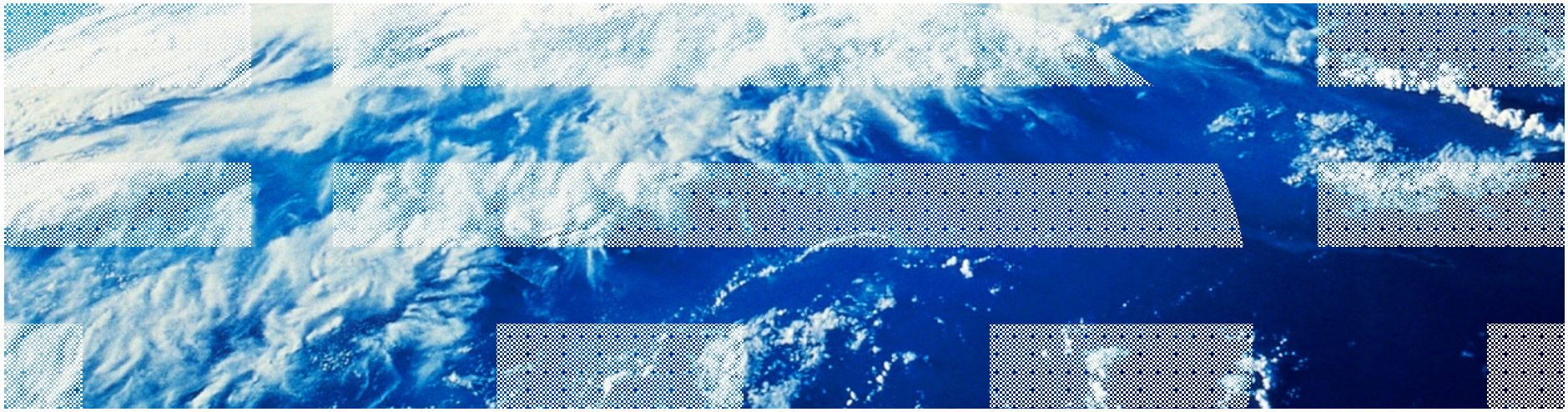
z/VSE Connectors Update

Ingo Franzki, IBM



Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

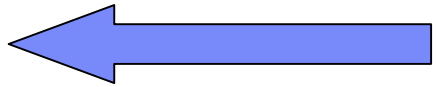
Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Agenda

§ z/VSE V5.1 Connector Enhancements



- VSE Script Connector: SYSIPT Variables Support
- VSE Script Connector: New functions
- VSE Script Connector: Logging of script input and output
- VSAM Redirector: MapperConfigGUI Enhancements
- VSE Connector Client & Server: LDAP signon support
- VSE Connector Client & Server: LIBR DATA=YES

§ z/VSE V4.3 Connector Enhancements

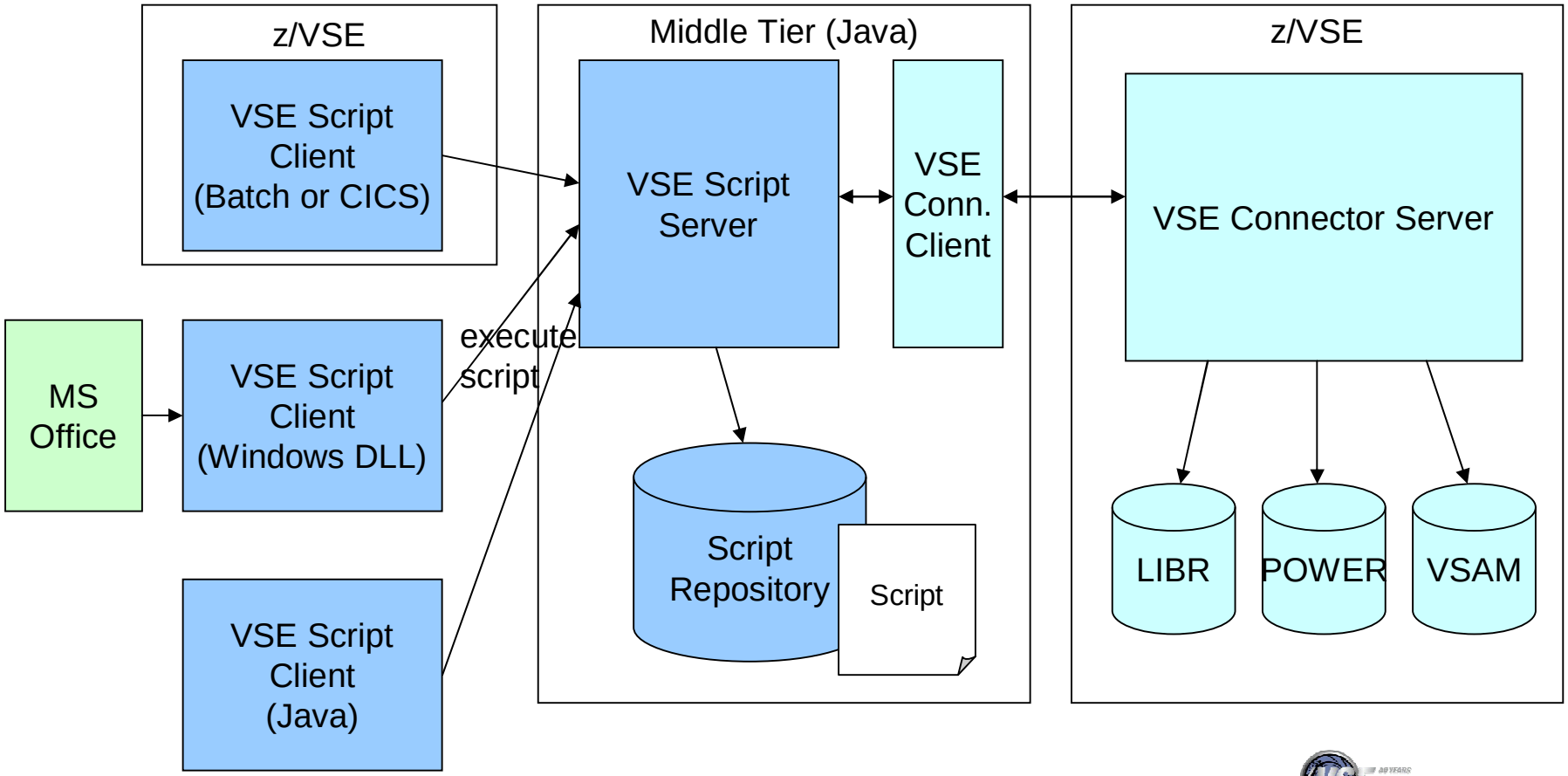
- POWER Output Generation Messages and exploitation in Java-based Connector
- Decimal Position support for Java-based Connector
- EXCPAD for VSAM Redirector
- Redirector Trace activation via VSAM SNAP
- New Tool: Virtual z/VSE FTP Daemon

§ z/VSE V4.2 Connector Enhancements

- Web Service (SOAP) Security
- Web Service support for long parameter names

z/VSE V5.1: VSE Script Connector Overview

- **Part of the z/VSE Connectors since z/VSE V3.1**
- **Allows remote access to z/VSE resources and data from non-Java platforms**



z/VSE V5.1: VSE Script Connector: SYSIPT Variables Support

- The SYSIPT variables support extends the VSE Script BATCH client programs by adding support for symbolic variables
- Customers can assign the variables dynamically in JCL before they invoke the VSE Script batch client
- Usage examples:
  - Feed in data from previous job steps
  - Centralize often used settings, such as IP address
- Example: sets the target host and the script to execute using variables:

```
* $$ JOB JNM=START,DISP=L,CLASS=A
// JOB START
// LIBDEF *,SEARCH=(PRD1.BASE,PRD2.SCEEBASE,PRD2.DBASE)
// SETPARM DESTIP='10.31.0.1'
// SETPARM SCRIPT='testscript.src'
// SETPARM HELLO='HELLO '
// SETPARM WORLD='WORLD'
// EXEC IESSCBAT,PARM='CODEPAGE=CP1047 SHOWERROR=YES SYMBOLS=YES'
&DESTIP:4711
&SCRIPT
Script input …
&HELLO&WORLD.!
/*
/&
* $$ EOJ
```

z/VSE V5.1: VSE Script Connector: SYSIPT Variables Support

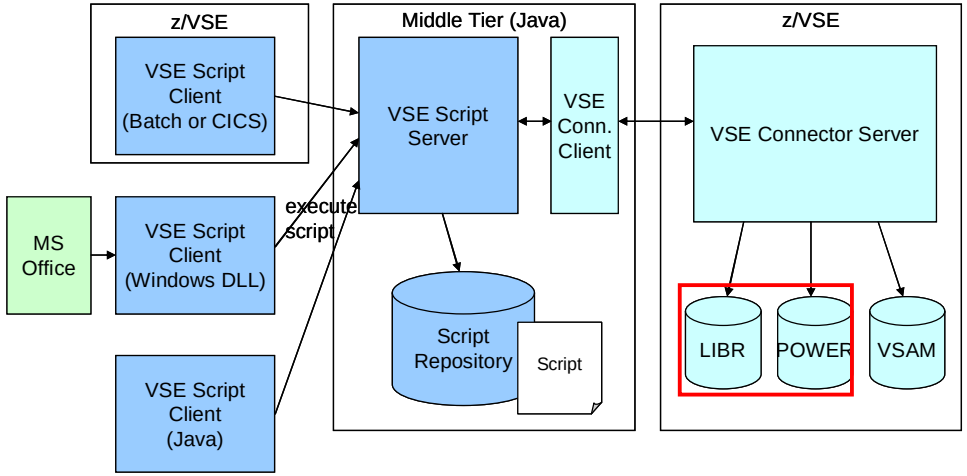
- **The support must be enabled by setting the new PARMS parameter SYMBOLS=YES**
  - The default for this new parameter is SYMBOLS=NO to ensure backward compatibility.
- **The defined format of the variables specified in SYSIPT will be the same format that is described in "System Control Statements" manual, Job Controls 'Symbolic Parameters' chapter, available here:**
  http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IESSOE51/3.7?SHELF=IESVSE71&DT=20090403085040
  - A symbolic variable starts with '&'
  - When a '&' is needed in the input, write it as '&&'
  - Symbolic variable name contains of characters [0-9][A-Z] (yes, uppercase!) (this is not checked by the library, but the symbol would be not found)
  - Any other character beside [0-9][A-Z] marks the end of the current symbol
  - A '.' after the symbol name marks the end of the symbol without printing a character, for example '&SYMBOL.ALL' where SYMBOL='HELLO' will result in 'HELLOALL' without a '.'
  - The maximum final line length is not limited
- **A symbolic variable can be defined in JCL using**

```
// SETPARM [SYSTEM] VARIABLE='VALUE'
```

z/VSE V5.1: VSE Script Connector: New functions

§ New LIBR functions

- List libraries, sub libraries, members
- Create/delete sub library
- Copy/move member
- Delete/rename member
- Download member (binary and text)
- Upload member (binary and text)
- Put member on POWER queue
- Get member from POWER queue



§ New POWER functions

- Get entry in binary
- Put entry in binary

§ Codepage related functions

- Convert a string to binary and vice versa, using a specific codepage
- Write/read a local file in binary

à Support for Binary data and Codepage tools allow to use VSE Script Connector with Double Byte Characters Set (DBCS) and Unicode data

z/VSE V5.1: VSE Script Connector: Logging of script input/output

- The VSE Script Server now optionally prints all input and output data into the server log
- This new feature can be used for audit purposes
- The logging can be enabled using the new optional configuration parameters
  - logscriptinputparams
  - logscriptoutput
- Additionally a script function was added to print directly into the server log:
  - PRINTLOG()
- This function can be exploited by user scripts to print audit-relevant messages to the server log

```
04.11.2010 08:56:30 (8) - Client connection request from 127.0.0.1
04.11.2010 08:56:30 (11) - Client has been accepted.
04.11.2010 08:56:30 (11) - Connection has been accepted from 127.0.0.1
04.11.2010 08:56:30 (11) - Using default system codepage.
04.11.2010 08:56:30 (11) - Executing script 'samples/gosub.src'
04.11.2010 08:56:30 (11) - Script receives 3 input parameter(s):
04.11.2010 08:56:30 (11) -   argv[1]='2'
04.11.2010 08:56:30 (11) -   argv[2]='test'
04.11.2010 08:56:30 (11) -   argv[3]='parameters'
04.11.2010 08:56:30 (11) - Script output follows:
04.11.2010 08:56:30 (11) -   'start'
04.11.2010 08:56:30 (11) -   'sub'
04.11.2010 08:56:30 (11) -   'end'
04.11.2010 08:56:30 (11) - PRINTLOG: 'New log output'
04.11.2010 08:56:30 (11) - Connection has been terminated from 127.0.0.1
04.11.2010 08:56:30 (11) - Client has been disconnected.
```

z/VSE V5.1: VSAM Redirector: MapperConfigGUI Enhancements

- MapperConfigGui is part of VSE VSAM Redirectors DBHandler
- The MapperConfigGui now allows to save profiles which contain the information needed to access a database target
- The user don't have to enter them again and again when he switches between different JDBC targets, e.g. a test database system and the production database system
- For security reasons the password is never saved in the profile

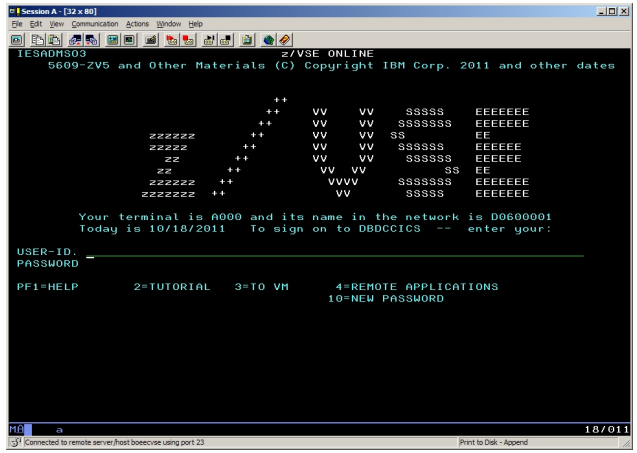
z/VSE V5.1: VSE Connector Client & Server: LDAP signon support

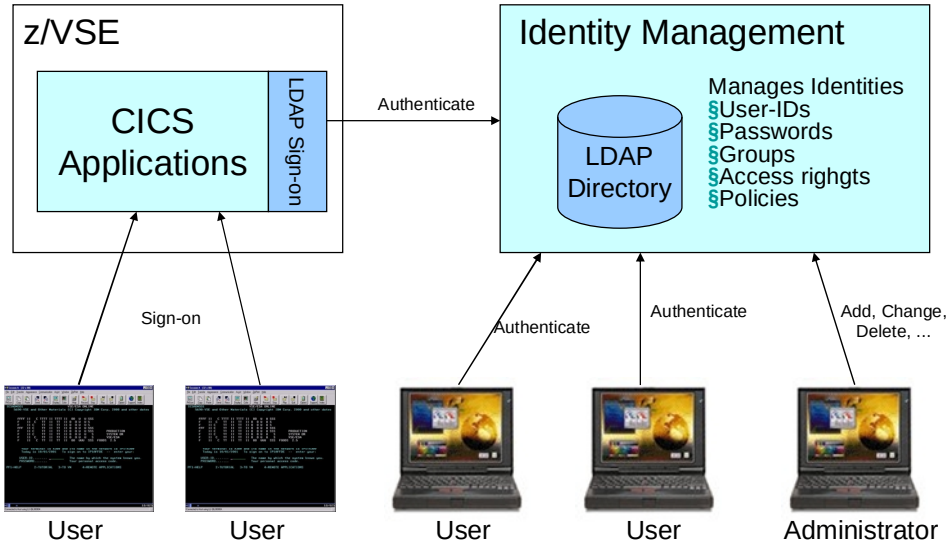
§ z/VSE V4.2 added support for LDAP Signon

- Authenticate against a corporate wide Identity Management System (using LDAP)
- Single Signon/Simplified Signon by using the same user-ID and password
- User-ID & passwords up to 64 characters

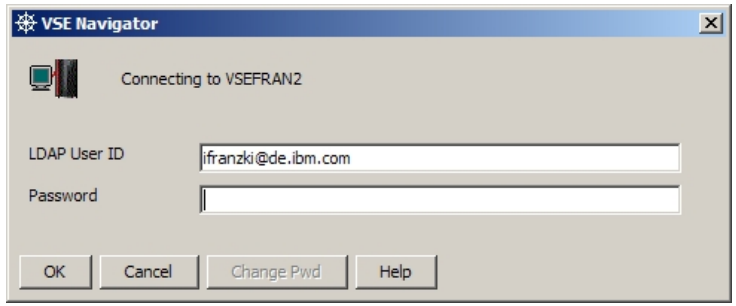
§ z/VSE V5.1 adds LDAP signon support for the VSE Connector Client & Server

- A Java based application can now use the same corporate user-ID and password as for IUI signon





Example: VSE Navigator



z/VSE V5.1: VSE Connector Client & Server: LIBR DATA=YES

- **VSE Connector Client & Server supports access to LIBR members since VSE/ESA 2.5**
  - Download LIBR members
  - Upload LIBR members
- **Also access of .PROC members (procedures) is possible**
- **Procedures may be cataloged with the DATA=YES attribute, if they contain SYSIPT data**

```
// EXEC LIBR
ACCESS S=lib.sublib
CATALOG member.type DATA=YES
....
/*
```

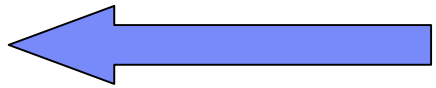
- **Prior to z/VSE V5.1, any members created by the VSE Connector Server used DATA=NO**
  - You could damage an procedure that was previously cataloged with DATA=YES
- **Since z/VSE V5.1, the VSE Connector Client & Server support the DATA=YES attribute**
  - You can store a member with DATA=YES
  - Use method `VSELibraryMember.setSYSIPTDataInProcedure(boolean sysiptdata)`
- **Example: VSE Navigator**
  - Double-click on a member to edit it
  - Member automatically retains its DATA=YES attribute

Agenda

§ z/VSE V5.1 Connector Enhancements

- VSE Script Connector: SYSIPT Variables Support
- VSE Script Connector: New functions
- VSE Script Connector: Logging of script input and output
- VSAM Redirector: MapperConfigGUI Enhancements
- VSE Connector Client & Server: LDAP signon support
- VSE Connector Client & Server: LIBR DATA=YES

§ z/VSE V4.3 Connector Enhancements



- POWER Output Generation Messages and exploitation in Java-based Connector
- Decimal Position support for Java-based Connector
- EXCPAD for VSAM Redirector
- Redirector Trace activation via VSAM SNAP
- New Tool: Virtual z/VSE FTP Daemon

§ z/VSE V4.2 Connector Enhancements

- Web Service (SOAP) Security
- Web Service support for long parameter names

z/VSE V4.3: POWER Output Generation Messages Support

§ As of z/VSE 4.2, VSE/POWER can generate the following notification messages for a SAS (Spool Access Support) application

- **Job Generation message 1Q5HI (JGM):**
  Informs that the job, submitted via SAS interface, has generated another job as punch output with DISP=I
- **Job Completion message 1Q5DI (JCM):**
  Informs that the job, submitted via SAS interface, has completed

§ With z/VSE 4.3, a new notification message has been added:

- **Output Generation message 1Q5RI (OGM)**:
  Is generated each time when the job, submitted via SAS interface, has created LST or PUN entry, and this entry became ready for processing

§ For details about how to use the VSE/POWER Spool Access Support programming interface, please see Manual "VSE/POWER Application Programming"

z/VSE V4.3: POWER Output Generation Messages

§ With the new OGM support, a Job Scheduler application can now control the whole lifetime of a job:

- Job Submission
- Job Generation (DISP=I)
- Job Completion
- Output Generation

§ Without OGMs, its hard to find all outputs generated by a job

- A job may produce various outputs
  - Multiple LST/PUN cards in the job
  - Output segmentation
- Outputs may have different names than the generating job (JNM=nnn in LST/PUN card)
- Outputs may have different numbers than the generating job
  - Segmentation overflow (more than 127 segments)
  - Multiple LST/PUN cards in the job

§ OGMs now provide a save way to retrieve all outputs generated by a Job

z/VSE V4.3: POWER Output Generation Messages

§ The VSE Connector Client & Server now support OGMs

– When submitting a Job via VSE Connector Client, an application can request to queue OGMs for the job:

```
VSEPowerEntry entry = new VSEPowerEntry(system,QUEUE_RDR,”MYJOB”);
entry.setQueueComplMsgs(true);   // request job completion messages
entry.setQueueOutputMsgs(true);  // request output generation messages
entry.put(jobfile);              // submit the job
```

– The application can check if a job has completed:

```
if(entry.isCompleted())          // check if the job execution is complete
```

– When the job has completed, the application can retrieve a list of outputs generated by the job:

```
entry.addVSEResourceListener(this);    // register as resource listener
entry.getOutputList();                 // retrieve the list of output entries
entry.removeVSEResourceListener(this); // de-register resource listener
```

– The application can then process the list of received VSEPowerEntry objects

– Example: com/ibm/vse/samples/SubmitJob.java in the samples directory

z/VSE V4.3: Decimal Positions

- **The VSE Connector supports decimal data types like PACKED or ZONED**
  - **in both signed and unsigned variants**
- **Those data types are often used by customer applications to store monetary type of information**
  - Monetary information usually has at least 2 decimal places, e.g. $123.45
  - COBOL or PL/1 applications usually store such decimal numbers as packed decimal (COMP-3) or zoned decimal data types, with implied decimal position
  - The implied decimal position (as the name implies) is not really stored as part of the decimal number, but it is implied when reading or updating the number
- **Example:**
  - The decimal value of 123.45 is stored as packed decimal: `X'12345C'`
  - The implied decimal position is 2 in this case (2 digits from the right).
- **Since the decimal position is not stored as part of the numerical data, that information needs to be stored as part of the mapping information together with the field name, offset, length and type**
- **With z/VSE 4.3, the VSE Connectors has been enhanced to support (implied) decimal positions**

z/VSE V4.3: Decimal Positions

- **Decimal positions apply to the following data types:**
  - PACKED — Packed Decimal (COBOL COMP-3)
  - UPACKED — Unsigned Packed Decimal
  - ZONED — Zoned Decimal (COBOL PIC 9(n))
  - UZONED — Unsigned Zoned Decimal
- **The decimal position can be:**
  - Zero — No decimal position (e.g. 12345)
  - Positive — Specifies the number of decimal digits from the right (e.g. 123.45 has a decimal position of 2)
  - Negative — Specifies the number of implied zero digits right to the number (e.g. 1234500 has decimal position of -2 if stored as 12345 as un-scaled value)
- **The decimal position is interpreted by the VSE Connector Client when passing such numerical data to the calling application**
  - The implied decimal position as stored in the mapping is applied to the (un-scaled) number, before passing it to the user application
  - Any number passed from user application to the VSE Connector Client is converted to its un-scaled value based on the implied decimal position
- **Decimal numbers with a non-zero decimal position are represented as Java java.math.BigDecimal object by the VSE Connector Client**

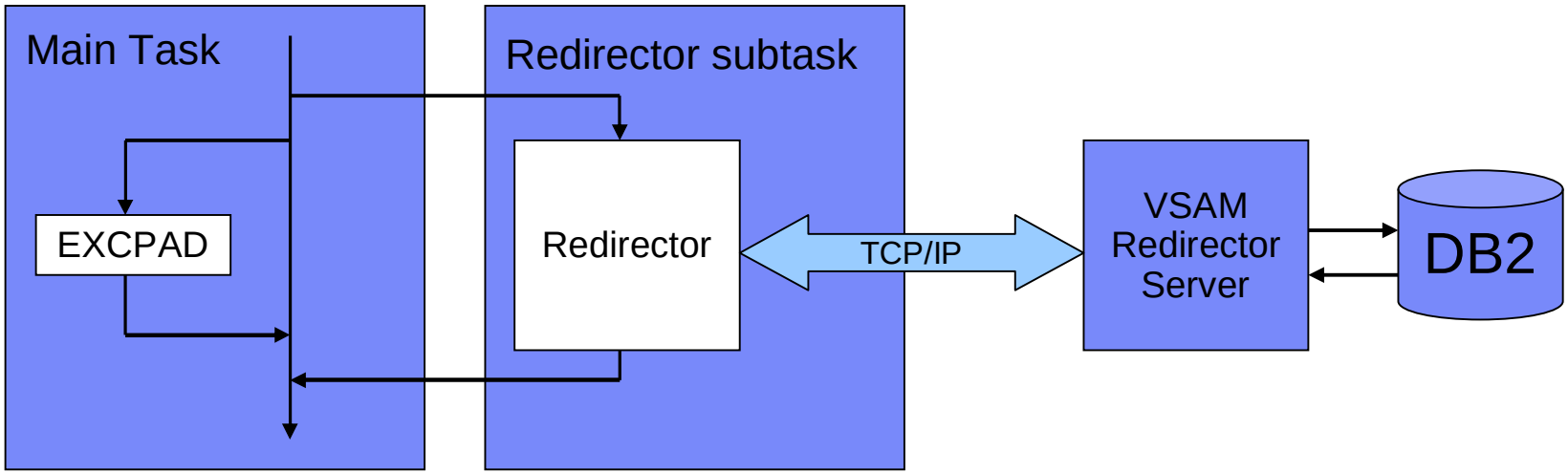
z/VSE V4.3: Decimal Positions

- **The following components have been updated to support decimal positions**
  - The mapping file (IESMAPD) to store the decimal position
  - VSE Connector Client to handle decimal positions and java.math.BigDecimals
  - VSAM JDBC Driver to support Decimal Positions
  - VSE Script Server to support Decimal Positions
  - IDCAMS RECMAP command to support Decimal Positions
  - VSAM Maptool to support Decimal Positions
  - VSE Navigator to support Decimal Positions

- **Any existing mapping stored in the mapping file IESMAPD can be used unchanged with z/VSE 4.3 or later**
  - Any migrated decimal field will have a zero decimal position, which is what they implicitly had when no decimal position support was existing.
- **Any existing application that did work with an older version of the VSE Connector Client will work unchanged with the z/VSE 4.3 version of VSE Connector Client**
  - As long as the mapping is not changed to use decimal positions other than zero
  - Mappings migrated or copied over from previous versions will automatically have a zero decimal position, as stated above
  - User applications may have to be adapted if non-zero decimal positions are used

z/VSE V4.3: Redirector EXCPAD

- **Prior to z/VSE 4.3 the VSAM Redirector was executed in the same subtask as VSAM and the application (caller)**
  - Redirector activities may be time consuming (network transfers, database operations, ...)
    - During this time, no other activities are possible for this subtask
  - Under CICS, VSAM normally returns back via EXCPAD exit when waiting for an I/O
    - Allows CICS to perform other activities concurrently
- **Since z/VSE 4.3 VSAM executes the Redirector under a separate subtask**
  - VSAM now also returns back to CICS via EXCPAD when waiting for Redirector
    - Allows CICS to perform other activities concurrently
  - This capability is primarily implemented for CICS TS transactions.
    - The Redirector EXCPAD is not used for VSAM files opened by CICS/VSE.



z/VSE V4.3: Redirector EXCPAD

- **Prior to z/VSE 4.3 heavy use of VSAM Redirector could slow down transaction processing in CICS**
  - Due to VSAM requests block the CICS I/O task when Redirector is active
- **With the new subtask the VSAM Redirector handling no longer blocks the CICS I/O task**
  - Allowing other transactions to do its work
  - Multiple redirected requests will be queued up for processing in the new subtask
- **The EXCPAD user exit is enabled automatically under the following conditions:**
  - a VSE/VSAM cluster is enabled for the Redirector
  - the EXCPAD exit is defined during the OPEN request
- **VSAM will attach only one Redirector subtask per partition even if multiple redirected files are opened in the partition with an active EXCPAD**
- **Support is transparent**
  - No need to configure or setup anything
  - All types of Redirector activities are processed in subtask (except OPEN/CLOSE)
    - VSAM Redirector OWNER=VSAM or REDIRECTOR
    - VSAM Capture Exit
    - Customer/Vendor implemented Redirector Exit

z/VSE V4.3: Redirector Trace activation via SNAP

§ The VSAM Redirector host parts consist of

- IESVEX01 (will be renamed to IKQVEX01 when activating redirection)
- IESREDIR – VSAM Redirector Client
- IESVSCAP – VSAM Capture Exit

§ All 3 parts have an internal trace facility

- Prior to z/VSE 4.3, the trace could only be activated through a MSHP PATCH
  - Trace was written to SYSLOG (console) only
- Since z/VSE 4.3, the trace can now be dynamically enabled (and disabled) via the VSAM SNAP trace
  - Trace is now written to SYSLST (listing) of job

§ Trace activation is done via IKQVEDA:

```
// EXEC IKQVEDA,PARM='SYSIPT'
ENABLE SNAP=0010,PART=F2
END
/*
```

z/VSE V4.3: VSAM SNAP Trace assignments

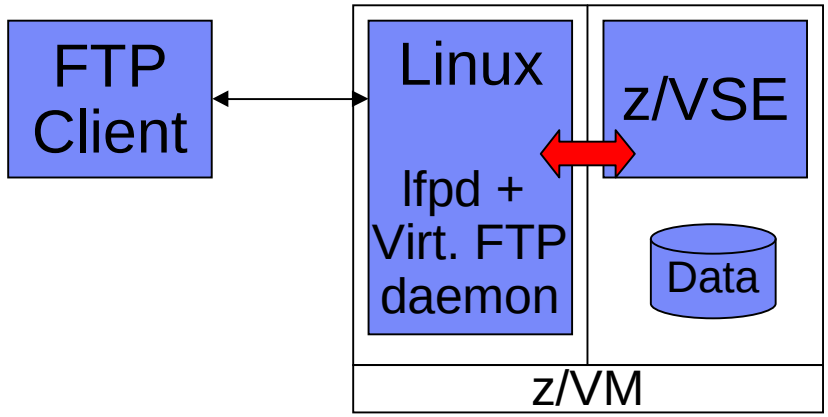
Type:	Enables:
0001	Catalog management error code trace
0002	Buffer manager trace
0003	OPEN control block dump (when OPEN processing is complete) OPEN error trace (prints control blocks if an error occurs during OPEN processing) CLOSE control block dump (at the beginning of CLOSE processing)
0004	VSE/VSAM I/O trace
0005	I/O error trace
0008	Catalog management I/O trace (prints all I/O operations done by VSE/VSAM catalog management)
0009	Record management error trace (prints control blocks for any error detected by VSE/VSAM record management)
0010	**Redirector Trace**
0013	In-core wrap trace for trace points within VSE/VSAM Record Management
0014	Level2 SNAP013 Trace (I/O, EXCPAD and z/VSE Lock Activity)
0015	Level3 SNAP013 Trace (Buffer Management)
0016	Produce a printout (PDUMP) each time the SNAP013 Trace Table wraps.

New Tool: Virtual z/VSE FTP Daemon

§ **The Virtual z/VSE FTP Daemon can be installed on any Java-enabled platform and emulates an FTP server**
  – The actual access to z/VSE resources is done using the VSE Connector Server.

§ **Download:** http://ibm.com/zvse/download

à Fits perfectly to Linux Fast Path



§ **The Virtual z/VSE FTP Daemon:**
  – Handles all incoming FTP clients.
  – Connects to one or multiple VSE Connector Servers.
  – Is responsible for connection-handling.
  – Is responsible for data translation (ASCII-EBCDIC).
  – Is IPv6 ready
    • You can connect FTP clients using IPv6, the Virtual z/VSE FTP Daemon connects to the VSE Connector Server using IPv4.
  – Supports SSL
    • both for the FTP connection (between FTP client and Virtual z/VSE FTP Daemon, using implicit SSL (FTPS)),
    • and for the connection to the VSE Connector Server (between Virtual z/VSE FTP Daemon and z/VSE host).

Agenda

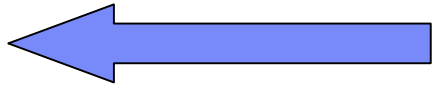
§ z/VSE V5.1 Connector Enhancements

- VSE Script Connector: SYSIPT Variables Support
- VSE Script Connector: New functions
- VSE Script Connector: Logging of script input and output
- VSAM Redirector: MapperConfigGUI Enhancements
- VSE Connector Client & Server: LDAP signon support
- VSE Connector Client & Server: LIBR DATA=YES

§ z/VSE V4.3 Connector Enhancements

- POWER Output Generation Messages and exploitation in Java-based Connector
- Decimal Position support for Java-based Connector
- EXCPAD for VSAM Redirector
- Redirector Trace activation via VSAM SNAP
- New Tool: Virtual z/VSE FTP Daemon

§ z/VSE V4.2 Connector Enhancements



- Web Service (SOAP) Security
- Web Service support for long parameter names

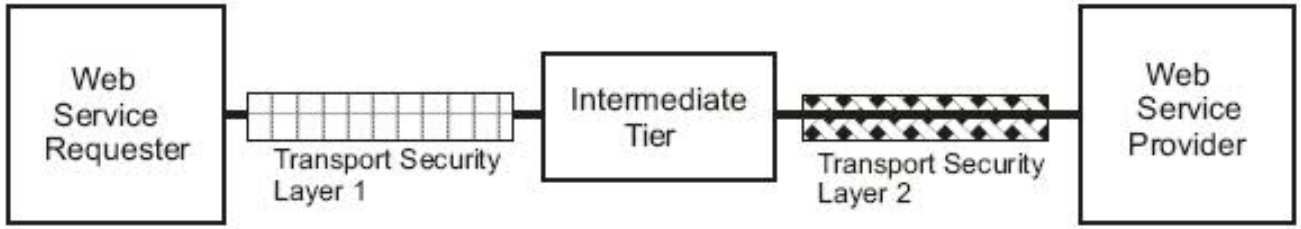
z/VSE V4.2: Web Service Security - What is SOAP

Web Servcies uses the Simple Object Access Protocol (SOAP) to transport requests and responses

```
<soap:Envelope>
  <soap:Header>
    ...
  </soap:Header>
  <soap:Body>
    <GetStock>
      <Company>IBM</Company>
    </GetStock>
  </soap:Body>
</soap:Envelope>
```

z/VSE V4.2: Web Service Security - Overview

- **Web Service security can be divided into:**
  - Transport-layer security
    - E.g. Secure Socket Layer, HTTPS, IPSec, VPN
  - Message-layer security
    - Security related elements within the SOAP message



- **Both transport-layer and message-layer security provide security features for:**
  - Authentication/authorization
  - Data encryption and signatures

z/VSE V4.2: Web Service Security - Authentication

§ Using authentication allows a service provider to check who is using the requested service

- The service provider may use this information to execute the service under a specific user-ID, with its associated access rights (authorization)

§ Terms:

- **Authentication:**
  - The process of identifying an individual using the credentials of that individual.
- **Authorization:**
  - The process of determining whether an authenticated client is allowed to access a resource or perform a task within a security domain. Authorization uses information about a client's identity and/or roles to determine the resources or tasks that a client can perform.
- **Credentials:**
  - A set of claims used to prove the identity of a client. They contain an identifier for the client and a proof of the client's identity such as a password. They may also include information, such as a signature, to indicate that the issuer certifies the claims in the credential.
- **Identification:**
  - The use of an identifier that allows a system to recognize a particular subject and distinguish it from other users of the system

z/VSE V4.2: Web Service Security - Authentication

§ Transport Layer Authentication

- The transport layer carries information about who is requesting the service
  - HTTP Authentication (Basic and Digest Access Authorization, see RFC 2617)
  - SSL Client Authentication with SSL/HTTPS

§ Message Layer Authentication

- The SOAP message itself carries information about who is requesting the service
  - Direct authentication, using plain text passwords or a password digest
  - Brokered Authentication, using a X.509 Certificate

    Carries the X.509 Certificate as part of the SOAP header

z/VSE V4.2: Web Service Security – Direct Authentication

§ **Direct authentication defines two ways of transporting the password:**

– **Plain text password**

- UsernameToken is used to transport the actual password.
- If you use plain-text password configuration, you must use a secure transport method (such as HTTPS)

```
<soap:Header>
  <Security xmlns=”...secext-1.0.xsd>
    <UsernameToken>
      <Username>John Smith</Username>
      <Password>Pass12wd</Password>
    </UsernameToken>
  </Security>
  ...
```

– **Password digest**

- See next foil

z/VSE V4.2: Web Service Security – Direct Authentication

- **Direct authentication with Password digest**
  - **The communicating parties (the requester and the service) use an insecure transport channel**
    - Steps must be taken to protect the passwords from being exposed to others
    - The requester creates a digest of the actual password that is concatenated with a set of random bytes (field nonce) and another value that is dependent on the creation-time (field created).

```
digest = Base64_encode(SHA-1(nonce+created+password))
```

  - **To authenticate the request, the service provider computes the digest value using the password bound to the received username.**
    - It compares the received digest value with the computed digest value.

```
<soap:Header>
  <Security xmlns="...secext-1.0.xsd>
    <UsernameToken>
      <Username>John Smith</Username>
      <Password Type="...#PasswordDigest">AFHHF23wger=</Password>
      <Nonce>ksSDGFljdfD=</Nonce>
      <Created>2010-07-15T07:12:19.573Z</Created>
    </UsernameToken>
  </Security>
  ...
```

z/VSE V4.2: Web Service Security – Brokered Authentication

§ **Brokered authentication using a X.509 Certificate**

– Carries the X.509 Certificate as part of the SOAP header

```
<soap:Header>
  <Security xmlns="...secext-1.0.xsd>
    <BinarySecurityToken EncodigType= "wsse:Base64Binary"
                         ValueType= "wsse:X509v3">
      MIICuzCCAiQCBF...
      ...
    </BinarySecurityToken>
  </Security>
  ...
```

– The certificate is base64 encoded

– The receiver can use the certificate to authenticate the requestor

– The use of a secure transport channel is recommended

- Secure Socket Layer, HTTPS

z/VSE V4.2: Web Service Security – VSE as Service Provider

- **Transport Layer Security**
  - **Encryption**
    - CICS Web Support already provides SSL support (HTTPS)
      - Configure TCPIPSERVICE in CICS for use with SSL
      - Create the required keys and certificates.
  - **Authentication**
    - CICS Web Support supports SSL client authentication (HTTPS), as well as HTTP Basic Authentication
      - To force a client to use HTTP basic authentication, you need to configure the TCPIPSERVICE to use the CICS provided converter program DFH$WBSB (specify URM=DFH$WBSB)
    - VSE SOAP Engine also passes userid and password to the service provider program

- **Message Layer Security**
  - **Authentication**
    - Support for extracting the authentication token from the SOAP header has been added
    - The VSE SOAP Engine passes the authentication information to the service provider program
    - The VSE SOAP Engine does not itself perform the authentication

z/VSE V4.2: Web Service Security – VSE as Service Requestor

- **Transport Layer Security**
  - **Encryption**
    - The z/VSE HTTP Client has been enhanced to support HTTP over SSL (HTTPS)
      - The URL must start with https:// …
      - You must provide a public/private key pair, together with certificates.
      - For details of how to specify the keys, refer to the skeleton SKSOAPOP in VSE/ICCF Library 59.
  - **Authentication**
    - SSL Client Authentication can be used
      - If requested, the SSL protocol can send the client's certificate to the server
    - The z/VSE HTTP client has been enhanced to support HTTP basic authentication
      - The Service requestor needs to tell the VSE SOAP Engine to use HTTP Authentication

- **Message Layer Security**
  - **Authentication**
    - The VSE SOAP Engine has been enhanced to support
      - UsernameToken with plain text password or password digest
      - BinarySecurityToken using X.509 Certificate
    - The Service requestor needs to tell the VSE SOAP Engine which kind of token to use (including the userid & password or certificate name)

z/VSE V4.2: Web Services – Further enhancements

§ **Mapping Long-Names to Short-Names**

– Due to the size restriction for TS Queue entries, SOAP parameters can only have names up to 16 characters (as shown in the SOAP_PROG_PARAM control block)

- If you wish to use SOAP parameters that are greater than 16 characters, you can supply your own mapping to map long names (greater than 16 chars) to short names (less than or equal to 16 characters)

– **The z/VSE SOAP Engine will translate:**

- Long names to their corresponding short names when it receives SOAP messages that contain parameters with long names
- Short names to their corresponding long names when sending out SOAP messages containing parameters with long names

– **Short names that belong to a long name must start with a "#" character**

- so that the z/VSE SOAP Engine can recognize this as a name that needs to be translated.

– **The mapping is specified in the SOAP option phase IESSOAPO.**

- Use Skeleton SKSOAPO in ICCF library 59 to supply the mapping

Questions ?

Mark your calendar:

WAVV 2012
Covington, KY, USA
April 13-17, 2012

IBM System z Technical Conference
Berlin, Germany
May 21-25, 2012

IBM System z Technical University
Las Vegas, NV, USA
October 1-5, 2012

European GSE/IBM
Technical University for
z/VSE, z/VM and Linux on System z
Mainz, Germany
October 22-24, 2012

http://ibm.com/vse/events/