding resource definition involves the following process:

- Specify the required overrides in a YAML file (*resource overrides file*). You can store that file directly in your source code management (SCM) system.
- Enable the resource definition overrides capability in the region and make the resource overrides file available.
- Install CICS resources and apply resource definition overrides as the resources are installed.
- Monitor overridden fields, values, and errors through CICS messages and CICS audit capabilities.



Whatever method of install is used. Resource overrides can be used to change what is installed into CICS. The definition itself is not changed, only the installed resource in CICS.



Resource overrides

Resource types that can be overridden

You can apply overrides to resource types that can be defined by using RDO:

ATOMSERVICE	JVMSERVER	SESSIONS
BUNDLE	LIBRARY	TCPIPSERVICE
CONNECTION	LSRPOOL	TDQUEUE
DB2CONN	MAPSET	TERMINAL
DB2ENTRY	MQCONN	TRANCLASS
DB2TRAN	MQMONITOR	TRANSACTION
DOCTEMPLATE	PARTITIONSET	TSMODEL
DUMPCODE	PARTNER	TYPETERM
ENQMODEL	PIPELINE	URIMAP
FILE	PROCESSTYPE	WEBSERVICE
IPCONN	PROFILE	
JOURNALMODEL	PROGRAM	

Symbols and symbol substrings

You can use symbols and symbol substrings in character attribute values in override mappings.

You can use one of the following symbols. Enclose the complete value in single or double quotation marks and enclose the symbol in double braces {{}}.

- CICS_APPLID. The application identifier of a CICS region.
- CICS_SYSID. The system identifier of a CICS region.
- CICS_LPAR. The name of a z/OS® LPAR for the CICS region.
- CICS_REGION_USERID. The CICS region user ID.
- CICS_DEFAULT_USERID. The CICS default user ID.
- CICS_RELEASE. CICS release, for example, 0740 for CICS TS open beta.

```
schemaVersion: resourceOverrides/1.100
resourceOverrides:
- file:
    # Set the dsname of FILE FILEA using
    # the APPLID symbol
    - selector:
        name: FILEA
      overrides:
        dsname: "{{CICS_APPLID}}.FILEA"
```

70

All 34 RDO resource types are supported

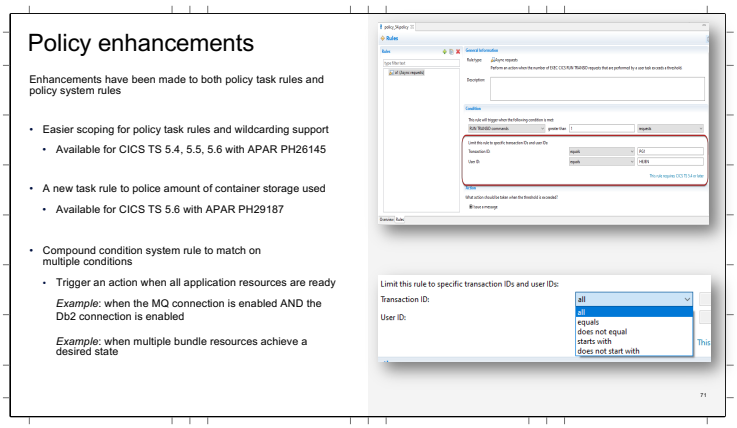
- No support for any of the ‘bundle only’ resources, e.g. Policy, Event Bindings e.t.c.
- However any RDO resource types that are defined in CICS bundles can be overridden, e.g. PROGRAM, TRANSATION, LIBRARY etc.

Allows resource attributes of ANY supported resource to be overridden as a resource is installed into CICS region

- The only attributes that cannot be overridden are the resource name, and for CSD installs the RDO group name

You cannot override any resource that is:

- Defined in a “DFH” group except those in any auto-install model groups
- CREATE’d by CICS, i.e. those with a DEFINESOURCE of “SYSTEM”



When you define a policy task rule, you can now limit this rule to be triggered when status changes are made in relation to a specific transaction or a range of transactions, a specific user ID or a range of user IDs, or both. This includes support for wildcarding and the ability to include or exclude sets of transactions.

To specify this limit, you can set Transaction ID and User ID filters in the Condition section in the Rules tab of the Policy definition editor.

This capability is also available on CICS TS 5.4, 5.5, and 5.6 with APAR PH26145.

There is a new task rule to limit the amount of container storage that can be used.

•Use this rule type to define a threshold for the amount of container storage allocated to a user task, and take an automatic action if the threshold is exceeded. This rule does not apply to EXCI containers or BTS containers.

•Channels and containers use 64-bit storage in CICS, so this rule monitors 64-bit storage allocated to containers for a user task.

A new compound condition system rule allows an action to be taken when multiple conditions are met. There is no limit to the number of conditions.

•Use this rule type when you want to define a system rule that specifies two or more conditions. CICS takes the defined action when all the specified conditions are met. For example, you can define a compound condition system rule that instructs CICS to set the z/OS WLM health status to OPEN only if both the Db2® connection status and the IBM® MQ connection status are CONNECTED.
You can specify filters based on the following types:

•Bundle available status

•Bundle enable status

•Db2 connection status

•DBCTL connection status

•File enable status

•File open status

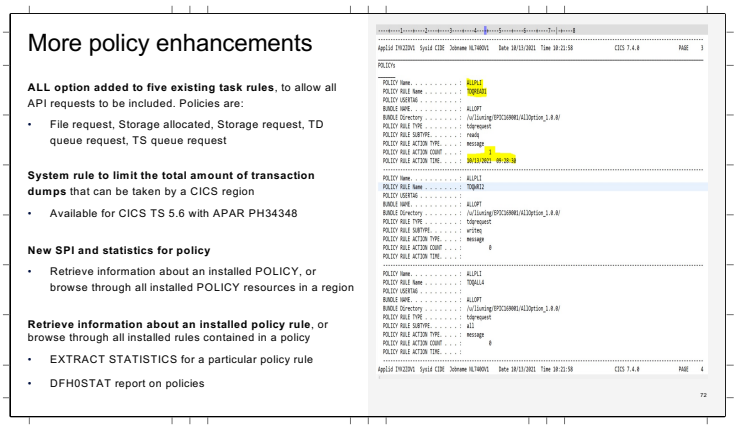
•IBM MQ connection status

•IPIC connection status

•MRO connection status

•Pipeline enable status

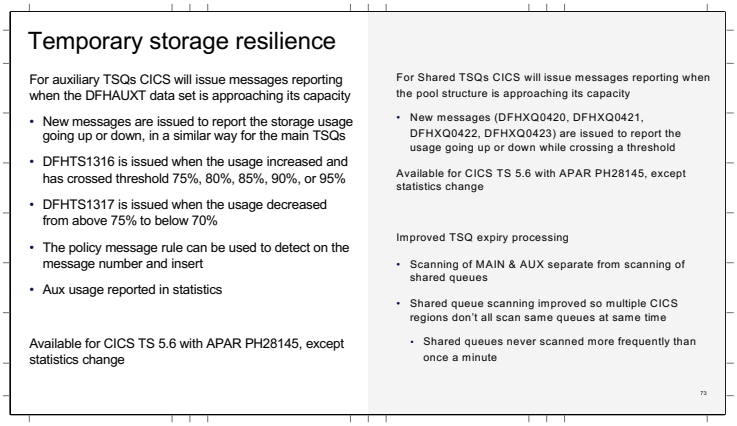
•Program enable status



For five existing policy task rules that allow are specific API request to be selected, there is a new ALL option to filter on all requests of that type, for example all File Control requests rather than selecting a particular API request, like WRITE FILE.

There is a new policy system rule to take action when the total number of transactions dumps exceeds a threshold. It allows action to be taken before the rogue region affects other regions on the same LPAR.

The sample statistics program DFH0STAT can now produce Policy reports. The Policy report shows information and statistics about installed policy rules in the region. In support for this enhancement, the EXTRACT STATISTICS system programming command supports a new RESTYPE option POLICY and a new SUBRESTYPE option POLICYRULE, which can be used to obtain statistics about a policy rule that is contained in a POLICY resource.
In addition, two new system programming commands INQUIRE POLICY and INQUIRE POLICYRULE have been introduced to support inquiries on information about installed POLICY resources and the policy rules contained within.



You are now alerted when auxiliary temporary storage data set usage is approaching a high percentage of its capacity so that you have time to free up storage before the auxiliary temporary storage becomes full.

CICS issues message DFHTS1316 when 75% or more of the maximum auxiliary temporary storage is in use, and message DFHTS1317 when storage usage falls below 70% of the maximum auxiliary temporary storage.

New statistics are available in Temporary storage: Global statistics to provide information about the current and peak percentage of auxiliary temporary storage being used.

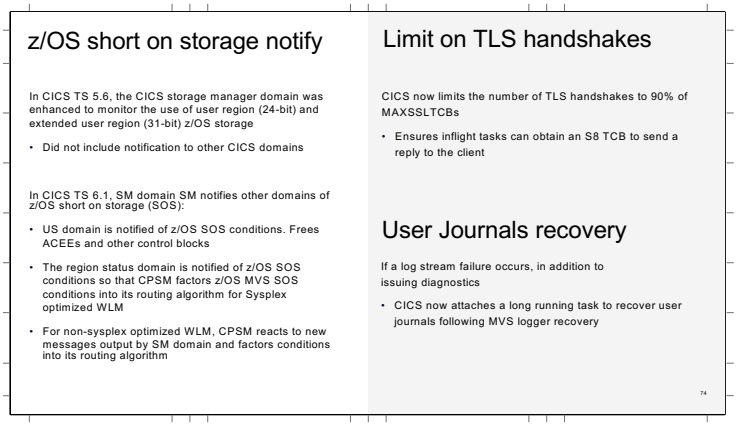
This capability is partially available on CICS TS 5.6 with APAR PH28145.

Likewise, there is similar functionality to monitor shared TS queues. This enhancement makes it easier for you to monitor capacity usage change for shared pool TS queues. When the percentage of entries or elements in use in a pool structure reaches a specified threshold, DFHXQ0422 or DFHXQ0423 is issued. When the percentage of entries or elements in use drops below a threshold, DFHXQ0420 or DFHXQ0421 is issued.

The processing of expired temporary storage queues has been improved as follows:

- Firstly, the processing of main and auxiliary tsqueues is separated from the processing of shared tsqueues so that they use separate calculated intervals.
- Secondly, for shared tsqueues, an internal queue is used to hold when the last scan was performed. The internal queue is used to prevent a CICS region from scanning shared TS queues if another CICS region has performed such a scan within the previous minute. This means that even if multiple CICS regions are using a shared TS pool, each with TS models installed that specify short expiry intervals, the shared queues are never scanned more frequently than once per minute.
- Thirdly, the CICS-MQ interface has been improved to only employ a DFHCKBR tsmodel with a nonzero expiry interval when the MQ bridge has been started; otherwise, it has a zero expiry interval. This avoids unwanted tsqueue scans.

This capability is also available on CICS TS 5.6 with APAR PH40863 and PH40409.



Some more resilience items to mention…..

CICS has long provided monitoring and short on storage (SOS) support for CICS-managed storage in dynamic storage areas (DSAs), which includes the capability of the CICS storage manager domain to notify other CICS domains so that they can take action upon an SOS event in CICS DSAs. In CICS TS 5.6, the CICS storage manager domain was enhanced to monitor the use of user region (24-bit) and extended user region (31-bit) MVS™ storage not managed by CICS, but this enhancement did not support SOS notification to other domains. In CICS TS 6.1, the SOS notification is enhanced to provide the same notification support for MVS storage SOS events as for CICS DSA SOS events.

- The DFHUS domain is notified of z/OS MVS SOS conditions so that any eligible user ID and its associated attributes are freed, including RACF control blocks. The freeing of these control blocks is normally subject to USRDELAY processing, but in the event of an SOS condition in 31-bit MVS storage, these control blocks are now freed immediately by the US and XS domains.
- The Region status domain is notified of z/OS MVS SOS conditions so that CICSPlex SM factors z/OS MVS SOS conditions into its routing algorithm, in the same way as it does for CICS-managed storage SOS conditions.

CICS TS 6.1 limits the number of concurrent TLS handshakes to 90% of the MAXSSLTCBS value specified at startup. If the maximum limit is reached, a task that is requesting a TLS handshake is suspended with a resource name of S8TLSHS of resource type DSWC.

To help you monitor concurrent TLS handshakes in a CICS region, new statistics are introduced in TCP/IP Global statistics. These statistics provide information about the maximum, current, and peak numbers of TLS handshakes that are running in parallel or that are waiting.

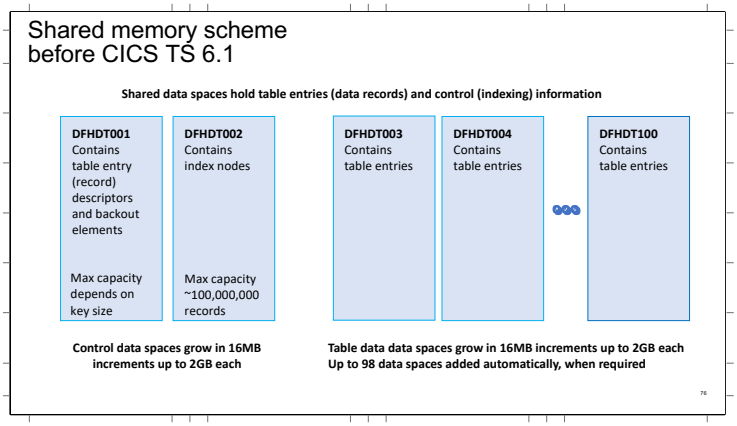
This enhancement helps avoid issues such as high CPU, MAXTASK, or lack of S8 TCBs when many TLS handshakes are performed concurrently. It also allows in-flight web alias or pipeline tasks to obtain an available S8 TCB in order to send a reply back to the client in the same situation.

When a log stream failure occurs, in addition to issuing message DFHLG0772 and taking a system dump, CICS now attaches CLGR at the time DFHLG0772 is issued. The new transaction CLGR attempts to recover and reset the failed user journal automatically for up to 60 minutes. This gives you an opportunity to fix the log stream problem, then allowing CICS to automatically recover journals for you. However, this feature comes with a cost in potential more system dumps being taken following a failed user journal, but you can control the number of system dumps taken.

Reduced cost of management Part 2

Increased data capacity, New cross region capabilities, new monitoring and diagnostics to better manage CICS regions.

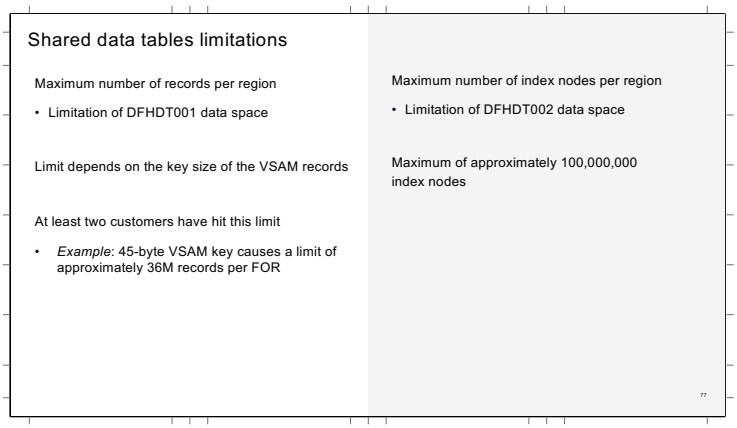
- Increased capacity of shared data tables
- Enhancements in support of IBM Db2
- Enhanced CICS event processing support
- Enhanced performance monitoring
- Enhanced API, SPI, and diagnostics
- Enhanced CPSM Sysplex Optimized WLM



The slide shows how Shared Data tables is supported prior to CICS TS 6.1. z/OS data spaces are used to hold the customer's VSAM data. There can be up to 98 of these dataspaces, each 2GB in size.

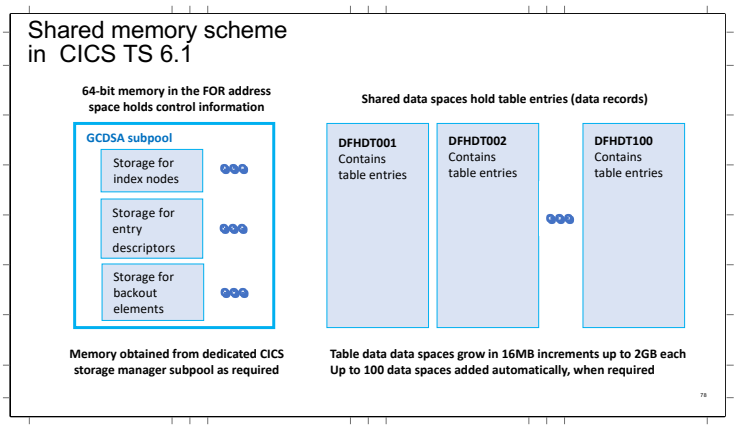
There are two 2GB control data spaces that hold meta data.

Data space DFHDT001 holds record descriptors and backout elements. Its maximum capacity depends upon how big the keys are in the customer's VSAM data.
Data space DFHDt002 holds index information, it can hold information relating to 100 million records.



We have customers who are storing very large amounts of data in shared data tables. These customers have not exhausted the number of datapsaces used to hold the VSAM records, but they have exhausted the two control data spaces that hold the meta data, because so many records were being stored.

For example, a 45-byte key would mean a limit of 36 million records per file owning region (FOR), and this limit on index information was reached long before all the data space storage available to hold the records was consumed.



The capacity of shared data tables has been increased. Shared data tables no longer use the two control data spaces named DFHDT001 (which was used for table entry descriptors and backout elements) and DFHDT002 (which was used for index nodes), and instead are now using 64-bit storage to hold this control information.

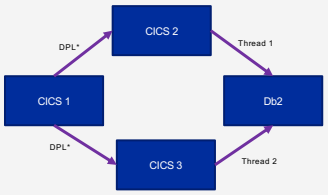
The use of 64-bit storage to hold the entry descriptors, backout elements, and index nodes removes the constraint on the number of records that can be stored. The records continue to be stored in 31-bit data spaces. Now, two more data spaces are available to hold the records, increasing from 98 to 100 data spaces.

Solving the diamond deadlock problem

The problem: Two routes into Db2 from two regions within the same global UOW look like two different units of work to Db2

Examples:

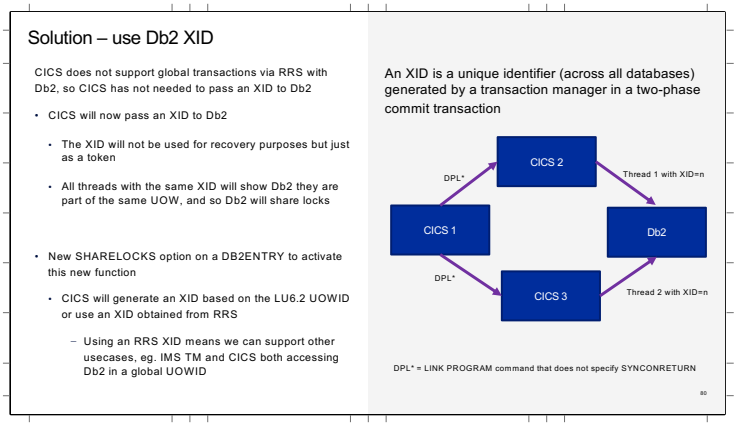
- CICS1 DPLs (non sync-on-return) to a program in CICS2 which updates Db2
  - CICS-Db2 Attach Db2 creates Thread 1, Db2 holds lock for thread 1 on updated resource
  - Control return to CICS 1 without syncpointing
- CICS 1 DPLs (non sync-on-return) to a program in CICS 3 which tries to update the same data in Db2
  - CICS-Db2 Attach Db2 creates Thread 2
  - Db2 waits for the lock to be released by Thread 1 – the result is deadlock



DPL* = LINK PROGRAM command that does not specify SYNCONRETURN

79

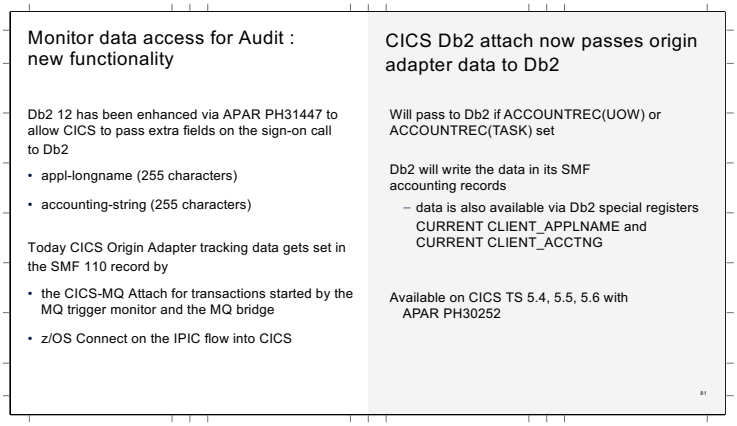
The slide show how today, two CICS regions accessing DB2 can result in an application in a global unit of work can deadlock itself. This is because DB2 is unaware that the two threads are part of the same global unit of work.



A new DB2ENTRY attribute SHARELOCKS is provided to enable CICS to pass an XID to Db2 and instruct Db2 to share locks between threads that pass the same XID. Using the same XID, other threads that originate from other CICS regions or from other transaction managers such as IMS TM can access Db2 in the same global unit of work (UOW). The XID token is not used for recovery between CICS and Db2.

The passing of an XID involves a partial signon to Db2 for each UOW. This action closes cursors, so held cursors across syncpoints are not supported when the passing of an XID is enabled. Applications will have to reposition cursors after a syncpoint. Passing an XID avoids having to deal with UOW affinities.

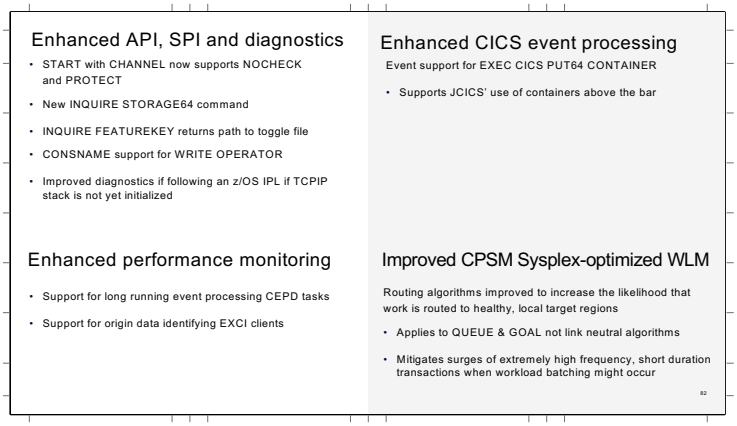
This capability is also available on CICS TS 5.5 and 5.6 with APAR PH39766, but is facilitated by feature toggle com.ibm.cics.db2.sharelocks={true|false}.



The CICS Db2 attachment facility is enhanced to pass adapter data to Db2. If a CICS task that is accessing Db2 has adapter data in the CICS origin data, the adapter ID is passed as appl-longname and the adapter data is passed as an accounting-string. Db2 writes the data in its SMF accounting records and the data is also available online through the Db2 special registers CURRENT CLIENT_APPLNAME and CURRENT CLIENT_ACCTNG.

This capability requires Db2 12 with APAR PH31447 or higher.

This capability is also available on CICS TS 5.4 through 5.6 with APAR PH30252.



A number of enhancements have been made for API, SPI and diagnostics.

- The START command with CHANNEL now supports the NOCHECK and PROTECT options. This makes it easier to migrate from passing data by interval control (START FROM) to passing data by using a channel (START CHANNEL). When you use a channel to pass data for a START request, you can now use the NOCHECK option to indicate that the request must be shipped to a remote system and no response is expected by the starting task, thus improving CICS performance. With the PROTECT option, you can make the START request effectively recoverable by instructing the starting task to take a syncpoint before committing the START request.
- INQUIRE STORAGE64 provides similar capability to INQUIRE STORAGE, but for 64-bit storage., to inquire on 64 bit storage owned by a task.
- The INQUIRE FEATUREKEY command now returns the full pathname of the zFS file that specifies the toggle setting being used.
- The WRITE OPERATOR command can now use the name of a console as an alternative to a routecode.
- New diagnostics are produced following a z/OS IPL, if CICS tries to connect to the TCP/IP stack before the stack is initialized and guidance given that a retry should be attempted.

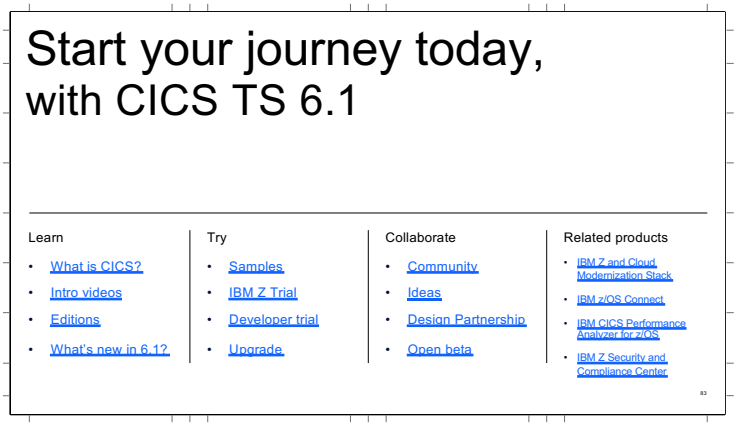
Monitoring enhancements include:

- Monitoring records output for the event processing CEPD long running task.
- The ability to identify requests that originated from EXCI clients.

Event processing can be instrumented to capture data from PUT64 CONATINER commands.

The default behavior of CICSPlex SM workload management routing algorithms has been updated to increase the likelihood that work is routed to healthy, local target regions. This change applies only to the QUEUE and GOAL algorithms, not to the link neutral variants (LNQUEUE and LNGOAL).

Where a routing region might be subject to surges of extremely high frequency, short duration transactions, workload batching might occur. A new feature toggle, com.ibm.cics.cpsm.wlm.surgeresist={true|false}, has been introduced to mitigate these surges by reducing the likelihood that recently selected target regions are reselected.



The slide gives a set of hyperlinks to more information