

**IBM CICS**

Transaction Server for z/OS

# Modern Language Applications in CICS

## Node.js and Java updates

**Steve Fowlkes**

[fowlkes@us.ibm.com](mailto:fowlkes@us.ibm.com)

@SAFowlkes



©2019 IBM Corporation

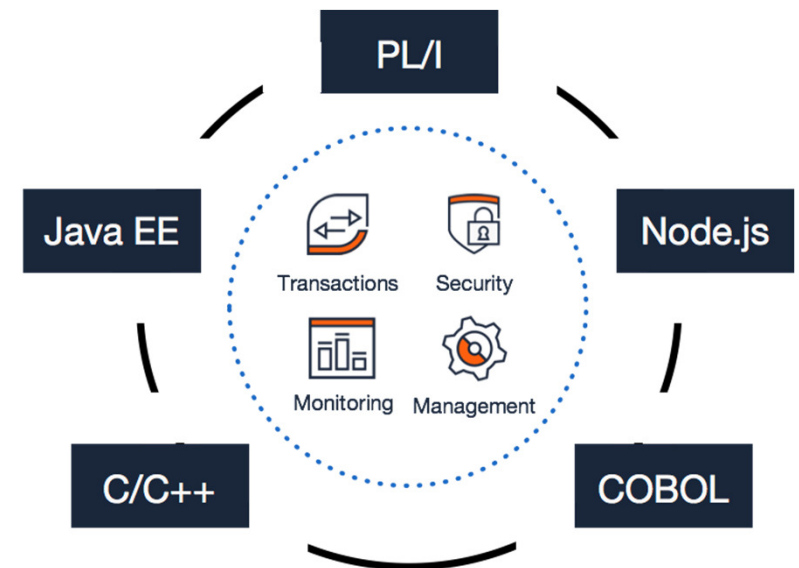
# *Session Agenda*

- *Support for Node.js in CICS*
  - *Introduction*
  - *Why Node.js in CICS*
- *Java in CICS*
  - *Why Java in CICS*
  - *Latest CICS updates*



# Unparalleled mixed-language application server

- **IBM CICS Transaction Server** has evolved to become the world's most powerful mixed language application server.
- Applications can share core programming contexts such as **transactionality, security, monitoring** and **management**, regardless of the language its components are written in, and take full advantage of IBM Z.
- CICS TS V5 allows developers to create incredible mixed-language applications, that include **Java EE 8 Full Platform** capabilities, with first-class interoperability.
- CICS TS V5.5 adds **Node.js** support.



**IBM CICS**  
Transaction Server for z/OS

## Support for Node.js applications



©2019 IBM Corporation

# JavaScript™

**First released in Netscape Navigator 2.0 in 1995**

**Scripting programming language for [dynamic content](#)**

- Interact with backend servers
- Typically used with HTML and CSS
- User interactions, asynchronous API calls, ...
- Language that is used within web browsers



**Designed to complement Java with a similar syntax**

**Standardised as ECMAScript**

- 1<sup>st</sup> edition 1997, latest 10<sup>th</sup> edition 2019

**Most browsers embed a [JavaScript runtime](#) and compete on its performance**

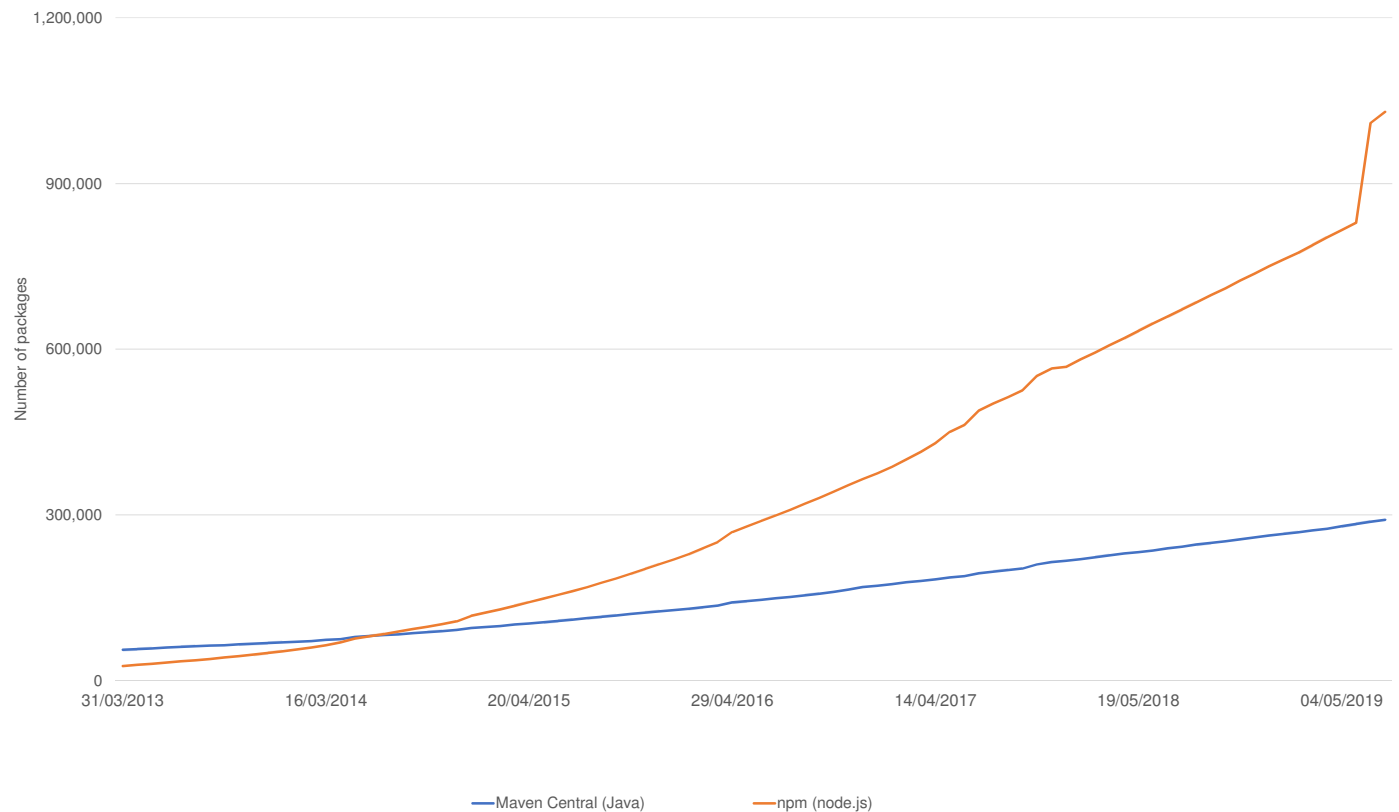


# Why do people choose JavaScript?

- Reusable skills
- Huge amount of reusable packages
- Quick time to market
- Isomorphic apps
- High performance

# The JavaScript ecosystem of packages

- JavaScript offers unparalleled reuse
- JavaScript has a culture of sharing and reuse
- The ecosystem continues to grow rapidly, coupled with regular releases of the core language specification



# Node.js

## Server-side JavaScript runtime platform

- Governed by the [Node.js Foundation](#)
- Built on Google's V8 JavaScript engine

## Designed to build scalable network applications

- Best suited for data and [I/O intensive applications](#)
- Focused on high network bandwidth applications

## Lightweight and efficient

- event-driven
- single-threaded
- non-blocking I/O model
  - leverages the underlying asynchronous I/O support in z/OS

## Module-driven approach to application design and development

- Scalable and encourages agile practices
- Becoming a favored choice for [digital transformation](#)
  - has the ability to provide and aggregate REST services





# Why use JavaScript and Node.js ?

**JavaScript is ubiquitous – client, server, cloud, browsers, embedded systems**

**Large ecosystem of frameworks and tools for application development**

- 950K+ modules available via the [Node Package Manager \(NPM\)](#) ecosystem
- Modules are already available for most tasks
  - saving considerable time for Node.js application developers

**Fast moving, community driven**

- High performance runtimes driven by competition in browsers
- “Battle tested” frameworks

**JavaScript on servers**

- Leverage huge JavaScript developer ecosystem
- Reuse components, tools, concepts, community

# Node.js on z/OS

- Recently updated to Node.js [version 12](#)
- Inherit the acclaimed capabilities of IBM Z
  - With full connection to the IBM Z critical assets
  - Leverage the trusted environment of IBM Z
    - maximize the [security and uptime](#) of critical Node.js applications
- [Co-locating](#) data and the Node.js application on the same system will keep the data secure
  - Control what data will be exposed externally
  - While keeping all confidential data secure on the IBM Z platform
- Node.js runtime on z/OS has the same capabilities as on distributed platforms
  - Extended Node.js packaging and capabilities with [z/OS specific extensions](#)
    - the Node.js runtime is available via SMP/E installation
    - Node.js applications can be invoked from OMVS, USS, a JCL job
- Strong [Tailored Fit pricing](#) models are available



# Why use Node.js in CICS ?

## Host APIs and web applications that consume APIs and data on z/OS

- Add logic to existing APIs
- Aggregate APIs and data
- Reuse modules to access external APIs
- Focuses on the **server side** and exposing backend services as APIs in a programmatic way

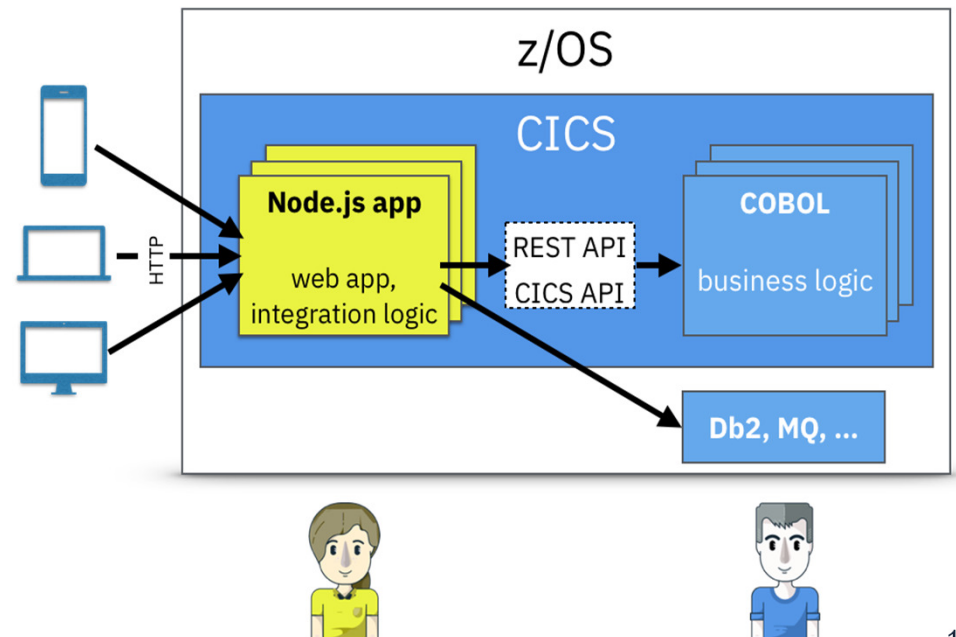
## Co-location for optimized response times

## Simplified deployment and management with CICS applications

## Welcome a new set of API and front-end developers onto z/OS platform

[CICS Developer Center](#) for Q&A

APAR [PH18618](#) for Node.js 12 support



# Why use Node.js in CICS ?

- A Node.js application in CICS might **aggregate calls** to existing business logic functions
  - To provide a single service interface for a front-end application
  - Avoiding the need for the front-end application to make several network calls
  - A Node.js application can also **add functionality** to existing business logic
    - by calling external services or by using NPM modules
- Node.js applications can **call services hosted in CICS** in order to invoke existing business logic
  - These could be JSON or SOAP web services
  - Exposed by using CICS web services technology or z/OS Connect
- Node.js applications can call CICS services by using **NPM modules**
  - Used for making HTTP requests and for consuming JSON and SOAP web services
- When a Node.js application is **hosted in the same CICS region** as a JSON web service
  - A locally optimized transport can be used
  - Uses a cross-memory approach to call the service
  - Must use the **ibm-cics-api** module
  - Service must be exposed using **CICS JSON** web services technology
    - suitable PIPELINE and URIMAP resources must exist

## The ibm-cics-api library

- The `ibm-cics-api` doesn't just work inside CICS
  - You can use it to call remotely into CICS too
- Develop the Node.js app on your own machine
  - Then test that same app without having to deploy it anywhere
- Enable in CICS by adding a `TCPIP SERVICE` in front of the `URIMAP` and `PIPELINE` that will be handling the request from Node.js
- The transport will now use HTTP rather than a cross-memory call



# Node.js application in CICS

## Add Node.js application to a CICS bundle

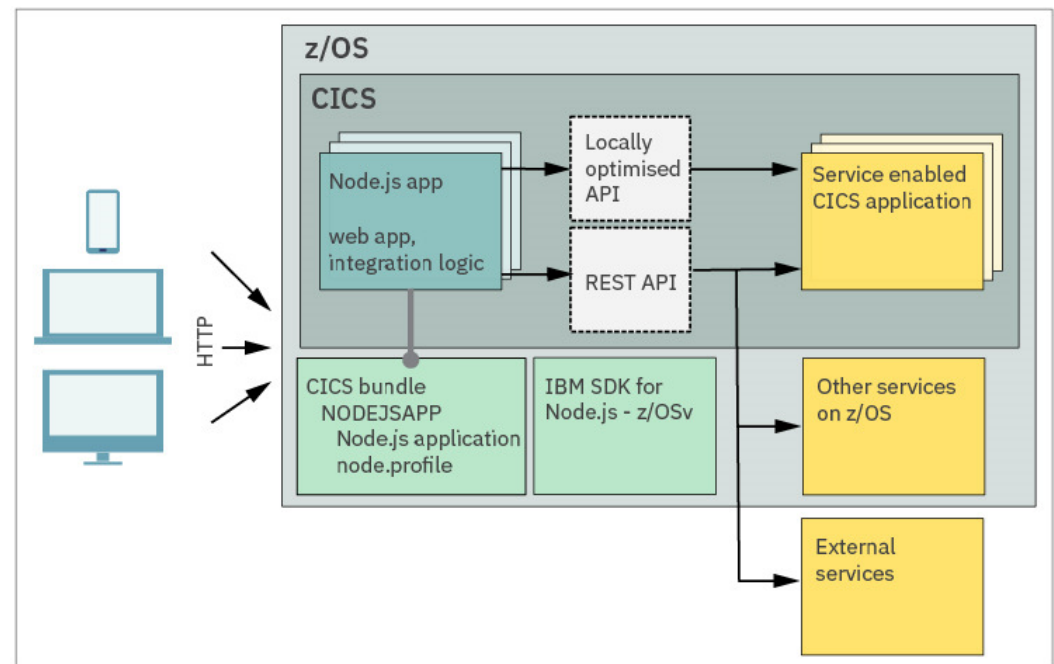
1. NODEJSAPP bundle part
2. Profile
3. Nodejs. application
  - Start script
  - Other application assets

## Build CICS bundle from source and deploy to zFS

- CICS build toolkit
- CICS Explorer
- Zowe CLI
- Tag text files, otherwise EBCDIC assumed
- Run npm to resolve dependencies
- DFHDPLOY, CICS TS plug-in for UCD, ...

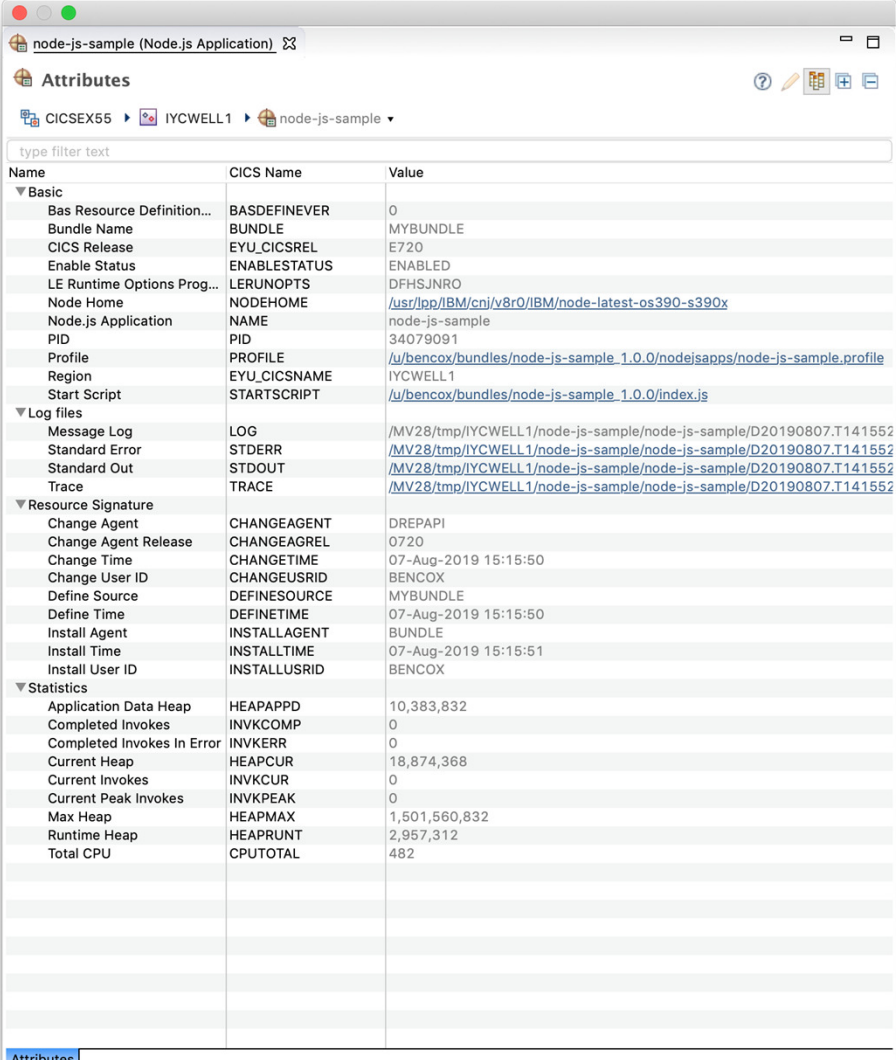
## Lifecycle Management via CICS bundle as usual

- CEDA, CEMT, SPI, CICS Explorer, CMCI, ...
- Node.js app is running when bundle part enabled
- IBM SDK used by CICS to run the application
- Unix signals used by CICS to end application



# Using Node.js in CICS

- The **NODEJSAPP** resource gives information on
  - Configuration
  - Status
  - Logs
  - Statistics



node-js-sample (Node.js Application)

Attributes

CICSEX55 ▶ IYCWELL1 ▶ node-js-sample ▼

type filter text

Name	CICS Name	Value
▼ Basic		
Bas Resource Definition...	BASDEFINEVER	0
Bundle Name	BUNDLE	MYBUNDLE
CICS Release	EYU.CICSREL	E720
Enable Status	ENABLESTATUS	ENABLED
LE Runtime Options Prog...	LERUNOPTS	DFHSJNRO
Node Home	NODEHOME	<a href="#">/usr/lpp/IBM/cnj/v8r0/IBM/node-latest-os390-s390x</a>
Node.js Application	NAME	node-js-sample
PID	PID	34079091
Profile	PROFILE	<a href="#">/u/bencox/bundles/node-js-sample_1.0.0/nodejsapps/node-js-sample.profile</a>
Region	EYU.CICSNAME	IYCWELL1
Start Script	STARTSCRIPT	<a href="#">/u/bencox/bundles/node-js-sample_1.0.0/index.js</a>
▼ Log files		
Message Log	LOG	<a href="#">/MV28/tmp/IYCWELL1/node-js-sample/node-js-sample/D20190807.T141552</a>
Standard Error	STDERR	<a href="#">/MV28/tmp/IYCWELL1/node-js-sample/node-js-sample/D20190807.T141552</a>
Standard Out	STDOUT	<a href="#">/MV28/tmp/IYCWELL1/node-js-sample/node-js-sample/D20190807.T141552</a>
Trace	TRACE	<a href="#">/MV28/tmp/IYCWELL1/node-js-sample/node-js-sample/D20190807.T141552</a>
▼ Resource Signature		
Change Agent	CHANGEAGENT	DREPAPI
Change Agent Release	CHANGEAGREL	0720
Change Time	CHANGETIME	07-Aug-2019 15:15:50
Change User ID	CHANGEUSRID	BENCOX
Define Source	DEFINESOURCE	MYBUNDLE
Define Time	DEFINETIME	07-Aug-2019 15:15:50
Install Agent	INSTALLAGENT	BUNDLE
Install Time	INSTALLTIME	07-Aug-2019 15:15:51
Install User ID	INSTALLUSRID	BENCOX
▼ Statistics		
Application Data Heap	HEAPAPPD	10,383,832
Completed Invokes	INVCOMP	0
Completed Invokes In Error	INVKERR	0
Current Heap	HEAPCUR	18,874,368
Current Invokes	INVKCUR	0
Current Peak Invokes	INVKPEAK	0
Max Heap	HEAPMAX	1,501,560,832
Runtime Heap	HEAPRUNT	2,957,312
Total CPU	CPUTOTAL	482

Attributes

## Developing a Node.js application for CICS

- Development of Node.js applications for CICS is, in general, like developing any other Node.js application
- Use common build and packaging techniques
  - look out for the rare package that uses natives and are not yet supported z/OS
- Node.js applications embedded in CICS can invoke programs in the rest of CICS
- This API is available in the [NPM repository](#)
  - npm install --save **ibm-cics-api**



# Getting started with CICS Explorer

- We have a new [6-part video series](#) to get you started with using CICS Explorer
- Just over half an hour takes you through
  - Downloading and installing
  - The basics of the interface
  - Getting connected
  - Using the main CICSplex Explorer view
  - Using resource views
  - Configuring resource views



IBM Developer



[CICS Developer Center](#) [About](#) [Blogs](#) [Podcasts](#) [Videos](#) [Samples](#) [Support](#)

## CICS Explorer getting started guide

CICS Explorer has plenty to offer to help you master managing CICS.

We've put together a short video series to help you get started.

[More about CICS Explorer](#)

[Download CICS Explorer](#)

### 1. Downloading and installing

Find out the quickest way to download and install IBM CICS Explorer on your local machine.



03:18

### 2. The basics of the interface

Discover the basic concepts of working with IBM CICS Explorer, which is based on the Eclipse IDE.

Once you've watched this video, you'll be familiar with the three major components of Eclipse (perspectives, views, and editors), and know how to use them to your advantage.



04:05

### 3. Getting connected

An introduction to the Host Connections view.

This view allows you to create new connections to CICS, assign credentials to connections, and define a default connection.



©2019 IBM Corporation

## Deploying to CICS

- Eventually, you will want to test the app in CICS
  - by creating a [CICS bundle](#), exporting it to zFS, and installing it into CICS
- Can deploy using CICS Explorer
  - but Node.js developers typically do not already have CICS Explorer
  - expect to perform deployment via the command line
- Use the Zowe CLI with the [zowe-cli-cics-deploy-plugin](#)



# Zowe CLI

- Zowe CLI offers a shell-based way of interacting with a z/OS machine, from your desktop
  - Use interactively or embed in shell scripts
  - Core capabilities around data sets, USS, jobs, TSO and console commands
  - Extensible by plugins
- `> zowe files dl ds "CICS.INFO.FILE"`
  - `> zowe uss issue ssh "ls"`
  - `> zowe jobs list jobs -o "*" -p "*"`
  - `> zowe tso issue cmd "LU"`
  - `> zowe console issue cmd "DISPLAY M"`




# Zowe CLI: CICS Deploy Plugin




- The [zowe-cli-cics-deploy-plugin](#) provides a way of deploying CICS bundles from the command line
  - Generate a CICS bundle from an existing workstation directory structure
  - Deploy a CICS bundle to a CPSM managed group of CICS regions
  - Undeploy a CICS bundle from a CPSM managed group of CICS regions
- Many more options than shown here
  - [github.com/IBM/zowe-cli-cics-deploy-plugin](https://github.com/IBM/zowe-cli-cics-deploy-plugin)
  - [npm.im/zowe-cli-cics-deploy-plugin](https://npm.im/zowe-cli-cics-deploy-plugin)
- > zowe cics-deploy generate bundle
- > zowe cics-deploy push bundle
  - name MYBUNDLE
  - target-directory /u/me/bundles





# Get started with Node.js on z/OS

- IBM Z Trial offers:
  - Many different products to evaluate
  - Available within a few hours
  - Configured for productive use
  - Guided scenarios
  - Absolutely free

 **IT Infrastructure**

Search   

IBM Z Hardware OS Software Resources 

 Chat Now

## IBM Z software trials

Try the latest IBM Z® software today at no charge, and with no installation required. These no-charge trials are available in two hours for three days. Register and get started today.

### Hands-on evaluation of IBM Z software is as easy as 1-2-3

1

**Register**  
Completely free. No credit card required.

2

**Get access**  
Trial will be available in 2 hours and accessible for 3 days including weekends

3

**Try it out**  
No installation required. Take a tutorial or just look around.

### Software trials


	CICS Transaction Server	IBM Information Management System	IBM SDK for Node.js – z/OS
<a href="#">Cloud/API integration</a>			
<a href="#">Enterprise DevOps</a>			
<a href="#">IT operations</a>			
<a href="#">Database and application server</a>			
<a href="#">Analytics and machine learning</a>			
	<p>Request this trial to gain access to a pre-configured, remote desktop environment where you can create a Java or Node.js web application and then deploy into IBM CICS Transaction Server.</p> <p>What's in the trial? Your trial environment includes hands-on tutorials that explain how to:</p> <ul style="list-style-type: none"><li>• Create and deploy a Java application into CICS</li><li>• Create and deploy a Node.js application into CICS</li></ul> <p>→ Register for the trial</p>	<p>The most secure, highest performing and lowest cost hierarchical database management software for online transaction processing (OLTP).</p> <p>Your trial includes hands-on tutorials that explain how to:</p> <ul style="list-style-type: none"><li>• Use IMS Enterprise Suite Explorer for Development to import and visualize IMS database and program definitions in IMS catalog.</li><li>• You will also create and execute SQL queries to access and manipulate IMS database.</li></ul> <p>→ Register for the trial</p>	<p>Modernize your applications and orchestrate your services with Node.js, the popular server-side JavaScript platform, now on z/OS. Develop and deploy Node.js on z/OS for better performance and security and connect to your z/OS assets.</p> <p>Your trial includes hands-on tutorials that explain how to:</p> <ul style="list-style-type: none"><li>• Deploy and test a Node.js LoopBack application on z/OS.</li><li>• Create and test a Node.js Express application accessing and managing a VSAM dataset on z/OS.</li></ul> <p>→ Register for the trial</p>

### You may also be interested in

**IBM Z DevOps tech zone**  
Application discovery capabilities and code analysis tools help you modernize apps for digital and hybrid cloud transformation.  
→ Explore IBM Z tech zone

**What's possible on Z?**  
Deliver hybrid cloud, smarter blockchain and agile DevOps – all with 8x more effective security than x86.  
→ Explore mainframe capabilities

**What can Z software do?**  
IBM Z software delivers faster insights and deployments with the resiliency and security of the mainframe.  
→ Explore featured software  
→ Find software A-Z

 ©2019 IBM Corporation

### Terms of use

By using this IBM offering, you accept the Terms and Conditions of the [IBM Cloud Services Agreement](#) for the IBM Z Software Trials. If you do not agree to these terms, do not access and use this IBM offering. Further details of the IBM Z Software Trials available as part of this IBM offering are described in the associated [Transaction Document](#).

**IBM CICS**  
Transaction Server for z/OS

# Java in CICS

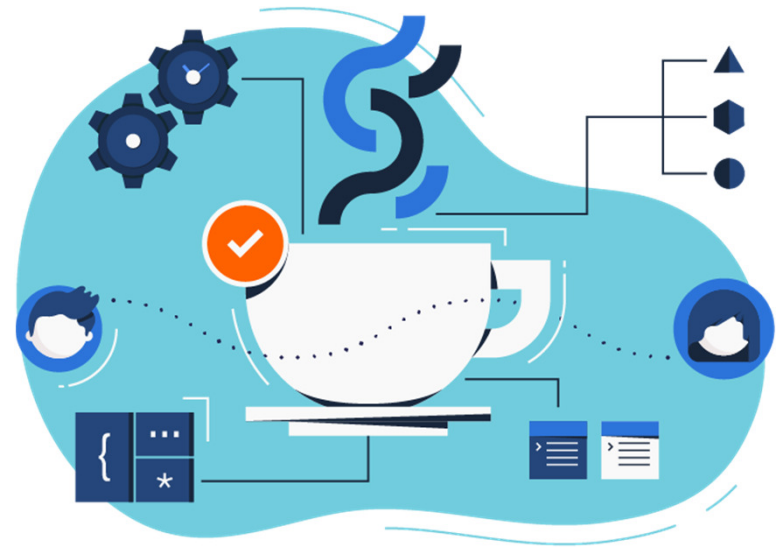


©2019 IBM Corporation



# Java EE 8 Full Platform application support

- CICS TS supports Java applications that are written to the Java Enterprise Edition 8 (Java EE 8) Full Platform specification
  - Using the embedded version of IBM WebSphere Liberty
  - APAR [PH15017](#) CICS TS V5.5
- Java applications that are hosted in CICS TS are integrated with CICS tasks by default
- A simple and powerful mechanism of modernizing CICS applications by using Java EE 8 features and capabilities





# Support for Jakarta EE 8 Platform

- The CICS Liberty JVM server supports now supports the Jakarta Enterprise Edition (EE) 8
- The [Jakarta EE 8](#) full platform technologies and specifications are an evolution of Java EE 8
  - Allows developers and applications to easily transition from Java EE to Jakarta EE
  - Available with APAR [PH19704](#)
- The promise of Jakarta EE is a community-driven open source model
  - More frequent releases than Java EE
  - Evolving more quickly to address the needs of modern applications



## Java support in CICS

- CICS provides the tools and runtime to develop and run Java applications in a [CICS JVM](#)
  - Java applications can interact with CICS services and applications written in other languages
  - You can develop applications using the IBM CICS SDK for Java, Maven modules, or Gradle modules
- The CICS JVM server
  - Eligible Java workloads can run on [specialty engine processors](#)
    - reducing the cost of transactions
  - Different types of work such as threadsafe Java programs and web services
  - Application life cycle can be managed in the OSGi framework
    - no need to cycle the JVM server
  - Java applications that are packaged using OSGi can be ported more easily between CICS and other platforms
  - [Java EE](#) applications can be deployed into the Liberty JVM server

# IBM CICS SDK for Java – The CICS Explorer

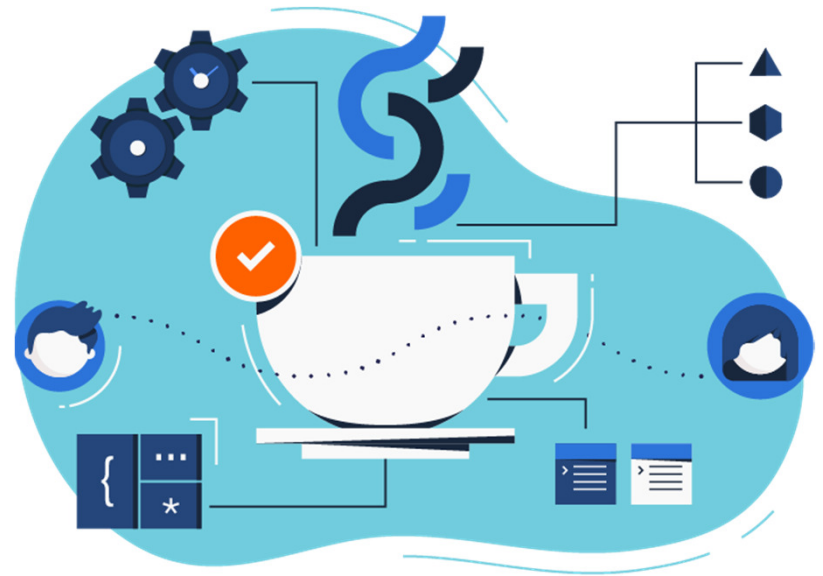
- The IBM CICS SDK for Java is included with the [CICS Explorer](#)
  - Provides support for developing and deploying applications that comply with the OSGi Service Platform specification
- The IBM CICS SDK for Java EE, Jakarta EE and Liberty is included as an option with the CICS Explorer
  - Supports packaging of Liberty applications into CICS bundles that can be deployed to CICS
- The OSGi Service Platform provides a mechanism for developing applications using a component model
  - Deploy applications into a framework as OSGi bundles
    - an OSGi bundle is the unit of deployment for an application component
      - contains version control information, dependencies, application code
- The IBM CICS SDK for Java allows development of Java applications for any supported release of CICS
  - The SDK includes the [Java CICS library \(JCICS\)](#) to access CICS services
  - [Examples](#) to get started with developing applications for CICS

# Using Maven or Gradle

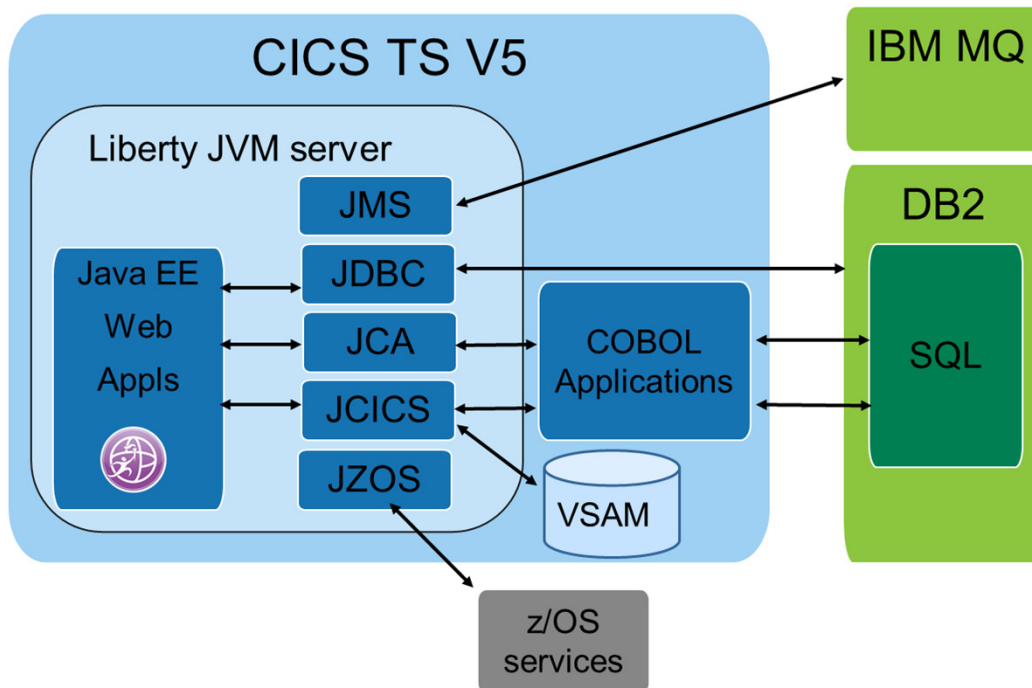
- You can use popular build tools such as [Maven](#) and [Gradle](#) to create your own scripts for building and deploying CICS Java programs
  - An alternative to the IBM CICS SDK for Java
  - Easy management of dependencies
    - Java developers can add the required versions of the Java CICS APIs and the CICS annotation processor to the Java dependencies
  - More freedom when choosing the [development environment](#)
    - Maven and Gradle support most Java IDEs
      - such as Eclipse, IntelliJ IDEA, and Visual Studio Code
  - Better integration into a build toolchain
    - Maven and Gradle integrate smoothly with other automation tools such as Jenkins and Travis CI
- The following artifacts are available on [Maven Central](#)
  - The Java CICS class library (JCICS)
    - Provides the EXEC CICS API support for Java applications in CICS TS
  - The CICS annotations library and the CICS annotation processor
    - Provides support that enables CICS programs to invoke Java applications in a Liberty JVM server
  - A bill of materials (BOM)
    - Defines the versions of the other artifacts to ensure that they are at the same CICS TS level

## The types of JVM server in CICS

- Aside from a couple of more niche use cases
  - JVM servers form these categories
    - [OSGi JVM server](#)
    - [Liberty JVM server](#)

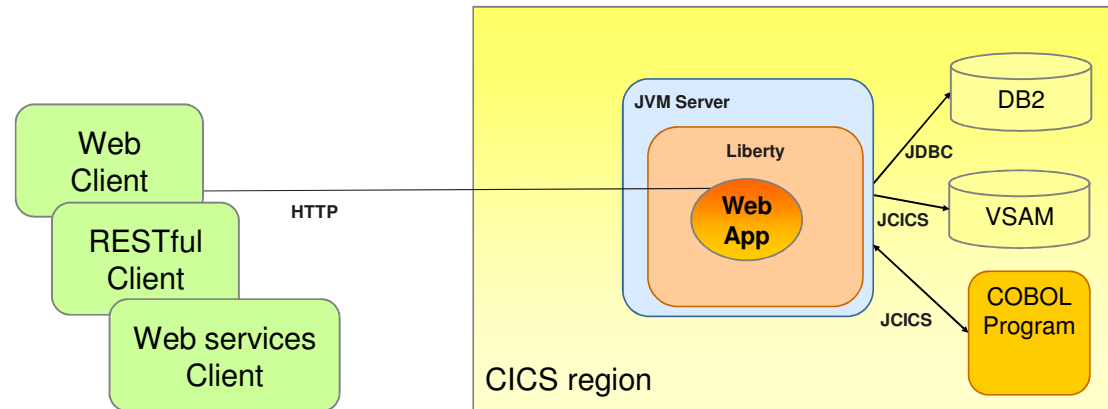


# Java EE 8 Full Platform in CICS



- ✓ The full **Java EE 8 profile** supported in an integrated Liberty JVM server
- ✓ **JMS** support for MQ in client mode
- ✓ **JDBC** and **SQLJ** for Db2 data sources and other relational databases
- ✓ **JCICS** to provide access to CICS API including linking to other CICS programs
- ✓ **JCA** local ECI adapter supports porting of CICS TG ECI applications into CICS
- ✓ **JZOS** provides access to z/OS services such as console, files

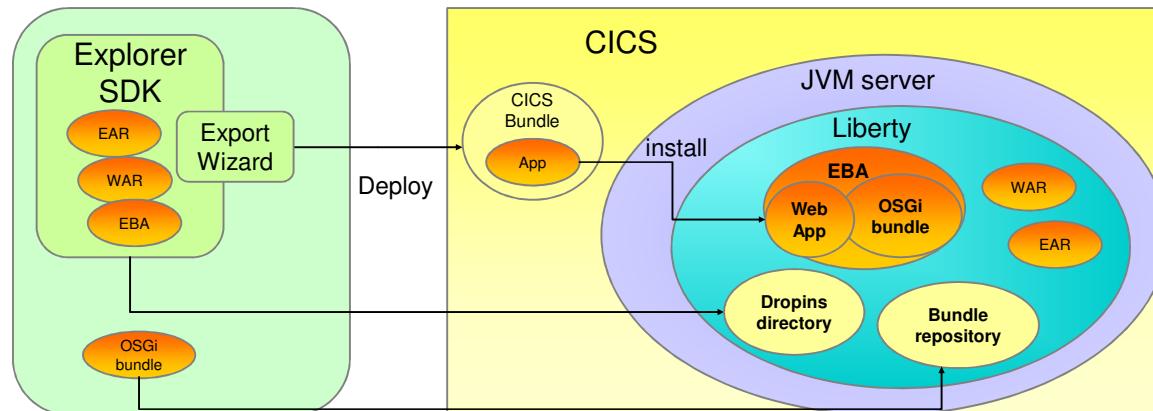
# Why use Liberty in CICS?



- **Porting web** application to z/OS
  - JEE8 Web profile and JCA local ECI - 'lift and shift' porting from other JEE servers
- New **integration logic** for existing CICS services
  - Restful services or SOAP Web services interfacing existing CICS components
- **Java business logic** in CICS
  - Access to DB2 data – JDBC, EJBs, JPA or VSAM/JCICS

# Application deployment – Liberty JVM server

- Liberty dropins directory
  - For development/testing
- CICS Bundle resource
  - Web application(WAR), JEE archive (EAR), OSGi Application(EBA)
  - CICS-managed application deployment
- Liberty application definition
  - For manual deployment via server.xml
- Liberty shared bundle repository or global library
  - For shared components





# CICS JVM profiles

## Include & share common configuration when cloning JVM servers across CICS regions

- For example unique ports, database configuration or log settings

`%INCLUDE=<file_path>`

## Append to variables are built up over multiple lines

`OSGI_BUNDLES=&CLONEDIR;/mybundle.jar`

`+OSGI_BUNDLES=/newpath/mybundle2.jar`

... is equivalent to ...

`OSGI_BUNDLES=&USSHOME;/&JVMSEVER;/bundles/mybundle.jar,/newpath/mybundle2.jar`

## Including server.xml snippets

- Inject Liberty configuration into server.xml
- A new JVM profile option `LIBERTY_INCLUDE_XML` is provided
  - to enable Liberty to load shared configuration
  - making it easier to administer, clone, and control Liberty JVM servers
- You can now use the `LIBERTY_INCLUDE_XML` property in JVM profiles
  - specify files that CICS will add `<include>` tags for
- In JVM profile  
`LIBERTY_INCLUDE_XML=<file>`

# Management

## CICS bundle status reflects Liberty application status

- CICS bundle with Web application bundle part remains in **ENABLING** state
  - until applications are installed in Liberty
- More robust application deployments

## Extended CICS JVM server message

**LOG\_LEVEL=**INFO | WARNING | ERROR | NONE

- New **dfhjvmlog zFS file** for CICS JVM server information, warnings and errors
- Can be redirected to MVS JES DD
- For example
  - a value of NONE suppresses all output
  - a value of WARNING gives log entries of warning level and above
  - the default value is INFO

# More than one Liberty JVM server per CICS region

## Multiple secure Liberty servers in a CICS region

- Provides improved application isolation or scalability without increasing number of regions
- Each Liberty server can have its own configuration and lifecycle – ideal for developers

## Wait for Liberty angel process \* (also in V5.4 [APAR PI92676](#))

`-Dcom.ibm.ws.zos.core.angelRequired=true`

- More robust CICS start-up and IPL procedures
- Ensure that a Liberty JVM server will connect to a Liberty angel process
  - before reaching the [ENABLED](#) state
- Integrates with named Liberty angel process `-Dcom.ibm.ws.zos.core.angelName`

# JSON Web Token

## Liberty JWT feature

- Programmatically parse, build and verify JWT tokens in Java applications
- Provides for authentication using digitally signed web tokens
- Also available on CICS TS V5.3 and 5.4 with APAR [PI91554](#)

## OpenID Connect Client feature

- Configure Liberty server to [authenticate](#) a request using a JWT token without writing any code
- Supports identity mapping
  - Map Subject in JWT to local registry user
  - Map distributed identity to SAF registry user via RACMAP



# Liberty Admin Center

- The Liberty Admin Center is now supported in CICS
- It is a site built into Liberty that allows you to
  - View and configure [server.xml](#) and related files
  - Examine applications running in the server
  - View [live statistics](#) about heap, CPU, and threads
- Available on CICS TS 5.5 with [PH08321](#)



The screenshot displays the Liberty Admin Center's 'Server Config' interface. At the top, there's a blue header with a server icon, the title 'Server Config', and user/refresh icons. Below the header, a dark bar shows 'server.xml' with 'Read only' and 'Close' buttons. The main area has tabs for 'Design' and 'Source', with 'Source' selected. A settings gear icon is on the right. The XML content is shown with line numbers 1 through 37. The XML defines a CICS Liberty profile with features like core, defaultApp, jsp, wab, security, and transportSecurity. It also includes an httpEndpoint and a configuration for monitoring application updates with a 5s interval and 'polled' trigger.

```
1: ?xml version="1.0" encoding="UTF-8"?><server description="CICS Liberty profile sample c
2:   <!-- Enable features -->
3:   <featureManager>
4:     <feature>cicsts:core-1.0</feature>
5:     <feature>cicsts:defaultApp-1.0</feature>
6:     <feature>jsp-2.3</feature>
7:     <feature>wab-1.0</feature>
8:     <feature>cicsts:security-1.0</feature>
9:     <feature>transportSecurity-1.0</feature>
10:  </featureManager>
11:  <!-- Default HTTP End Point -->
12:  <httpEndpoint host="*" httpPort="26006" httpsPort="26007" id="defaultHttpEndpoint"/>
13:  <!-- CICS Bundle Installed Applications -->
14:  <include location="{server.output.dir}/installedApps.xml"/>
15:  <!-- The following configuration controls how often server.xml
16:       is scanned for updates. The default is every 500ms which may
17:       cause excessive I/O and CPU cost on z/OS.
18:       The values shown below reduce the overhead while still
19:       providing a relatively timely detection of new applications
20:       that have been installed/removed via a CICS Bundle
21:       (WAR bundlepart). If you use CICS bundles to install Web
22:       Applications (WAR files) do not disable the polling.
23:  -->
24:  <config monitorInterval="5s" updateTrigger="polled"/>
25:  <!-- Further scanning is performed to detect application updates or
26:       addition/removal of applications to the dropins directory. If
27:       you are using CICS Bundles as the vehicle for Application
28:       deployment you should disable the dropins directory.
29:       Further I/O and CPU reduction can be achieved by disabling
30:       the application scan. To effect changes to your applications
31:       while the server is still running, you should disable and
32:       re-enable the CICS bundle that contains the Web application.
33:       The pollingRate is only applicable when the updateTrigger is
34:       set to the 'polled' value.
35:       Consult the WebSphere Application Server Liberty Profile
36:       documentation for further information on these parameters.
37:  -->
```



# Java 8 Required for CICS TS V5

**CICS TS V5.5 requires IBM 64-bit Java SDK for z/OS, V8**

**IBM 64-bit SDK for z/OS, Java Technology Edition V7.0.0 and V7.1.0**

- To be withdrawn from service on September 30, 2019
- [Withdrawal Announcement 916-121 – August 2, 2016](#)

**WebSphere Application Server Liberty**

- [WebSphere Application Server Liberty base – Removal Notices](#)
- From September 2019, Liberty with fix pack 19.0.0.9 onwards will only run with Java 8 SDK

**For details see article [Java 8 recommended for CICS TS V5](#)**



# Get started with Java in CICS

- If you have not yet configured CICS for Java, or have other impediments to trying out Java
  - use IBM Z Trial at [ibm.biz/ibmztrial](https://ibm.biz/ibmztrial)
- CICS Transaction Server trial
  - Create and deploy a Java application into CICS

**IBM** IT Infrastructure

Search

IBM Z Hardware OS Software Resources

**IBM Z software trials**

Try the latest IBM Z® software today at no charge, and with no installation required. These no-charge trials are available in two hours for three days. Register and get started today.

**1** Hands-on evaluation of IBM Z software is as easy as 1-2-3

**Register**  
Completely free. No credit card required.

**2** Get access  
Trial will be available in 2 hours and accessible for 3 days including weekends

**3** Try it out  
No installation required. Take a tutorial or just look around.

**Software trials**

Cloud/API integration  
Enterprise DevOps  
IT operations  
**Database and application server**  
Analytics and machine learning

**CICS Transaction Server**  
Request this trial to gain access to a pre-configured, remote desktop environment where you can create a Java or Node.js web application and then deploy into IBM CICS Transaction Server.  
What's in the trial?  
Your trial environment includes hands-on tutorials that explain how to:  
• Create and deploy a Java application into CICS  
• Create and deploy a Node.js application into CICS  
→ Register for the trial

**IBM Information Management System**  
The most secure, highest performing and lowest cost hierarchical database management software for online transaction processing (OLTP).  
Your trial includes hands-on tutorials that explain how to:  
• Use IMS Enterprise Suite Explorer for Development to import and visualize IMS database and program definitions in IMS catalog.  
• You will also create and execute SQL queries to access and manipulate IMS database.  
→ Register for the trial

**IBM SDK for Node.js - z/OS**  
Modernize your applications and orchestrate your services with Node.js, the popular server-side JavaScript platform, now on z/OS. Develop and deploy Node.js on z/OS for better performance and security and connect to your z/OS assets.  
Your trial includes hands-on tutorials that explain how to:  
• Deploy and test a Node.js LoopBack application on z/OS.  
• Create and test a Node.js Express application accessing and managing a VSAM dataset on z/OS.  
→ Register for the trial

**You may also be interested in**

**IBM Z DevOps tech zone**  
Application discovery capabilities and code analysis tools help you modernize apps for digital and hybrid cloud transformation.  
→ Explore IBM Z tech zone

**What's possible on Z?**  
Deliver hybrid cloud, smarter blockchain and agile DevOps – all with 8x more effective security than x86.  
→ Explore mainframe capabilities

**What can Z software do?**  
IBM Z software delivers faster insights and deployments with the resiliency and security of the mainframe.  
→ Explore featured software  
→ Find software A-Z

**Terms of use**  
By using this IBM offering, you accept the Terms and Conditions of the IBM Cloud Services Agreement for the IBM Z Software Trials. If you do not agree to these terms, do not access and use this IBM offering. Further details of the IBM Z Software Trials available as part of this IBM offering are described in the associated Transaction Document.



©2019 IBM Corporation



# Get started with Java on CICS

- We have video series aimed at getting started with Java on CICS
  - [Developing a RESTful Web application for Liberty in CICS](#)
  - [Architecting Java solutions for CICS](#)
  - [Extending a CICS web application using JCICS](#)



# Keeping up to date

- For all the latest developments subscribe to the blogs at [developer.ibm.com/cics](https://developer.ibm.com/cics)

The screenshot shows the IBM CICS Developer Center website. The header includes the IBM logo and navigation links: CICS Developer Center, About, Blogs, Podcasts, Videos, Samples, and Support. The main banner features the IBM CICS logo and the text "Welcome to the CICS Developer Center". Below this, a paragraph describes CICS as a family of mixed language application servers. Three buttons are visible: "Try CICS!", "Download CICS Explorer", and "Read CICS Papers".

**What's new**

**Introducing CICS Bundle Maven plug-in version 0.0.1**  
by Stewart Francis · on July 31, 2019 · in CICS Explorer, CICS TS, DevOps, Java, Other  
If you're a member of our design partnership as part of the Z design forum, you'll know we're hard at work on improvements for CICS TS application developers. The first thing that's come out of that initiative is our brand new Maven plug-in, designed to stream-line the process of authoring CICS bundles for Java applications....  
[Continue reading](#)

**Avoiding HTTP outages by managing Liberty HTTP endpoints**  
by Ephan · on July 8, 2019 · in Java, Liberty, Policies  
This blog describes a solution to avoid HTTP 404 errors when a CICS region starts up due to Liberty accepting HTTP requests before an application is ready.  
[Continue reading](#)

**CICS CM adds to new capabilities**  
by SatehTanna · on June 3, 2019 · in CICS Tools, CICS TS, DevOps  
CICS Configuration Manager is the premier configuration tool for CICS Transaction Server. The current version is CICS CM V5.4 and this has been enhanced via PTF for APAR PHO9609. As more CICS sites start to exploit the rich functions in CICS Configuration Manager V5.4, new enhancements are being requested. As a result of Request for...  
[Continue reading](#)

**All new Node.js in CICS Z Trial now available!**  
by Natasha McKenzie-Kelly · on April 26, 2019 · in Node.js  
This Z Trial takes you through a 30 minute scenario where you can try using Node.js in CICS. It will take you through the steps to package a sample Node.js web application into a CICS bundle and deploy it into IBM CICS Transaction Server.  
[Continue reading](#)

**New enhancements to CICS Performance Analyzer v540**  
by SatehTanna · on March 29, 2019 · in CICS Tools, CICS TS, Other, Performance  
CICS Performance Analyzer is the premier performance reporting and analysis tool for CICS Transaction Server. The current version is CICS PA V5.4 and this has been enhanced via PTF for APAR PHO8968. The following enhancements are introduced by CICS PA V540 APAR PHO8968 RFE 116861. Prior to this enhancement, statistical values for Form based Performance...  
[Continue reading](#)

**Managing enterprise-wide deployment of CICS Explorer**  
by DaveN · on March 25, 2019 · in CICS Explorer  
One of the most common questions after system administrators experience CICS Explorer is "what is the best way to deploy this to my

**Hot topics**

**CICS Explorer Enhancements**  
IBM CICS Explorer has a host of new capabilities, making it easier than ever to manage your CICS environments.

**Node.js in CICS**  
Taking application serving to the next level with this light-weight, efficient, language. Learn about how CICS is Node.js ready.



©2019 IBM Corporation

# CICS and Java Wildfire Workshop (CICSJAV1)

- This 2.5 day Wildfire workshop details CICS Java support
  - Including development, debugging, diagnostic, and tuning information
  - Upon completion of this workshop, you should have a better understanding of how to use Java in your CICS environment
  - This workshop will benefit Application Programmers, Systems Programmers, and Enterprise Architects
- Development of new Java applications is simplified
  - Due the mixed language environment of CICS and accessibility to your CICS resources
  - You can take a pure Java approach, or utilize CICS's many aids such as JSON parsing, Web Services support, or access to VSAM files and DB2
- Support for the Liberty Profile running in CICS
  - Offers the ideal integration of Java
  - Java specifications are supported
    - Java Servlets, JavaServer Pages, JAX-WS, and JAX-RS, while fully integrating into the CICS environment
- Lectures
  - CICS API for Java (the equivalent of the EXEC CICS API)
  - CICS and OSGi
  - CICS and the Liberty Profile
  - CICS and DB2 for Java
  - CICS/Java Application Tuning and Debugging
- Lab Exercises
  - The CICS Explorer
  - Writing a simple Java application in CICS
  - Using Servlets and JSPs in CICS
  - Writing a JAX-RS Java application for CICS
  - Writing a JAX-WS Java application for CICS
  - Adding a REST/JSON Interface to a Java program using z/OS Connect in CICS
  - Enabling CICS APIs using z/OS Connect EE



