z/VSE Connectors
Best practices and use cases

Ingo Franzki
ifranzki@de.ibm.com
z/VSE Connectors - Introduction

— The z/VSE Connectors started as a single function in VSE/ESA 2.5
— Since then, it evolved over time
— Its now more a technology than just a function
  • consisting of many different features and functions
  • supporting various connector solutions
z/VSE Connector & Tools - Downloads

— **Connector components**
  
  • Delivered as part of the z/VSE system as WBOOKs in PRD2.PROD
  
  • Also available on the z/VSE Homepage
    https://www.ibm.com/it-infrastructure/z/zvse-downloads

— Many ‘as-is’ tools are provided for download

— Information about the Connector Components can be found here:
  https://www.ibm.com/support/knowledgecenter/SSB27H_6.2.0/fa2ws_connectionPossibilites_use_.html
Scenario 1: Web- / Mobile-enabling of Applications

→ Web-enable z/VSE Applications
→ Mobile-enable z/VSE Applications
→ Provide RESTful APIs for z/VSE Applications (microservices)
→ Modernize User Interface for applications
Scenario 2: Central Database

→ Use a central database for all business related applications
→ Allow z/VSE applications to work with central database
→ Add analytics and business intelligence
Scenario 3: Modern z/VSE Administration, Operation, Automation

→ Use graphical Tools for z/VSE administration
→ Add cross-system automation

- VSE Navigator
- CICS Explorer
- VSE Health Checker
- Keyman/VSE
- Automation cross system
- z/VSE Batch
- Transaction Servers (e.g. CICS TS)
- VSE Connector Server
- TCP/IP
- Data (e.g. VSAM, DL/I)
**Java-based Connector**

- **Remote access to z/VSE data and programs from a Java program**
  - Real time access to VSAM, DL/1, LIBR, POWER, Console, Jobs, ...
  - From standalone Java programs
  - From web/mobile applications (servlets, JSPs, Mobile Apps etc.)
  - Deployable as J2C Resource Adapter or JDBC Data Source into J2EE web applications servers, such as:
    - WebSphere Application Server
    - IBM Mobile First Server
VSAM Redirector

— Synchronization of VSAM data with a database

— Real-time: VSAM Redirector Client/Server
  - Data synchronization (OWNER=VSAM):
    - any INSERT, UPDATE or DELETE request is immediately replicated into the database
    - Read requests go against the VSAM dataset on z/VSE
  - Data migration (OWNER=REDIRECTOR):
    - All VSAM requests are sent to the database.
    - No access to the VSAM dataset anymore (except OPEN / CLOSE).

— Near real-time: VSAM Capture Exit
  - Data changes are collected in a delta file or MQ queue
  - Delta file is downloaded and applied to database from time to time
**VSAM Redirector - modes of operation**

**Data Migration**
- EXIT=IESREDIR, OWNER=REDIR
  - Real time access to data in database
  - All VSAM requests are redirected to the database
  - High performance impact

**Data Synchronization**
- EXIT=IESREDIR, OWNER=VSAM
  - Real time data replication (one way)
  - Only updating VSAM requests (update, insert, delete) are redirected to database
  - Medium performance impact

**Capture & Apply**
- EXIT=IESVSCAP
  - Near real time data replication (every n minute, etc.)
  - Changes to VSAM data are captured and collected, and then applied to the database asynchronously
  - Low performance impact
z/VSE Database Call Level Interface (DBCLI)

— Allows z/VSE applications to access a relational database on any suitable database server
  • IBM DB2, IBM Informix, Oracle, MS SQL Server, MySQL, etc.
    → The database product must provide a JDBC driver that supports JDBC V3.0 or later
— Utilize advanced database functions and use SQL statements provided by modern database products
z/VSE applications accessing Databases

Scenario 2: Central Database

DBCLI & DRDA with DB2 Federation

MS SQL Server

Oracle

MySQL

And others ...

CICS or batch Application

DBCLI
z/VSE 6.2: Interactive Query Tool for DBCLI

NEW!


**z/VSE 6.2: Batch Query Tool for DBCLI**

```
* $$ JOB JNM=RUNDBCLI,DISP=D,CLASS=4
* $$ LST DISP=D,CLASS=Q,PRI=3
// JOB RUNDBCLI
// LIBDEF *
,SEARCH=(PRD2.CONFIG,PRD1.BASE,PRD2.TCPIPC)
// EXEC IESDBCLB,PARM='SYMBOLS=YES ECHO=ON'
1554I PHASE IESDBCLB IS TO BE FETCHED FROM PRD1.BASE
DBCLI BATCH QUERY TOOL
CONNECT SERVER= my.database.server.com DBNAME=SAMPLE
USER=db2user PASSWORD=(PASSWORD SUPPRESSED);
INFO: CONNECTED TO SERVER 'my.database.server.com' DBNAME 'SAMPLE'.
INFO: DATABASE PRODUCT 'DB2/NT' VERSION 'SQL09075'
INFO: LAST RC=0
SELECT EMPNO,FIRSTNAME,LASTNAME,SALARY,BONUS
FROM EMPLOYEE;
DISCONNECT; /*
/&
```

```
// JOB RUNDBCLI
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE,PRD2.TCPIPC)
// EXEC IESDBCLB,PARM='SYMBOLS=YES ECHO=ON'
1554I PHASE IESDBCLB IS TO BE FETCHED FROM PRD1.BASE
DBCLI BATCH QUERY TOOL
CONNECT SERVER= my.database.server.com DBNAME=SAMPLE
USER=db2user PASSWORD=(PASSWORD SUPPRESSED);
INFO: CONNECTED TO SERVER 'my.database.server.com' DBNAME 'SAMPLE'.
INFO: DATABASE PRODUCT 'DB2/NT' VERSION 'SQL09075'
INFO: LAST RC=0
SELECT EMPNO,FIRSTNAME,LASTNAME,SALARY,BONUS
FROM EMPLOYEE;
DISCONNECT;
```

```
EMPNO       FIRSTNAME     LASTNAME        SALARY       BONUS
----------- --------------- --------------- ------------ ------------
000010      CHRISTINE    HAAS               152750.00      1000.00
000020      MICHAEL      THOMPSON            94250.00       800.00
000030      SALLY        KWAN                98250.00       800.00
...
INFO: ROWCOUNT: 42
INFO: LAST RC=0
DISCONNECT;
INFO: DISCONNECT SUCCESSFULL.
INFO: LAST RC=0
```

```
1555I LAST RETURN CODE WAS 0000
EOJ RUN     MAX.RETURN CODE=0000
```
**z/VSE Script Connector**

- **z/VSE Script Connector can be used to:**
  - Access z/VSE resources from distributed non-Java applications
  - Call/Execute processes on distributed systems from z/VSE applications or Jobs
  - Cross-System automation

![Diagram of z/VSE Script Connector](image)
CICS Connectivity

— CICS Web Support is the base of CICS connectivity
SOAP / Web Services support

— Web Service-enable z/VSE CICS TS applications
  • Provide existing CICS applications as Web Service to the outside world
    o z/VSE as the SOAP server
  • Use/call external Web Services from within z/VSE CICS applications
    o z/VSE as the SOAP client
  • CICS2WS Tool is used to generate proxy code or mapping rules

— z/VSE 6.2:
  • z/VSE SOAP Engine now supports Channels & Containers
z/VSE 6.2: RESTful Web Services support

— Use REST (Representational State Transfer) with CICS applications
  • Provide existing CICS applications as RESTful Web Service to the outside world
    o z/VSE as the REST server
    o Provide an easy to use RESTful API to services for z/VSE services
  • Use/call external RESTful Web Services from within z/VSE CICS applications
    o z/VSE as the REST client
    o Use external RESTful APIs within z/VSE applications
  • Payload can be:
    o JSON (JavaScript Object Notation)
    o XML
    o Plain text, Binary, Form fields, Multipart
What is REST (Representational State Transfer)?

— Representational State Transfer (REST) is a **software architecture style consisting of guidelines and best practices** for creating web services

— REST has gained widespread acceptance across the web as a **simpler alternative to SOAP** and WSDL-based web services

— RESTful systems typically communicate over the **Hypertext Transfer Protocol (HTTP)**
  • with the same HTTP verbs (GET, POST, PUT, DELETE, and so on) used by web browsers

— The **payload** (message) transported by RESTful web services can be of various types (content types)
  • Commonly used is **JSON** as well as **XML**, but it can also be plain text, or even binary data
What is REST (Representational State Transfer)?

— A RESTful web service typically operates on a certain ‘object’ on a server
  • The object is typically addressed through the URI (part of the URL)
    o http://host:port/resource-uri

— Actions on such resources are typically denoted by the HTTP request types:
  • **GET** would typically **read** the resource
  • **PUT** would typically **update/replace** the resource
  • **POST** would typically **create** the resource
  • **DELETE** would typically **delete** the resource

— Additional parameters can be supplied via the URL query string
  • http://host:port/resource-uri?query-string
What is REST (Representational State Transfer)?

— RESTful web services are typically **stateless**
  - Each request from any client contains all the information necessary to service the request
  - The session state is therefore held in the client

— RESTful web services may use **HTTP specific features**
  - **HTTP headers** – to transport additional attributes
  - **Cookies** – to manage state information between requests

→ As denoted by the term 'typically' in above descriptions, there is no hard requirement for any of the described properties
HTTP basics

Request:

```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

Response:

```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```
### Example: A RESTful web service

Request a list of books:

**Request:**
GET /api/v1/books

**Response:**
```json
{
  "meta": {},
  "data": [
    {
      "id": 24,
      "title": 'Behavior-Driven Development',
      "author": 'Viktor Farcic'
    },
    {
      "id": 25,
      "title": 'Continuous Integration',
      "author": 'Viktor Farcic'
    }
  ]
}
```

Request a single book:

**Request:**
GET /api/v1/books/id/24

**Response:**
```json
{
  "meta": {},
  "data": {
    "id": 24,
    "title": 'Behavior-Driven Development',
    "author": 'Viktor Farcic'
  }
}
```

Create a book:

**Request:**
POST /api/v1/books/id/24

```json
{
  "id": 24,
  "title": 'Behavior-Driven Development',
  "author": 'Viktor Farcic'
}
```

**Response:**

Status: 201 Created
```json
{
  "meta": {},
  "data": {
    "uri": /api/v1/books/id/24
  }
}
```

Delete a book:

**Request:**
DELETE /api/v1/books

```json
{
  "id": 24,
}
```

**Response:**

Status: 202 Accepted
```json
{
  "meta": {},
  "data": {}
}
```

---

Example taken from: https://technologyconversations.com/2014/08/12/rest-api-with-json/
**z/VSE 6.2: z/VSE as a REST Server**

Description of the REST Parameter Block:
[https://www.ibm.com/support/knowledgecenter/SSB27H_6.2.0/fa2ws_how_rest_control_blocks_are_used.html](https://www.ibm.com/support/knowledgecenter/SSB27H_6.2.0/fa2ws_how_rest_control_blocks_are_used.html)
z/VSE 6.2: z/VSE as a REST Server

— The z/VSE REST-Engine...

- Receives the request (via CICS Web Support)
- Extracts information from the request:
  - User program to call from the URL:
    http://host:port/cics/CWBA/IESRESTS/user-program/resource-uri?[query-string]
  - URL parameters from the query string (if any)
  - HTTP headers
  - Cookies (if any)
  - Request data (if any)
- Calls the user program
- Constructs the response:
  - HTTP status code
  - HTTP headers
  - Set-Cookies requests (if any)
  - Response data (if any)
- Sends the response back to the client
z/VSE 6.2: z/VSE as a REST Client

Description of the REST Parameter Block:
https://www.ibm.com/support/knowledgecenter/SSB27H_6.2.0/fa2ws_how_rest_control_blocks_are_used.html
z/VSE 6.2: z/VSE as a REST Client

— The z/VSE REST-Engine...
  • Gets called from the user program
  • Constructs the request
    o Splits the URL into host and port and resource-uri
      \texttt{http://host:port/resource-uri[?query-string]}
    o Adds URL parameters to the query string (if any)
    o HTTP headers
    o Cookies (if any)
    o Request data (if any)
  • Sends the request to the server
  • Receives the response from the server
  • Extracts information from the response:
    o HTTP status code
    o HTTP headers
    o Set-Cookies requests (if any)
    o Response data (if any)
  • Returns back to the user program
The REST parameter block

— Contains information about the request
  • Request type (GET, PUT, POST, …)
  • URL
  • Content-Type
  • Data-type (XML, JSON, plain text, binary)
  • URL parameters from query string (http://.....?a=b&c=d)
  • Form fields
  • HTTP headers
  • Cookies
  • Authentication information
  • Response status code

— Copybooks in PRD1.BASE:
  • IESRESTH.H LE/C
  • IESRESTL.C COBOL
  • IESJSONP.P PL/1
  • IESRESTA.A HLASM

* COMMAREA layout used by the REST Engine to call the user program
* (VSE as REST server) or to get called by the user program
* (VSE as REST client):

```asm
01 REST-COMMAREA.
  02 REST-VERSION          PIC 9(9) BINARY.
  02 REST-EBCDIC-CODEPAGE  PIC X(16).
  02 REST-FLAGS            PIC 9(9) BINARY.
  02 REST-RETCODE          PIC 9(9) BINARY.
  02 REST-PRIVATE          USAGE IS POINTER.
  * Request specific fields:
    02 REST-REQ-ACTION       PIC 9(9) BINARY.
    02 REST-REQ-URL          PIC X(2048).
    02 REST-REQ-COMMANDTYPE  PIC X(128).
    02 REST-REQ-DATA-TYPE    PIC 9(9) BINARY.
    02 REST-REQ-DATA-PRTR    USAGE IS POINTER.
    02 REST-REQ-DATA-LENGTH  PIC 9(9) BINARY.
    02 REST-REQ-URL-PARAMS-TSQ PIC X(8).
    02 REST-REQ-FIELD-TSQ    PIC X(8).
    02 REST-REQ-HTTP-HEADERS-TSQ PIC X(8).
    02 REST-REQ-COOKIE-TSQ   PIC X(8).
    02 REST-REQ-USER-TYPE    PIC 9(9) BINARY.
    02 REST-REQ-USER-PRTR    USAGE IS POINTER.
    02 REST-REQ-USER-LENGTH  PIC 9(9) BINARY.
    02 REST-REQ-USER-NAME    PIC X(64).
    02 REST-REQ-USER-PASSWORD PIC X(64).
    02 REST-REQ-ACCEPT       PIC X(128).
  * Response specific fields:
    02 REST-RESP-HTTP-STATUS-CODE PIC 9(9) BINARY.
    02 REST-RESP-HTTP-STATUS-TEXT PIC X(128).
    02 REST-RESP-COMMANDTYPE    PIC X(128).
    02 REST-RESP-DATA-TYPE      PIC 9(9) BINARY.
    02 REST-RESP-DATA-PRTR      USAGE IS POINTER.
    02 REST-RESP-DATA-LENGTH    PIC 9(9) BINARY.
    02 REST-RESP-FIELD-TSQ      PIC X(8).
    02 REST-RESP-HTTP-HEADERS-TSQ PIC X(8).
    02 REST-RESP-COOKIE-TSQ     PIC X(8).
    02 REST-RESP-LOCATION      PIC X(2048).
```

© Copyright IBM Corporation 2019
Handling XML and JSON data

The z/VSE REST Engine automatically translates request and response data

— **XML:**  
  Content-Type: `text/xml` or `application/xml`  
  - XML data is parsed by the XML parser  
  - An XML tree in memory is passed to the user program

— **JSON:**  
  Content-Type: `text/json` or `application/json`  
  - JSON data is parsed by the JSON parser  
  - A JSON tree in memory is passed to the user program

— **URL encoded:**  
  Content-Type: `application/x-www-form-urlencoded`  
  - Form field data is parsed and passed via TS queue entries

— **Plain text:**  
  Content-Type: `text/*` (other than xml or json)  
  - ASCII-EBCDIC converted

— **Binary data:**  
  anything else

— **Multipart data:**  
  - Each part is converted individually based on its content type
**JSON data**

Example:

```json
{"menu": {
  "id": "file",
  "popup": {
    "menuItem": [ {
      "value": "Open",
      "onclick": "OpenDoc()"
    }, {
      "value": "Close",
      "onclick": "CloseDoc()"
    } ]
  }
}
```

The JSON control blocks are defined in copybooks in PRD1.BASE:
- IESJSONH.H LE/C
- IESJSONC.C COBOL
- IESJSONP.P PL/1
- IESJSONA.A HLASM
XML data

Example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<VSAMmap type="temporary" status="test">
<!-- VSAM map layout -->
<mapname>mapname12</mapname>
<field>
  <name>str</name>
  <type>STRING</type>
</field>
<field>
  <name>sign</name>
  <type>SIGNED</type>
</field>
</VSAMmap>
```

The XML control blocks are defined in copybooks in PRD1.BASE:
- IESXMLAH.H LE/C
- IESXMLCB.C COBOL
- IESXMLPL.P PL/1
- IESXMLAS.A HLASM
LDAP Sign-on Support

Scenario 3: Modern security solutions

Identity Management
Manages Identities
- User-IDs
- Passwords
- Groups
- Access rights
- Policies

z/VSE
CICS Applications and Batch Jobs
LDAP Sign-on
User mapping

User
Sign-on

Administrator
Add, Change, Delete, ...

User
Authenticate

User
Authenticate

User
Authenticate

User
Authenticate

VM Workshop 2019
32 © Copyright IBM Corporation 2019
z/VSE SNMP Monitoring Agent support

- z/VSE Monitoring Agent enables customers to monitor z/VSE systems using standard monitoring interfaces (SNMP V1)
  - Available since z/VSE V4.3
  - It also includes an open interface, which enables customers or vendors to use own programs (plugins) to collect additional data

- Data collected by the IBM provided plugins contains
  - Information about the environment (e.g. Processor, LPAR and z/VM information)
  - Number of partitions (static, dynamic, total, maximum)
  - Partition priorities
  - Number of CPUs (active, stopped, quiced)
  - Paging (page ins, page outs)
  - Performance counters overall and per CPU
  - CPU address and status
  - CPU time, NP time, spin time, allbound time
  - Number of SVCs and dispatcher cycles

- Plugin interface allows Vendors to provide additional monitoring data
Rational Developer for IBM Z (RDz)

with z/VSE Development & Debug Plug-in for RDz – from QGroup

Scenario 3: Modern application development
Access to VSE’s resources

Editor with Syntax Highlighting

Outline View shows structure of program

Project View allows grouping of program files
Remote Compile shows error in source editor

Compiler Messages after Remote Compile

CHAR-COUNT3 WAS NOT DEFINED AS A DATA-NAME. THE STATEMENT WAS DISCARDED.
Access to VSE console
3270 Terminal Emulator included
Remote Debugger allows debugging of COBOL programs.
Summary

The z/VSE Connectors provide lots of possibilities to connect z/VSE with the rest of the world.
Questions?

?  

THANK YOU
Notices and disclaimers

— © 2019 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

— U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

— Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

— IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”

— Any statements regarding IBM’s future direction, intent or product plans are subject to change or withdrawal without notice.

— Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

— References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

— Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

— It is the customer’s responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.
Notices and disclaimers continued

— Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM’s products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

— The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

— IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)