Introduction and Overview
Topics to be Discussed

Links to Pages

🔗 • Setting Context
   Background on the API Economy, REST, and other factors that led to the development of z/OS Connect EE V2.0

🔗 • Product Overview
   A tour of what it is, how it works, and the value it brings

🔗 • Getting Started
   Details on prerequisites and how to acquire and use

🔗 • Additional Resources
   Reference on other sources of information
Setting Context

Background on z/OS Connect EE V2.0
Evolution of Mainframe Integration Patterns

- **Service Oriented**
  - Loose coupling using network-based protocols such as SOAP/WSDL

- **Web Access**
  - Screen-scraping of 3270 applications

- **Messaging**
  - Using technologies such as IBM MQ

- **Tight Integration**
  - All components located on mainframe, and linkages are tightly coupled

- **API Economy**
  - Integration based on a set of well-understood and easily-accessible APIs, increasingly based on REST/JSON patterns

*This is where z/OS Connect fits into the picture*

These earlier patterns are still in use and in many cases are the best pattern for the application. Just because they came earlier, does not mean they are now obsolete.
**Simplified Overview of REST/JSON**

**REST** - “Representational State Transfer” ... which uses HTTP and HTTP verbs to allow a client to interact with a server over the TCP/IP network.

**JSON** - “JavaScript Object Notation” ... a name/value pair representation of data that is relatively lightweight and generally simpler to handle and parse than XML.

**REST is increasingly popular as an integration pattern because it is stateless, relatively lightweight, is relatively easy to program to, and operates well with discovery mechanisms such as IBM’s API Management product.**
1. z/OS Connect a “feature” of Liberty
2. “Service Provider” = backend connectivity
3. “Interceptors” = configurable function
4. Extensible interface = flexibility

Such as a WAS z/OS server hosting REST

A long-running task using WOLA APIs to host a service
V1: Good Start ... Known Areas for Improvement

Confusing Entitlement / Delivery

z/OS Connect was a no-charge feature entitled under with WAS z/OS, CICS, IMS or DB2 license. Multi-backend usage was different depending on entitlement. Acquisition and installation was different depending on entitlement.

Summary: entitlement and delivery model created some confusion

Inconsistent Tooling Across Backends

The tooling support was a function of the entitlement option used to acquire z/OS Connect V1. IMS and DB2 provided an Eclipse based tooling environment. CICS employed CICS Explorer, but not to the degree IMS and DB2 integrated z/OS Connect with tooling. WAS z/OS relied on manual service definitions.

Summary: consistent tooling preferred

First Generation REST

z/OS Connect V1.0 REST implementation was essentially an RPC model. Query parameters were not accessible. Header information not accessible. HTTP verb usage pattern not well-formed and consistent.

Summary: more complete REST implementation model was needed
Version 1.0 available in 2H2014

Version 1.0 enhancements in 2H2015
- File-based Logging
- Proxy support
- Configurable options for SSL and authentication

IBM z/OS Connect
Enterprise Edition V2.0
Available December 2015

Functional improvements from V1 to V2:
- More advanced REST functionality
- More advanced API tooling and management

Focused IBM Development and Support
- Dedicated team rather than separate teams

Functional stability for the z/OS Connect feature*

* Except where IBM has issued a statement of direction for V1.0. Go to this page for more.
As the number of APIs grows, the need to systematically manage the APIs becomes apparent. API Management provides a way to improve business value of APIs created.

The relationship is complementary -- not required by z/OS Connect or other API providers, but *very helpful* in an API environment.
IBM Bluemix is an open-standards, cloud platform for building, running, and managing applications. With Bluemix, developers can focus on building excellent user experiences with flexible compute options, choice of DevOps tooling, and a powerful set of IBM and third-party APIs and services.

IBM Bluemix Cloud Environment

Virtual Runtime

Virtual Runtime

Virtual Runtime

Virtual Runtime

Virtual Runtime

IBM API Management

z/OS Connect EE V2.0

REST/JSON is a good integration pattern for a cloud environment such as Bluemix.

That’s the relationship -- if you have Bluemix environments that need access to z/OS data, then z/OS Connect provides a good way to expose REST APIs to backend z/OS data.

The relationship is complementary.
Product Overview
Understanding z/OS Connect EE V2.0
V1.0 Functions Carried Forward into V2.0

What you knew about z/OS Connect V1.0 is still useful when approaching z/OS Connect EE 2.0.

Similar concepts, but with key enhancements as you’ll see.
High-Level Overview of z/OS Connect EE V2.0

Runtime Server
1
• Runs on Liberty z/OS
• Hosts APIs you define to run in it
• Connects with backend system
• Liberty + z/OS Connect = “instance”
• You may have multiple instances

Liberty z/OS

Backend Systems (CICS, IMS, DB, etc.)
• IBM z/OS 2.1 or higher
• IBM 64-bit SDK for z/OS, Java Technology Edition V7.1.0 or V8.0.0

Tooling Platform
2
• Integrates with an Eclipse environment
• Define APIs
• Define data mapping
• Deploy APIs to runtime server
• Export API archive for other tools to deploy

Eclipse
• IBM CICS Explorer V5.3
• IBM IMS Explorer for Development V3.2
• IBM Explorer for z/OS Aqua V3.0

Eclipse
• Liberty z/OS
• Backend Systems (CICS, IMS, DB, etc.)
Comparison of REST Support V2.0 vs. V1

z/OS Connect V1.0:

**POST** /account/create  +  (JSON with account create information)

**POST** /account/balance  +  (JSON with account number)

**POST** /account/update  +  (JSON with account number and deposit)

This corresponds to the lower level we showed earlier. It’s very basic. It may be “good enough” for some use-cases, but it falls short of what many developers seek when creating REST APIs.

z/OS Connect V2.0:

**POST** /account?name=Fred  +  (JSON with Fred’s information)

**GET** /account?number=1234

**PUT** /account?number=1234  +  (JSON with dollar amount of deposit)

HTTP Verb conveys the method against the resources; i.e., POST is for create, GET is for balance, etc.

URI conveys the resource to be acted upon; i.e., Fred’s account with number 1234

The JSON body carries the specific data for the action (verb) against the resource (URI)

More aligned with developer requirements for REST APIs
Eclipse-based Tooling for z/OS Connect EE V2.0

- Eclipse project view, which is familiar to developers who have used Eclipse tooling for other development projects
- API projects can be exported and imported for portability between developers
- Assign API function based on HTTP verb
- Access query parameters from the URI
- Provide data mapping definitions to the service

API definitions are created through the tool, which is consistent across backend systems (CICS, IMS, etc.)
Request Mapping Capabilities

The API mapping model adds a powerful abstraction layer between the API consumer and the underlying z/OS assets.

- Mapping of HTTP headers, path parameters (URI templates), and query parameters to the fields in the request message JSON body.
- Pass-through, redaction, or defaulting of fields in the request or response message JSON body.
- Mapping and defaulting of HTTP headers in the HTTP response message.
API Archive (AAR) -- API Packaging

z/OS Connect V1.0

Service definitions in server configuration file (or in related side files)

Discovery function returned JSON with services, but:

- Not Swagger definition
- Only service URIs, but did not contain information about connectivity to backend

Portability of service limited:

- No good “export” of service from a hosting server
- Import into tooling largely a manual process

API Archive (AAR) File

- ZIP-format file
- Contains Swagger documentation of service
- Contains JSON schema and API information
- Produced by tooling
- Exportable to server runtime | Consumable by tooling

Provides a standardized method for defining, transporting and deploying services

More flexibility and greater productivity
Better model from V1.0’s service definitions
Because the service definitions have been encapsulated into a deployable unit, it becomes eligible for deployment by automated tools. This further enhances productivity.
The inclusion of Swagger 2.0 support in z/OS Connect EE V2.0 makes exchange of API information standardized, which provides compatibility with a wider set of devices and functions.

* An emerging accepted industry standard, but not an official open standard
z/OS Connect EE V2.0 instances can be duplicate

- On same LPAR
- Across LPARs

Because REST is *stateless*, network routing functions can be placed in front of duplicated instances and balance traffic.
Example 1 - Multiple Backend Systems

z/OS Connect EE V2.0 becomes the REST API entry point to the LPAR for access to several backend systems where the data program resides.

Duplicate the z/OS Connect EE V2.0 instance for greater availability and/or greater throughput.
Example 2 - Multiple Lines of Business

There are times where the business requires operational separation and isolation. In that case, multiple instances of z/OS Connect EE V2.0 applies.
IBM makes the following statements of general direction:

- IBM intends to deliver IBM z/OS Connect Enterprise Edition (EE) components and technologies through continuous delivery of new features in the coming months.
- IBM intends that a future release of IBM CICS Transaction Server for z/OS (CICS TS) will provide support for z/OS Connect EE to enable it to execute embedded within CICS TS.
- IBM intends that a future release of IBM MQ for z/OS will provide support for both z/OS Connect and z/OS Connect EE.
- IBM intends to update IBM System Automation for z/OS V3.5.0 to deliver a new sample policy to allow automated operations and restart of z/OS Connect and z/OS Connect EE.
- IBM intends that a future release of IBM IMS Enterprise Suite will provide support for z/OS Connect EE.
- IBM intends to offer IBM DB2 for z/OS Version 11, or later, with support for the external interface delivered in z/OS Connect EE V2.0, and DB2 RESTful API support that is fully integrated into the DB2 for z/OS Distributed Data Facility.

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remain at our sole discretion.
Summary

z/OS Connect is a mechanism that provides a REST interface platform to z/OS as a Systems of Record.

z/OS Connect V1 was a no-charge feature; it was adequate for some use-bases but for others key functional enhancement were needed.

z/OS Connect EE V2.0 is a separately orderable product from IBM that provides those enhancements:

- More sophisticated handling of REST URIs patterns
- More sophisticated workstation tooling for API creation
- Inclusion of Swagger 2.0 for wider publication of API descriptions
- APIs as exportable artifacts for better deployment management
Getting Started
How to acquire and start the journey
How to Order z/OS Connect EE V2.0

Liberty z/OS

- z/OS Connect EE V2.0 Runtime
  - Product Number: 5655-CEE
  - Service and Support: 5655-CES

Eclipse

- z/OS Connect EE V2.0 Workstation Tooling
  - Under the product offerings section, select "z/OS Connect".

Prerequisite information on the next page
Hardware and Software Prerequisites

For connectivity to backend systems:

- **CICS -- CICS TS 4.1 or higher**
  This is based on the minimum level for the WebSphere Optimized Local Adapter (WOLA) support. For WOLA and Liberty, the minimum CICS TS level is 4.1.

- **IMS -- IMS 12.1 or IMS 13.1**
  See IMS Mobile Feature Pack component of IBM IMS Enterprise Suite for z/OS, V3.1.1 announcement letter 214-220, which details the software requirements. The z/OS Connect EE V2.0 requirements for IMS are the same as are outlined in that announcement letter.
Additional Resources
Pointers to more on z/OS Connect EE V2.0
Resources for Additional Learning

z/OS Connect EE V2.0 announcement letter

z/OS Connect EE V2.0 web page

z/OS Connect EE V2.0 Knowledge Center

z/OS Connect EE V2.0 API Editor Knowledge Center Article

API Management web page

z/OS Connect EE V2.0 Techdoc Page

IBM z/OS Connect Enterprise Edition Performance Reports

IBM z Systems: The Heart of the Mobile and API Economy
Document Change History

This document’s version Date: *March 7, 2016*

Change History:

• December 3, 2015 -- original publication

• December 8, 2015 -- updated to change “SAR” (service archive) to “AAR” (API archive)

• December 10, 2015 -- updated to better clarify what HTTP methods V1.0 supported

• December 16, 2015 -- updated with link to Knowledge Center

• January 6, 2016 -- removed statement of dynamic update when AAR deployed; with initial release a server restart is required when an API package is deployed.

• January 21, 2016 -- added a few more links to the “Resources for Additional Learning” page

• March 7, 2016 -- updated to reflect z/OS 1.13 as an acceptable level of z/OS for z/OS Connect EE V2.0