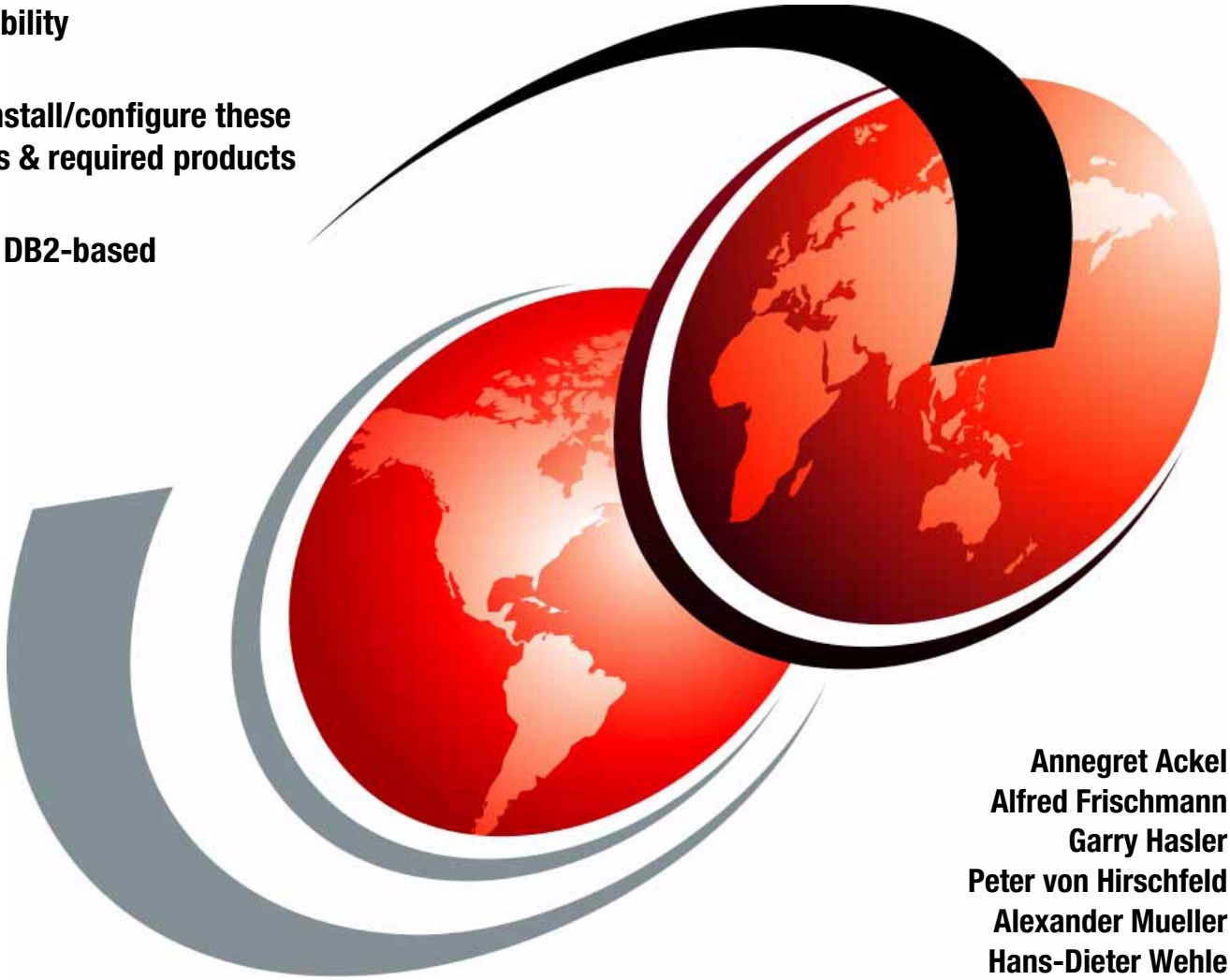
IBM

e-business Connectivity for VSE/ESA

- VSE e-business connectors enhance interoperability
- Learn to install/configure these connectors & required products
- Java- and DB2-based examples



Annegret Ackel
Alfred Frischmann
Garry Hasler
Peter von Hirschfeld
Alexander Mueller
Hans-Dieter Wehle

Redbooks

ibm.com/redbooks

International Technical Support Organization

SG24-5950-00

e-business Connectivity for VSE/ESA

November 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix C, “Special notices” on page 109.

First Edition (November 2000)

This edition applies to VSE/ESA Version 2 Release 5, Program Number 5690-VSE

This document created or updated on November 3, 2000.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Preface

The VSE customer environment is increasingly characterized by multiple servers - including different types of servers - and the need to exploit Internet technologies. Interoperability for VSE environments is now enhanced with the addition of VSE e-business connectors in VSE/ESA 2.5.

This IBM Redbook gives you a broad understanding of the new VSE e-business connectors: the Java-based connector and the DB2-based connector. Detailed usage examples will help you to install, configure, and set up these connectors and required products.

This publication complements the IBM Redbook *e-business Solutions for VSE/ESA*.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working for the International Technical Support Organization Poughkeepsie Center. The project was run in the IBM Germany Development location in Boeblingen.

Ackel Annegret, from the IBM Development Laboratory Boeblingen, was the project leader.

Alfred Frischmann, from IBM Germany

Garry Hasler, from IBM Australia

Peter von Hirschfeld, from IBM South Africa

Alexander Mueller, Marienberg, Germany

Hans-Dieter Wehle, from the IBM Development Laboratory Boeblingen, Germany

Thanks to the following people for their invaluable contributions to this project:

Erich Amrehn
International Technical Support Organization, Poughkeepsie Center

Wilhelm Mild
IBM Development Laboratory Boeblingen, Germany

Ingolf Salm
IBM Development Laboratory Boeblingen, Germany

Joerg Schmidbauer
IBM Development Laboratory Boeblingen, Germany

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Please send us your comments about this or other Redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks review” on page 121 to the fax number shown on the form.
- Use the online evaluation form found at **ibm.com**/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com

Chapter 1. VSE/ESA Version 2 Release 5 functionality

The VSE customer environment is increasingly characterized by multiple servers (including different types of servers), and by the need to exploit Internet technologies and e-business opportunities. VSE customers also want to protect and leverage their investments in S/390 applications, data, skills and hardware. In this chapter we provide a brief overview of VSE/ESA Version 2.5 functionality.

1.1 VSE/ESA Version 2 Release 5 objectives

The main objective of this release is to integrate VSE into e-business solutions on other IBM platforms in order to enhance interoperability. This is achieved through e-business connectors to VSE resources, based on a DB2 infrastructure and VSE external services, and TCP/IP enhancements.

At the same time, this release helps position VSE/ESA for S/390 growth and promote closer OS/390 affinity through enhancing CICS capabilities to strengthen the traditional workload environment, and supporting new hardware technology.

1.2 Overview

In this release of VSE/ESA, interoperability is enhanced with the addition of VSE e-business connectors. These connectors allow easy access to VSE resources from other systems. The VSE/ESA e-business connectors provide you with the resources to extend core applications to e-business applications, and to therefore protect and leverage your core application investments.

VSE/ESA 2.5 will be the vehicle for making CICS Web Support and the CICS 3270 Bridge function generally available.

VSE/ESA 2.5 also provides enhancements to VSE/ESA traditional functions such as FlashCopy and IXFP/SnapShot, as well as more dynamic classes and other improvements. These are described in 1.4, “VSE/ESA 2.5 new functions” on page 4.

1.3 A new focus for VSE/ESA 2.5.0

Most VSE users already have non-S/390 systems installed; in fact, they have different types of platforms (for example Windows, UNIX and so on). However, in many cases they are not connected. And even when they are connected, periodic file transfer is sometimes the highest level of data sharing used.

However, when you begin to exploit e-business opportunities, real-time access to current VSE production applications and data may be needed. VSE interoperability is key for integration.

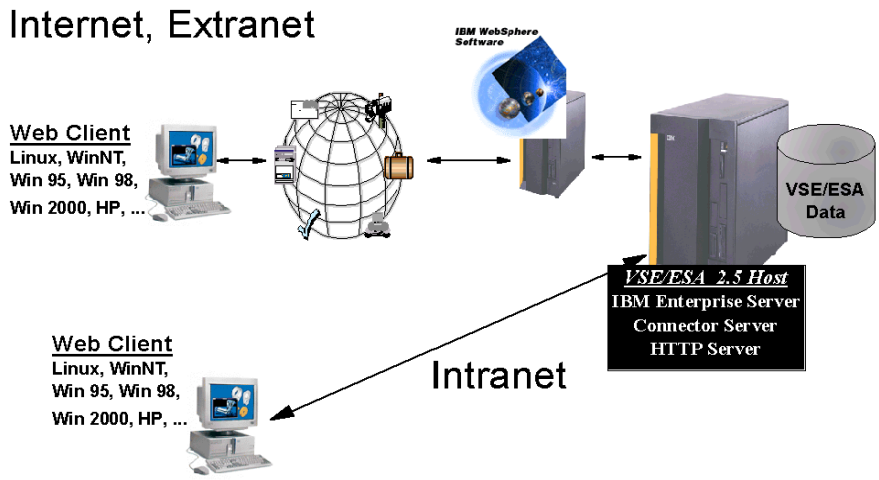


Figure 1. VSE/ESA V2.5 new focus

The new VSE/ESA focus, shown in Figure 1, is based on VSE/ESA e-business connectors, CICS Web Support, and the 3270 Bridge.

1.3.1 Interoperability through e-business connectors

VSE/ESA Version 2 Release 5 e-business connectors are designed for 3-tier and 2-tier environments. Using the 3-tier model, the user interface is normally a standard Web browser. Requests from the user are handled by a Web application server. The target Web application server is WebSphere Application Server.

The 2-tier model is generally suitable for intranet solutions only. For the intranet environment, it is not necessary to have a high level of security.

The 3-tier model allows VSE customers to build world-class e-business applications, while protecting their sizable investment in VSE skills and information assets.

The objective of all connectors is easy access to VSE resources. For example, a Java programmer writing portions of a new WebSphere-based e-business application on other platforms than VSE/ESA will be able to access the VSE/POWER queue if needed.

VSE e-business connectors will enable VSE customers to:

- Extend the reach of existing applications to new customers and end users
- More easily integrate their legacy data with new e-business applications
- More easily manage their heterogeneous environment
- Access VSE data and applications from other platforms in a direct, fast and reliable way

1.3.1.1 The current VSE/ESA e-business connectors

At present there are two types of connectors:

- The Java-based connector
- The DB2-based connector

1.3.1.2 Overview of the Java-based connector

The Java-based connector has two components. The first component is a portfolio of Java-based services that can be used in any kind of Java program, such as applets, servlets, JSPs, EJBs, or Java applications (VSE Connector Client). The second component is a set of services that run in a VSE partition (VSE Connector Server). TCP/IP is the connection between VSE and the client.

The VSE Connector Client contains the VSE JavaBeans, Java samples and online documentation. The VSE Connector Client enables you to access VSE file systems (VSE/VSAM, VSE/POWER, VSE Librarian and VSE/ICCF) and the operator console.

The VSE Connector Client consists of a single installation file which you can download from either of the following locations:

- The VSE sublibrary PRD1.BASE stored on the VSE/ESA host as IESINCON.W
- The VSE Home page with the name install.class at
  `http://www.s390.ibm.com/products/vse/support/vseconn/vsecon.htm`

The VSE Connector Server is installed on the VSE/ESA host. It is a batch application that runs in a VSE/ESA partition and provides a TCP/IP socket listener which can handle multiple clients. Java programs running on clients or a middle tier can use VSE JavaBeans to build connections to the VSE Connector Server running on the VSE/ESA host. To start this server you use the job STARTVCS, which is placed in the VSE/POWER reader queue during the installation of the VSE/ESA 2.5.0 base.

Requirements

To use the VSE/ESA e-business connectors based on VSE external services, you need the Java Development Kit (JDK) and/or the IBM WebSphere Application Server or equivalent. For the required level of JDK, check the WebSphere Application Server documentation.

1.3.1.3 Overview of the DB2-based connector

The DB2-based connector also consists of two parts. The first part is a set of VSE services based on DB2 stored procedures. The second part is a portfolio of client services based on JDBC, ODBC or Call Level Interface (CLI) standards. These connectors use DB2 Distributed Relational Database Architecture (DRDA) to access non-relational data such as VSE/VSAM and DL/I data.

Requirements

To use the DB2-based connectors, you need the DB2 Server for VSE & VM with the stored procedure capabilities. You also need DB2 Connect or equivalent.

1.3.1.4 Key prerequisites

The VSE/ESA e-business connectors, CICS Web Support (see 1.5, “New CICS functions” on page 4), and DB2 Server for VSE & VM Version 7 (DRDA support

via TCP/IP) require the TCP/IP for VSE/ESA Version 1 Release 4 Application Pak.

1.4 VSE/ESA 2.5 new functions

VSE/ESA 2.5 provides new hardware support and fulfills various traditional requirements, including VSAM performance improvements, more dynamic classes and VSE/POWER PNET over TCP/IP. For a detailed description, refer to the *VSE/ESA Release Guide.*

1.4.1 New hardware support

VSE/ESA 2.5 provides new hardware-related support as follows:

- Support for the Enterprise Storage Server (ESS) FlashCopy function:
  FlashCopy supports fast duplication of disk volumes. It improves online availability by virtually eliminating the need to stop applications for extended periods of time to create backups.
- VSAM exploitation of IXFP/SnapShot:
  A new SNAP function has been added to IDCAMS that exploits the fast duplication of selected disk volumes.

1.4.2 VSE/ESA 2.5 traditional requirements

The following traditional VSE/ESA requirements are now supported:

- More than 10 dynamic classes:
  Up to 23 dynamic classes may now be specified.
- VSAM performance improvements (buffer hashing):
  A VSAM hashing algorithm allows faster access to large VSAM Local Shared Resource (LSR) pools.
- VSE/POWER PNET exploitation of TCP/IP:
  VSE/POWER extends the existing PNET BSC/SNA/CTCA-based networking to TCP/IP.
- SDAID multiprocessor support:
  SDAID can now run in a multi-processor environment with 1 to 10 processors.
- MSHP procedure and HTML-source support:
  MSHP now supports the Librarian member type .PROC and .HTML.

1.5 New CICS functions

While many enhancements in VSE/ESA Version 2 Release 5 support the 3-tier e-business model, the 2-tier model may enable timely, cost-effective e-business solutions that extend the robust CICS environment. The additions to CICS Transaction Server for VSE/ESA (CICS TS) bring additional OS/390 functionality to VSE/ESA, expanding the affinity between VSE and OS/390.

CICS connectivity in a 3-tier configuration using CICS Transaction Gateway is described shortly in 5.3, “CICS connectivity in a 3-tier environment” on page 59. With the addition of CICS Web Support and CICS 3270 Bridge to CICS TS, CICS connectivity in a 2-tier configuration is enhanced.

This is discussed in more detail in 5.4, “CICS connectivity in a 2-tier environment” on page 60. For a complete description of this new functionality, refer to the document *CICS Transaction Server for VSE/ESA Internet Guide*, or to the IBM Redbook *CICS TS New Functions.*

1.6 New MQSeries functions

MQSeries Version 2.1.1, which is available with VSE/ESA 2.5, is enhanced with new interoperability functions. You can now connect to an MQSeries Java client installed on a Web browser. This means that you can run a Java applet on a browser that issues calls and queries to a VSE/ESA-based MQSeries queue manager, accessing applications running on your VSE/ESA host. See 5.5, “MQSeries-based connector” on page 62 for more details.

1.7 Overview of the VSE connectors

Figure 2 on page 6 provides an overview of the connectors in the VSE environment, and is based on the following available connectors:

- Java-based connectors
- DB2-based connectors
- CICS-based connectors
- MQSeries-based connectors

The chart shows the products you need to transform your business into an e-business. VSE uses the IBM Application Framework for e-business for new applications. This results in different layers, with Web server, Web application server, and so forth on the middle tier. On the S/390, you have the backend services.

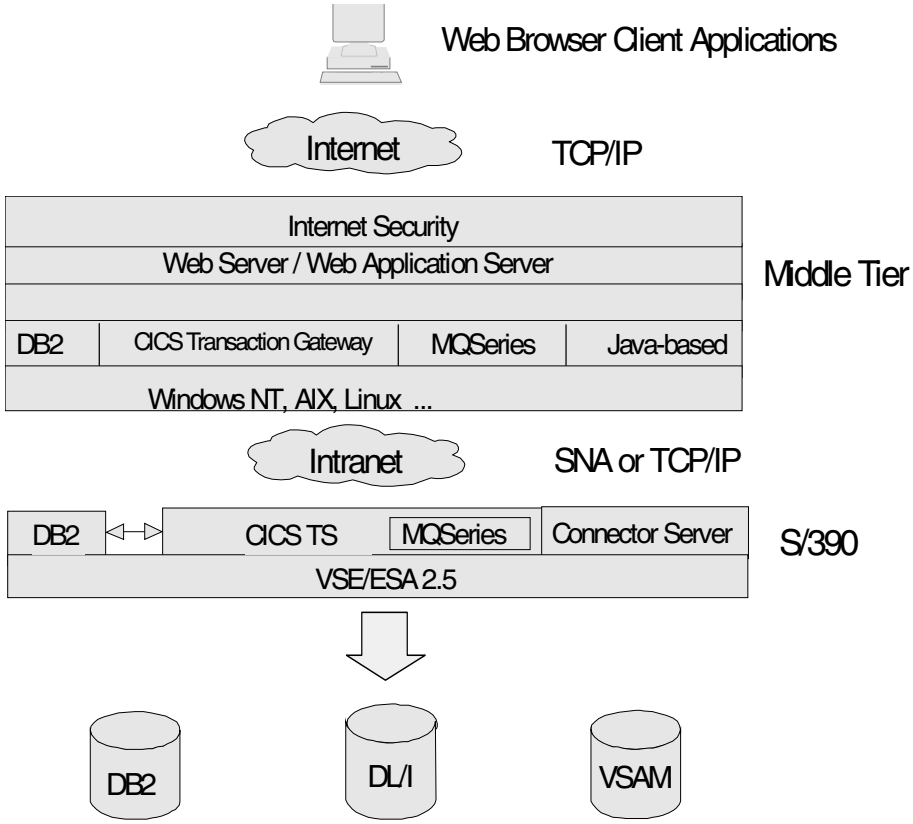


Figure 2. Overview of VSE connectors

Chapter 2. TCP/IP for VSE/ESA V1.4 functionality

TCP/IP for VSE/ESA, the native TCP/IP solution for VSE, is preinstalled in VSE/ESA 2.5. IBM has licensed this program from Connectivity Systems Incorporated. The two different function sets (Base Pak and Application Pak) are key-enabled.

The Base Pak includes the protocol stack and the socket API. The Application Pak includes, in addition, the FTP client and server, Telnet/TN3270 client and server, LPD server, LPR client and HTTP Server.

The Network File System (NFS) and General Print Server (GPS) are not included in the Application Pak; rather, they are priced features of the TCP/IP for VSE/ESA product.

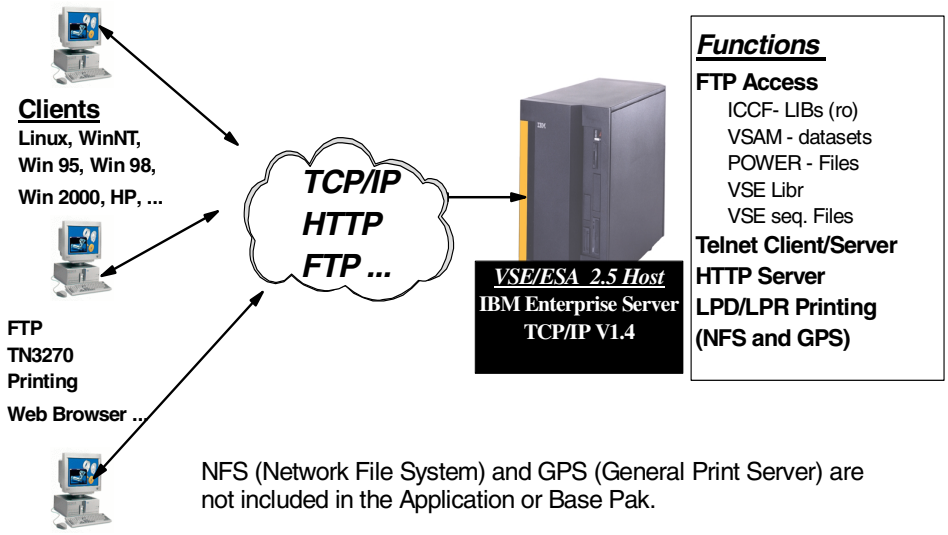


Figure 3. TCP/IP functions

Today, TCP/IP provides corporations with the ability to merge differing physical networks, while giving users a common suite of functions. It allows interoperability between equipment supplied by multiple vendors on multiple platforms, and it provides access to the Internet. In fact, the Internet, which has become the largest computer network in the world, is based on the TCP/IP protocol suite.

2.1 Hardware attachments

TCP/IP requires a network processor and associated components for attachment to the teleprocessing network. These are some of the possible network attachments you can establish:

- IBM 9221 Integrated Communication Processor
- IBM 2216 Nways Multiaccess Connector
- IBM Open Systems Adapter (OSA-2)
- Common Link Access to Workstation (CLAW)
- Channel-to-Channel support

2.2 TCP/IP applications

The aim of this section is to provide a brief overview of some TCP/IP applications. For a description of functions and applications that are available with TCP/IP for VSE/ESA, refer to *TCP/IP for VSE/ESA - IBM Program Setup and Supplementary Information,* and the IBM Redbook *Getting Started with TCP/IP for VSE/ESA 1.4*.

2.2.1 File Transfer Protocol: FTP client and server

File Transfer Protocol (FTP) provides the function of transferring files between two TCP/IP hosts.

File transfers using TCP/IP for VSE/ESA can be initiated either from VSE or from a client communicating with VSE. When you initiate a transfer from your client (such as from your PC), you are using both an FTP client and the TCP/IP for VSE/ESA FTP server. When you initiate a transfer from VSE using the FTP command in CICS, automated FTP, or a batch job, you are using the TCP/IP for VSE/ESA FTP client, and must have an FTP server installed on the remote host (the PC).

The VSE FTP clients support file transfer with all types of remote hosts. However, the remote host must be running an FTP daemon. The FTP server on the VSE system provides access to:

- VSE/VSAM ESDS and KSDS datasets
- VSE/POWER spool files
- VSE sequential datasets
- VSE libraries and sublibraries
- VSE/ICCF libraries (read only)

2.2.2 Telnet/TN3270 client and server

Teletypewriter network, or Telnet, is the virtual terminal protocol in TCP/IP. It allows users of one host to log in to a remote host and interact as normal terminal users of that host.

TCP/IP for VSE/ESA includes a Telnet server that supports the TN3270 protocol. This allows you to log on from any TCP/IP-connected TN3270 client to VTAM-based applications that are on the VSE/ESA host or accessible from the VSE/ESA host.

The Telnet/TN3270 client allows VSE users access to systems that are running Telnet/TN3270 servers. VSE users can, for example, log on to a Linux system and issue commands.

2.2.3 Remote printing: LPR client and LPD server

The Line Printer Requester (LPR) allows access to printers on other computers running the Line Printer Daemon (LPD) as though they were on your computer. Remote printing is an application that can help to increase productivity, because printouts can be printed where needed.

The clients provided (LPR, LPQ, and so on) allow you to send files or redirect printer output to a remote host running a remote print server (LPD). Some of these clients can also be used to query the status of a job.

TCP/IP for VSE/ESA provides support for your VSE system to participate in network printing using the TCP/IP Line Printer Daemon and Line Printer Requester applications. Your VSE system can be an LPD print server, an LPR client, or both. In the VSE/ESA environment, VSE/POWER reports may be printed on any TCP/IP-attached printers.

2.2.4 HTTP server and Web server

Hypertext Transfer Protocol (HTTP) is designed to allow the transfer of Hypertext Markup Language (HTML) documents. HTTP is based on request-response activity. A client, running an application called a *browser*, establishes a connection with a server and sends a request to the server in the form of a request method. The server responds with a status line including the message's protocol version, and a success or error code, followed by a message containing server information, entity information, and possible body content.

An HTTP transaction is divided into four steps:

1. The browser opens a connection.
2. The browser sends a request to the server.
3. The server sends a response to the browser.
4. The connection is closed.

TCP/IP and Web technology can easily extend the reach of an existing host-based application to new external or internal customers or end users.

TCP/IP for VSE/ESA includes an HTTP server to support the use of Web browsers such as Netscape Communicator or Microsoft Internet Explorer to access a variety of data residing on the VSE system. The data can consist of HTTP objects, such as Java applets, HTML documents and many others. The role of the Web server is to access this data and send it to the Web browser. The Web browser receives the data, interprets it, and formats it for presentation to the Web user.

2.2.5 Network File System (NFS)

The Network File System (NFS) enables machines to share file systems across a network. It allows authorized users to access files located on remote systems as if they were local.

The NFS feature of TCP/IP for VSE/ESA allows another system running an NFS client to access VSE files directly. TCP/IP for VSE/ESA offers NFS server support only. It does not offer an NFS client on VSE. NFS clients are widely available for a number of PC systems. For more information see the document *TCP/IP for VSE Optional Products, Release 1.4* .

The NFS client users can access:

- VSAM ESDS files
- VSE/ESA libraries
- VSE/POWER queues

2.2.6 General Print Server (GPS)

The GPS feature for TCP/IP for VSE/ESA provides TCP/IP for VSE/ESA users with the ability to transparently map VTAM-based printing to LPR/LPD-based network printers. The GPS feature now enables the customer to perform the 328x printing capability in a TN3270 environment without requiring a change to the application.

Chapter 3. DB2 Server Version 7 new functions/enhancements

DB2 Server for VSE & VM Version 7 has been improved by adding the capability to use Distributed Relational Database Architecture (DRDA) with TCP/IP for VSE/ESA, as well as SNA. This chapter provides a brief overview of the benefits of DRDA.

3.1 The DRDA environment

The Distributed Relational Database Architecture (DRDA) provides a full SQL client-server environment through the use of application requesters and application servers. DRDA environments provide the support to access data that is distributed across multiple installations using a variety of platforms.

We recommend that you install the DRDA code as soon as possible after the installation of the base code *unless* you are not planning to use the DRDA support. The DRDA code can also be removed at a later date if required. For performance reasons, if you do not plan to utilize DRDA support, you should not install the feature.

The DB2 application server, with DRDA support, should be executed within a partition of at least 15 MB in size.

DRDA support and the VSE Guest Sharing facility cannot be used within the same DB2 application server.

3.1.1 Benefits available with the DRDA environment support

When DB2 Server for VSE/ESA & VM Version 7 has been installed with DRDA support, there are a number of added benefits available that are not provided with previous connection protocols such as APPC.

- A DRDA environment allows DB2 Server for VSE data to be accessible by remote users equipped with a DRDA application requester on other platforms, for example, OS/2, AIX, OS/390, OS/400 and Windows NT.
- It provides access to data on remote application servers to online CICS application requesters running on the VSE host.
- The new SHOW CONNECT operator command provides information on the status of local and remote connections to the VSE host DB2 application server.
- It provides improved diagnostic information for IBM service personnel, thus assisting in faster problem diagnosis.
- It has the ability to execute stored procedures on request from a remote application requester. This protects host variables and data from even the most sophisticated workstation user. Business logic can be stored within the server rather than across multiple application programs. This makes application maintenance simpler.
- It provides database administrators (DBAs) with the ability to execute ISQL commands from the local host on a remotely connected DB2 database.

3.1.2 Added DBA responsibilities in a DRDA environment

A number of new responsibilities for DBAs are introduced with the ability of remote users to access local data on the VSE host and for VSE user applications to access remote databases:

- Because communication between database managers can be in different countries and timezones, DBAs need to make allowances for task scheduling and communication difficulties.
- DBAs who are responsible for a distributed database environment may need to understand transport protocols and remote interfaces, including terminologies used by these protocols and non-VSE-based DB2 environments.

3.1.3 Application considerations within a DRDA environment

Applications may require modification to allow for the different operating environments of the remote database managers. (For example, apart from the collating sequence differences between ASCII and EBCDIC, collating sequence differences can occur between two database managers, both using EBCDIC. The same character can appear in a different position, depending upon how the system processes information.)

Chapter 4. System setup and configuration

This chapter shows the system setup and configuration of the VSE/ESA e-business connectivity environment. It is not meant to show all features and setup possibilities provided with the products, but it will provide all information necessary to install and configure the products so that you can use VSE/ESA Version 2.5.0 together with e-business technology.

4.1 The 2-tier environment

In a 2-tier model, the Web client requests an HTML page or an applet from an HTTP server on the VSE/ESA host. The HTTP server sends the Web page or the applet code to the Web client. The Web client runs the applet and displays the Web page together with the requested data.

Figure 4 shows the 2-tier architecture in the VSE environment.

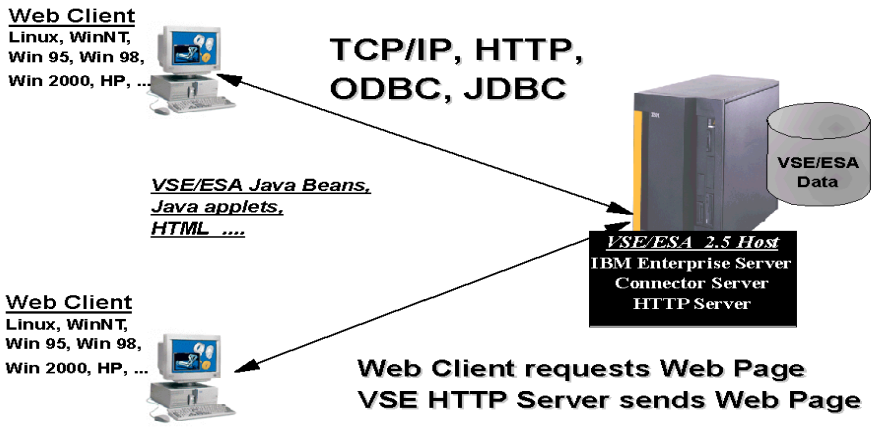


Figure 4. 2-tier client/server architecture

HTTP sessions are used between the Web client and the VSE/ESA host for sending and receiving data.The 2-tier environment is generally suitable for intranet solutions only (there is no firewall necessary). In 2-tier environments, the Web client and VSE/ESA host communicate with each other.

For the 2-tier environment you can use:

- The Java-based connector
- The CICS Web Support feature

4.2 The 3-tier environment

In 3-tier environments, the Web client and the VSE/ESA host communicate with each other via an intermediate tier called the middle tier. The middle tier provides access to VSE/ESA data and applications.

The Web client requests an HTML page or an applet from the IBM HTTP Server running on the middle tier. The IBM HTTP Server sends the HTML page and the applet to the client and forwards the request data from the VSE/ESA server.

Figure 5 shows the 3-tier architecture in the VSE environment.

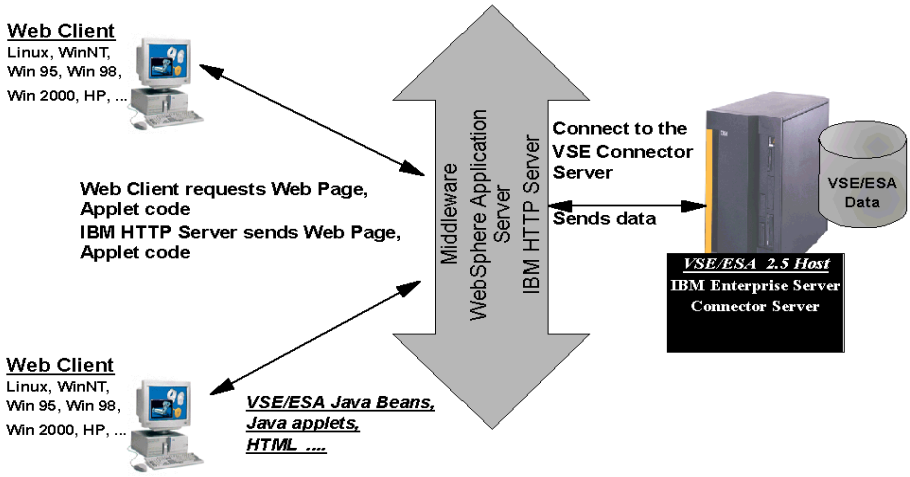


Figure 5. 3-tier client/server architecture

The 3-tier architecture is more flexible and scalable than the 2-tier architecture.

HTTP sessions are used between the Web client and the middle tier for sending and receiving data. Connect sessions are used between the middle tier and the VSE/ESA host for sending and receiving data.

For the 3-tier environment you can use:

- The Java-based connector
- The DB2-based connector
- CICS connectivity
- MQSeries connectivity

4.3 Installation overview

e-business applications for VSE/ESA Version 2.5.0 are built with Java and/or DB2 stored procedures to allow any clients to access your VSE/ESA data without having to make any changes to the existing data structure. This section provides a series of global steps that were used to install the systems.

4.3.1 Visual layout

Figure 6 shows the implementation overview in our environment.

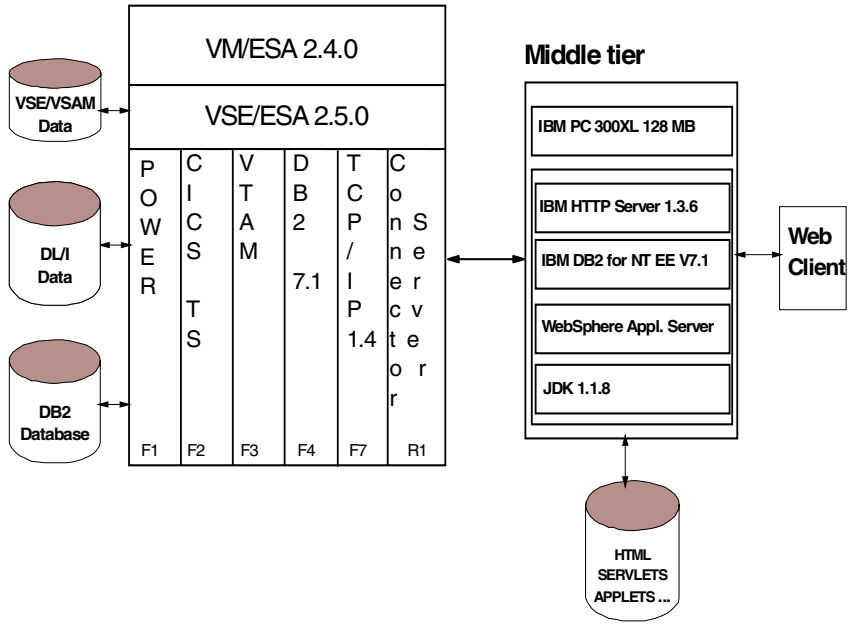


Figure 6. System layout

The VSE e-business connectors environment was located in the IBM development laboratory in Boeblingen, Germany. The major goal of the setup was to always use TCP/IP connections to VSE systems.

Notice the difference between the 2-tier and the 3-tier implementation and the necessary products for both environments.

Following is a list of the major components that we used for our environment:

On the host site:

- VSE 2.5.0 (including the VSE Connector Server)
- TCP/IP for VSE/ESA 1.4 (including the HTTP Server)
- DB2 Server for VSE & VM V7.1 (only for DB2-based connector)

3-tier environment (middle tier)

- TCP/IP
- DB2 Connect for access to DB2 Server for VSE & VM DB2 databases
- DB2 for Windows NT V7.1
- Java Development Kit (JDK) 1.1.8
- WebSphere Application Server 3.0 for NT which plugs into the HTTP Server for dynamic content generated by servlets
- IBM HTTP Server 1.3.6, which serves static HTML pages to browsers

The 3-tier environment is represented by the WebSphere Application Server at the middle tier.

2-tier environment (client installation)

- TCP/IP
- Netscape 4.7
- JDK 1.1.8 (only for Java-based connector)
- VSE Connector Client (only for Java-based connector)
- VSE/ESA Navigator (only for Java-based connector)

4.3.2 TCP/IP installation notes

You require TCP/IP for VSE/ESA 1.4 in order to use either of the VSE/ESA e-business connectors:

- The Java-based connector
- The DB2-based connector

4.3.2.1 License product code

When IBM ships TCP/IP for VSE/ESA, a product code is included that permits you to run in demonstration mode. While you are in demonstration mode, you can run one session of each type of daemon at a time. The demo version shuts down automatically after each hour of execution and is disabled entirely within 45 days.

When you license TCP/IP for VSE/ESA, you receive permanent product codes. Your license allows you to run on backup CPUs, in disaster recovery situations, and so on.

4.3.2.2 TCP/IP for VSE/ESA installation

TCP/IP is included in VSE/ESA Version 2.5.0 in a demonstration version. You must purchase and then configure TCP/IP for VSE/ESA. For details on how to activate and configure TCP/IP for VSE/ESA, refer to the IBM Redbook *Getting Started with TCP/IP for VSE/ESA 1.4* , or to *TCP/IP for VSE/ESA - IBM Program Setup and Supplementary Information*. This publication is also available online at the VSE/ESA Home page:

`http://www.s390.ibm.com/vse/`

4.3.2.3 TCP/IP for VSE/ESA customizing

You can customize TCP/IP for VSE/ESA natively on a VSE/ESA system, or you can use the VSE Workdesk tool TCP/IP for VSE/ESA Configuration Support on a Windows or OS/2 system. We used the VSE Workdesk tool, which you can download from:

`http://www.S390.ibm.com/products/vse/support/tcpip/iestcp.htm`

The configuration tool supports TCP/IP for VSE/ESA commands to set up initialization members, creates the TCP/IP for VSE/ESA startup job, and creates the VTAM B-book to make APPL statements for each defined Telnet daemon. For a listing of the initialization member EBUSVSE.L that we used in our environment, see Appendix A.1, “TCP/IP for VSE/ESA V1.4 customizing” on page 87.

Following are the customization steps:

1. Start the TCP/IP for VSE/ESA Configuration dialog.
2. At first you define your network parameters as shown in Figure 7:

- **IP Address** specifies the VSE IP address (9.164.155.55).
- **Subnet mask** specifies your subnet mask (255.255.224.0).
- **Max segment** allows you to limit the size of a TCP data segment that will be accepted by TCP/IP for VSE/ESA (recommended value 700).

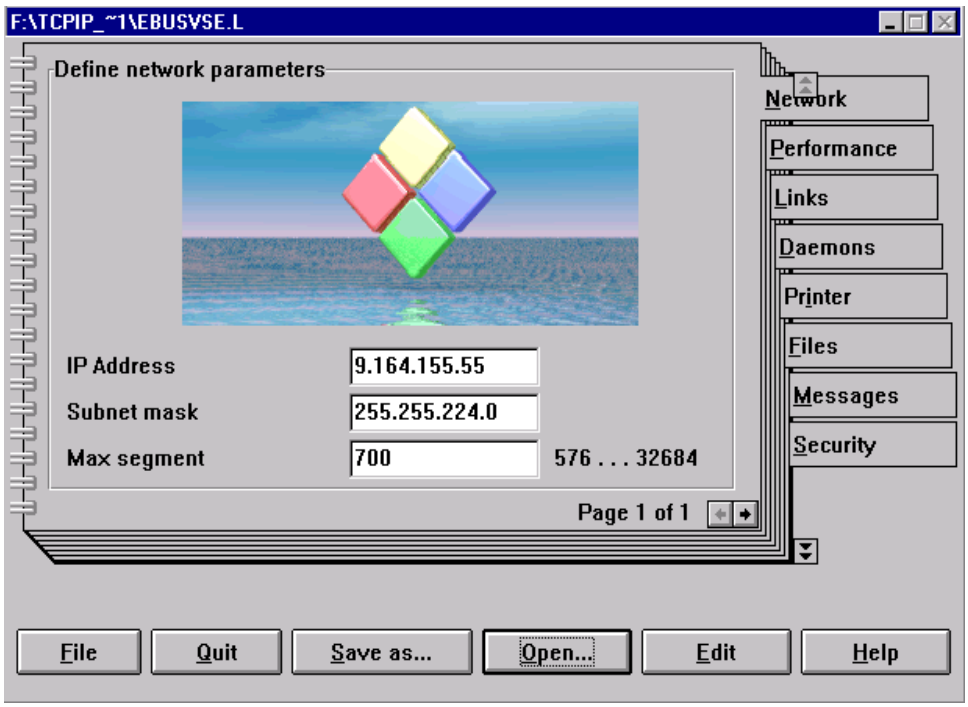


Figure 7. Define network parameters

3. Change the window size from `65535` to `8192` in the Performance panel. The window size specifies the amount of data in bytes that can be transmitted.
4. Define the communication links in the Links panel. In our environment, we used a CTCA link to a VM/ESA system.
5. Define the HTTP daemon in the Daemons panel, as shown in Figure 8 on page 18. The HTTP daemon is necessary for a 2-tier implementation or for applet testing.
   - Click **ADD** and define your HTTP daemon:
     - **ID** identifies this definition (HTTPD_ID1*).*

- **Root** is the public name of a library and sublibrary to be used by the daemon (HTTPLIB.ROOT).

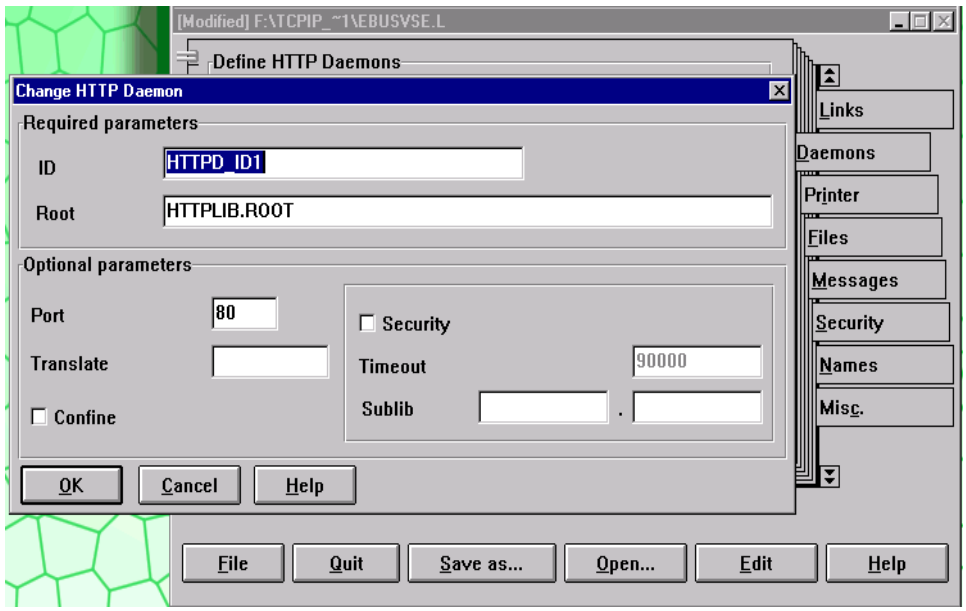


Figure 8. HTTP daemon

- Press **OK**.
  Note: The library must already be defined. For a sample job, see Appendix A.2, “Define the HTTP library” on page 89.

6. Now you can define FTP and Telnet daemons, although this is not necessary for the VSE e-business connectors.
7. In the File panel, define the HTTP source library for the HTML documents, the applets, and so on.
   - **Public name** is the unique name by which this dataset will be known to remote users (HTTPLIB).
   - **Type** specifies the type of link or file (LIBRARY).
   - **DLBL** specifies the name of the dataset as known to VSE (HTTPLIB).
   - Press **OK**.

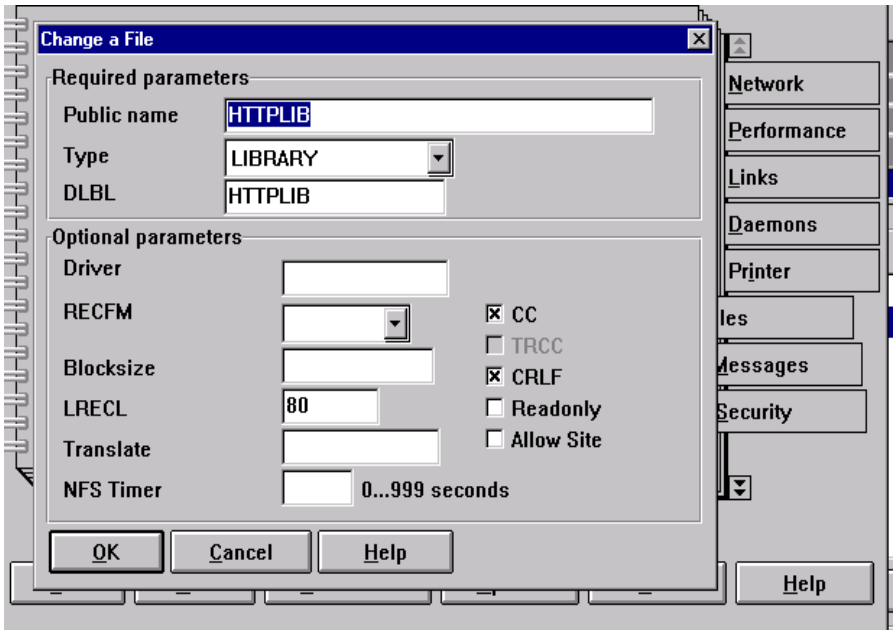


Figure 9. The TCP/IP file definition

- The next step is on page 3 in the File panel Change a File, which is shown in Figure 9.
- **Public name** for this field, select the library that you defined in the previous step.
- Press **OK**.

8. Now press the **File** button, to save the TCP/IP configuration.

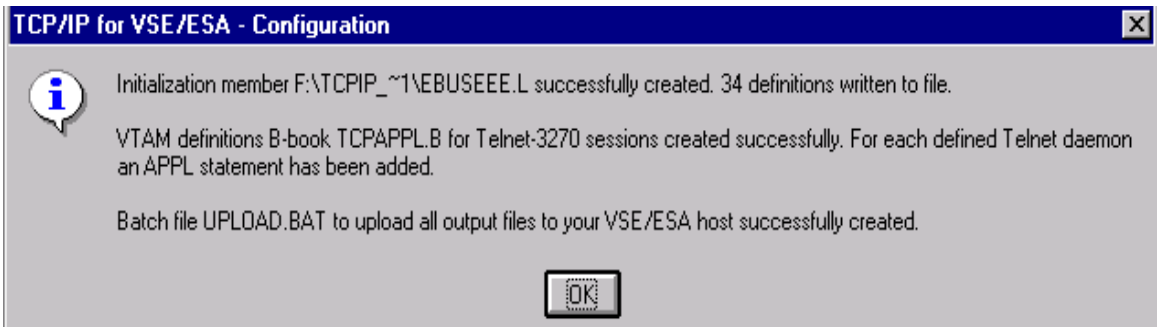


Figure 10. TCP/IP for VSE/ESA configuration

Now you can upload the configuration file to your VSE system with the upload.bat program.

4.3.3 DB2 Server for VSE V7 installation notes

The recommended installation method for DB2 is to use the supplied Job Manager process as this will automate the largest part of the installation process. Be explicit about the option you want to select , since the displayed default may not always do what is expected. After you have punched out the ARISIMGJ job,

you may want to submit it on DISP=L so that re-running failed steps, or verification programs, is easier.

Important

This redbook does not replace the Program Directory DB2 for VSE Version 7, but simply points out helpful information we noted during the installation process followed for this section. The information conveyed here should be used in conjunction with the supplied Program Directory.

Perform the DB2 library restore before starting any other steps. The sample JCL provided in Preparation Step 5 of the Program Directory for DB2/VSE V7 should be used.

9. Carefully tailor the ARISIVAR.Z before starting the Job Manager installation process. Correct variable settings and spelling will ensure that a problem-free, automated installation can be done.
10. Execute the complete Preparation step using the Job Manager. Continue with the required Installation and Migration steps for your installation configuration.
11. Ensure you have a current product key available (either a demo key, or a full function key). You will need this before you will be able to activate the DB2 server manager in multiuser mode after the installation is complete.
12. If the CICS system you are planning on using with DB2 has an APPLID (application identification) other than DBDCCICS, you will need to tailor job ARIS080D before executing.
13. If you plan to use the DRDA support with TCP/IP connectors, ensure that you tailor the DB2 Application Server startup JCL to have a TCPPORT > 0, RMTUSERS > 0 and NCUSERS > 0 (or > 2 if ISQL from CICS is required). Verify that TCP/IP for VSE/ESA is running in another partition before activating the DB2 Application Server.
14. At time of writing, we were also aware of a problem with ISQL and CICS TS SIT settings. To get ISQL to function correctly in a CICS TS system, it is necessary to set STORPROT=NO and SEC=NO in the DFHSIT. Without this, the CIRB transaction may fail with an 0C4 and the ISQL transaction may fail with an abend code of blanks.

4.3.4 Connector Server configuration notes

The VSE Connector Server is an application that runs in batch in dynamic class R by default and which implements a TCP/IP connection.

The startup job STARTVCS is loaded into the VSE/POWER reader queue during initial installation of VSE/ESA Version 2.5.0. You can find the startup job skeleton and all other VSE Connector Server members in the VSE/ICCF library 59.

SKVCSCFG:This member specifies the general settings for the VSE Connector Server (catalog in PRD2.CONFIG as IESVCSSRV.Z).

SKVCSLIB:This member specifies the VSE libraries that can be accessed (catalog in PRD2.CONFIG as IESLIBDF.Z).

SKVCSUSR:This member specifies the users or groups of users, who can logon to the VSE Connector Server (catalog in PRD2.CONFIG as IESUSERS.Z).

SKVCSPLG: This member specifies the server plugins to be loaded during startup of the VSE Connector Server (catalog in PRD2.CONFIG as IESPLGIN.Z).

You should catalog these members in the PRD2.CONFIG library as Z-books.

Normally you can use the defaults in the configuration members for the test between VSE Connector Client and Server. By default, the server runs in a partition of dynamic class R.

Note: When started, the VSE Connector Server listens for incoming TCP/IP traffic on port 2893.

Figure 11 shows the startup job STARTVCS.

```
* $$ LST CLASS=A,DISP=D
// JOB STARTVCS START UP VSE CONNECTOR SERVER
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE,PRD2.SCEEBASE)
// ID USER=VCSRV,PWD=VCSRV
// EXEC IESVCSRV,PARM='DD:PRD2.CONFIG(IESVCSRV.Z)'
/*
/&
```

Figure 11. Connector Server startup job STARTVCS

Figure 12 shows the console log after the startup job STARTVCS has been submitted.

```
R1 0045 // JOB STARTVCS START UP VSE CONNECTOR SERVER
         DATE 08/16/2000, CLOCK 08/49/02
 R1 0045 IESC1001I BEGINNING STARTUP OF VSE CONNECTOR SERVER
 R1 0045 IESC1011I USING CONFIG FILE:            DD:PRD2.CONFIG(IESVCSRV.Z)
 R1 0045 IESC1012I USING LIBRARIAN CONFIG FILE:  DD:PRD2.CONFIG(IESLIBDF.Z)
 R1 0045 IESC1013I USING USERS CONFIG FILE:      DD:PRD2.CONFIG(IESUSERS.Z)
 R1 0045 IESC1014I USING PLUGIN CONFIG FILE:     DD:PRD2.CONFIG(IESPLGIN.Z)
 R1 0045 IESC1018I LOADING PLUGIN:  IESSAPLG
 R1 0045 IESC1018I LOADING PLUGIN:  IESHTOHP
 R1 0045 IESC1018I LOADING PLUGIN:  IESCOMPH
 R1 0045 IESC1018I LOADING PLUGIN:  IESVSAPL
 R1 0045 IESC1018I LOADING PLUGIN:  IESDLIPL
 R1 0045 IESC1002I FINISHED STARTUP OF VSE CONNECTOR SERVER
 R1 0045 IESC1003I WAITING FOR CONNECTIONS OF CLIENTS...
```

Figure 12. Console log of VSE Connector Server startup

When you install a 2-tier environment, or when you would like to test the connection to the VSE Connector Server, install the VSE Connector Client. See

5.1.3, “Customizing” on page 36 or check the VSE Home page at the following site for installation instructions:

http://www.s390.ibm.com/products/vse/support/vseconn/vsecon.htm

4.4 Customizing the prerequisites

In this section we describe how to customize the prerequisites for the 2-tier and 3-tier environments. The installation and customization steps involved are:

1. VSE/ESA 2.5 setup.
2. Windows NT setup.
3. Install JDK.
4. Install IBM HTTP Server.
5. Install DB2 UDB and fixpack.
6. Install IBM WebSphere Application Server.

4.4.1 VSE/ESA 2.5 setup

VSE/ESA 2.5
The standard installation of VSE/ESA 2.5 provides all resources for setting up and running the e-business connectors on the host site (VSE Connector Server), and parts to be downloaded to the client site (VSE Connector Client).

All VSAM clusters for e-business connectors will be defined automatically.

TCP/IP for VSE/ESA 1.4
In order to have comprehensive usage of TCP/IP for VSE/ESA you must apply the TCP/IP product key for licensed users.

VSE HTTP Server
The Application Pak of TCP/IP for VSE/ESA contains the VSE HTTP Server.

4.4.2 Windows NT setup

This section describes the prerequisites for the middle tier installation on a Windows NT system.

Hardware:

- Pentium II processor or better
- CD-ROM drive
- 256 MB minimum memory, 512 MB recommended
- 800 MB or more, free disk space

Software:

- Windows NT 4.0 with Service Pack (SP) 4 or 5
- Netscape Navigator 4.6 or higher or MS Internet Explorer 5
- TCP/IP and network interface support

You must have TCP/IP networking installed, and it is important that your host name remains fixed. You must also have a fixed IP address or the MS Loopback Adapter. For detailed information, see MS Windows documentation.

4.4.3 Application installation

The application installation includes all necessary product installation steps for VSE e-business connectivity.

4.4.3.1 Installing IBM JDK 1.1.7 or 1.1.8

Attention

Sun JDK 1.2 causes unexpected results if you install it before installing WebSphere Application Server and JDK 1.1.7 or JDK 1.1.8. To prevent these results, uninstall Sun JDK 1.2 before installing WebSphere Application Server.

To install IBM JDK 1.1.7 or 1.1.8, which you can obtain from the IBM Web site at `http://www.ibm.com/java/jdk/download/index.html`, do the following:

1. Run the file ibm-jdk-n117p-win32-x86.exe or jdk118rc.exe.
2. Select **Yes**, select **Next**, read the license agreement, and accept it.
3. Leave the default selections and then select **Next** twice.
4. Select **Finish**.

4.4.3.2 Installing IBM HTTP Server 1.3.6

To install IBM HTTP Server, which you can obtain from an IBM Web site, do the following:

1. If you have the product CD, run \httpd\Setup.exe. If you have the downloaded file, run the program which you have downloaded.
2. Select the language and click **OK**.
3. Click **Next**, read the license agreement, and accept it.
4. Select the installation directory.
5. Select **Yes** to create the directory. Then select **Next**.
6. Select the **Typical** installation stream, click **Next** to accept the program folder. For the user ID and password, consider using a single Windows NT user ID and password for all software packages. Also consider using a user ID that has administrative rights. Fill in the user ID and password here.
7. Click **Install** and finish the install package. *Do not* restart your computer at this time.
8. Configure the HTTP Admin server password:
   - Go to a command prompt for the x:\http directory, and enter:

     ```
     htpasswd -m conf\admin.passwd <userid>
     ```

   - Type a password, and confirm it.

Configure the ServerName of the Web server at this point, using a text editor such as Notepad, and edit the file x: \http\conf\httpd.conf. Alternatively, you can use the administration console of the HTTP Server and configure the Web server at a later time.

Open a browser and browse `http://localhost/`. You should see Figure 13:

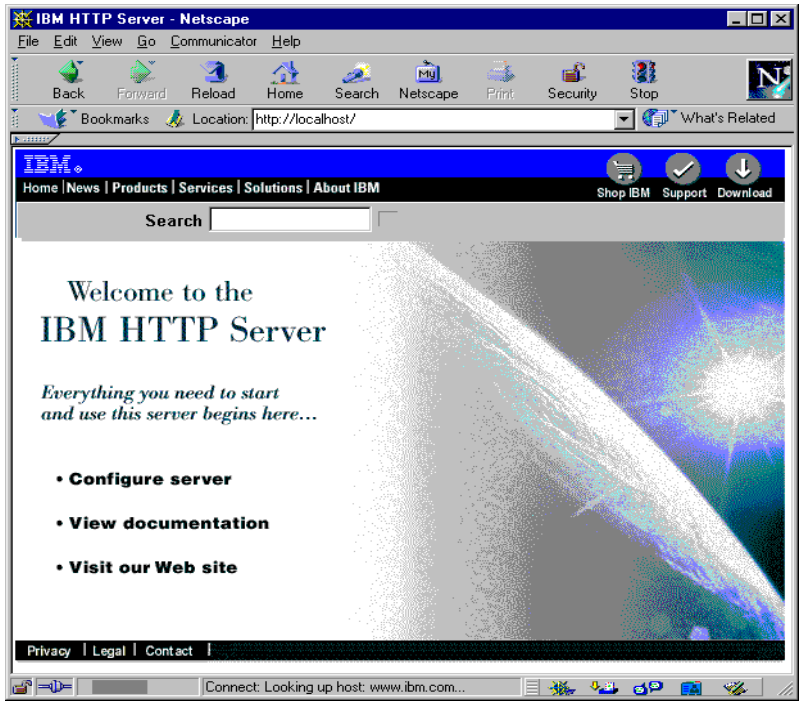


Figure 13. IBM HTTP Server

Select **Configure server**. Enter the username and password that you entered in the `htpasswd` command.

If you need to install DB2 UDB for a 3-tier environment, do not reboot yet. If you have already installed DB2 UDB, reboot your system.

4.4.3.3 DB2 Universal Database Version 7 for Windows NT

DB2 Universal Database for Windows NT is only necessary when you install a 3-tier environment as a prerequisite for IBM WebSphere Application Server.

To install the base product, perform the following steps:

1. Run db2inst.exe in the \install subdirectory for the appropriate language. For example, for English look in the \nt\db2\en\install subdirectory.
2. Select **Next**, choose the Enterprise Edition, and then select **Next**.
3. Put a check mark beside the **Typical** installation, keep the default installation directory and click **Next**.
4. Overwrite the default Username and Password with the same user ID you specified previously (if you installed IBM HTTP Server). This will make it the DB2 administration ID as well.
5. Select **Next**, and continue with the installation.
6. After the product has been installed, opt for restarting your system.

To install the DB2 UDB fixpack, now do the following:

1. From the **Services** panel of the Control Panel, stop the services DB2 - DB2 and DB2 - DB2DAS00. If you have the Netfinity Support Manager or other

monitoring agents, you may need to stop these as well, since they can lock DB2.

2. Unzip the fixpack file (downloaded from the Internet) into a temporary directory.
3. Run setup.exe from the fixpack temporary directory.
4. Click **Next**, and then **Next** again. Select the check box to always start the DB2 instance. Consider selecting the check box to start the control center.
5. Click **Next**. If needed, fill in your username and password, and click **Next** again. Then, click **Install**.

When the installation finishes, restart your computer and log in as the user you have been using to configure the software.

After your system restarts, the DB2 First Steps and Control Center dialogs may display. At this point, you may create the database sample used by WebSphere Application Server samples. To create the database sample, click **Create the SAMPLE database** in the DB2 First Steps dialog and then click **Yes** in the confirmation dialog.

4.4.3.4 DB2 Server for VSE & VM

First ensure you have completed *at least* installation steps 5, 7, 8, 9 and 10 as listed in the *Program Directory for DB2 Server for VSE*. Include any other site-required configurations that your installation may require, such as the online resource adaptor (ORA) for CICS and existing database migrations.

The DB2 Application Server should now be running in multiuser mode and the installation verification programs, using the languages you have installed, should have completed correctly.

Restriction

If you want to use the Distributed Relational Database Architecture (DRDA) support over TCP/IP, you will not be able to use the VSE Guest Sharing function on the same named database server. You can, however, define the same database server with an alias and connect to the alias using DRDA over TCP/IP. Generally, database servers defined with a connection that uses VSE Guest Sharing are considered to be local connections.

To use the DB2 connector environment, ensure that you have completed the DRDA installation process when installing the DB2 Server for VSE product.

This should include the execution of the following jobs:

- Installation Step 13 - Add New Users to The Database.

  Add the users that will be accessing the database remotely and locally in this step. The user details added here will be used by remote access through TCP/IP. You can either run the sample JCL in the program directory for DB2 for VSE after tailoring and adding your specific user IDs, or you can use ISQL, if installed, to perform the same function online.

- Installation Step 15 - System Customizing Activities.

Next you will need to link the DRDA code into the DB2 modules. The job to use is ARIS615D and should look something like the following JCL:

```
* $$ JOB JNM=ARIS615D,CLASS=0,DISP=D
* $$ LST CLASS=S,DISP=D
// JOB ARIS615D - LINKEDT DRDA SUPPORT
* STEP 1 - LINKEDT ORA WITH DRDA SUPPORT
// LIBDEF *,SEARCH=PRD2.DB2710,CATALOG=PRD2.DB2710
// OPTION CATAL
 INCLUDE ARISLKZE
// EXEC LNKEDT,PARM='MSHP'
/*
// IF $RC GT 2 THEN
// GOTO $EOJ
* STEP 2 - LINKEDT ORA DRDA ROUTER
// LIBDEF *,SEARCH=PRD2.DB2710,CATALOG=PRD2.DB2710
// OPTION CATAL
 INCLUDE ARISLKZA
// EXEC LNKEDT,PARM='MSHP'
/*
*  STEP 3 - LINK EDIT RDS WITH DRDA SERVER SUPPORT
// LIBDEF *,SEARCH=PRD2.DB2710
// LIBDEF PHASE,CATALOG=PRD2.DB2710
// OPTION CATAL
 INCLUDE ARISLKRA
// EXEC  PGM=LNKEDT,PARM='MSHP,AMODE=31,RMODE=ANY'
/*
/&
* $$ EOJ
```

You can now restart the DB2 Application Server after tailoring the Application Server startup JCL. You should receive messages similar to the following if the DRDA support has been successfully installed:

```
F4 0004 ARI0045I Ready for operator communications
F4 0004 ARI4103I TCP/IP service GETHOSTBYADDR failed. Return code = 1
F4 0004 ARI4100I TCP/IP service initialized from TCPPORT. TCP/IP Port is
446
```

Figure 14. Sample console display after DRDA support has been installed

4.4.3.5 IBM WebSphere Application Server 3.0

The IBM WebSphere Application Server is an e-business application deployment environment built on open standards-based technology. It is the cornerstone of WebSphere application offerings and services.

The Standard Edition lets you use Java servlets, Java Server Pages, and XML to quickly transform static Web sites into vital sources of dynamic Web content. The Advanced Edition is a high-performance Enterprise JavaBeans (EJBs) server for implementing EJB components that incorporate business logic. The Enterprise Edition integrates EJB and CORBA components to build high-transaction, high-volume e-business applications.

The WebSphere Application Server has two main components:

1. A Java virtual machine configuration
2. Support for a servlet engine to handle servlet requests

After installation, WebSphere is configured with the default application server.

To install WebSphere Application Server 3.0, do the following:

1. Stop the IBM HTTP Web Server (if running).
2. Run the executable that you downloaded from the Internet, or run \nt\setup.exe if you have the product CD. You will need 60 MB free in your temp directory.
3. The installation shield will start; select a language and click **OK** (CD-Version).
4. Click **Next** to pass the introductory page.
5. If you already have a JDK and database program installed, but do not have supported versions, a dialog opens informing you of the need to upgrade the prerequisite software. If this dialog is not shown, proceed to step 6, otherwise perform the upgrade first.
6. If WebSphere Application Server with the appropriate prerequisites is already installed on your system, a dialog giving you the option to upgrade WebSphere Application Server displays. Click **Next**. Then continue with the installation.
7. In the Install Options dialog, select **Custom Installation** and the installation directory, then click **Next**.
8. In the Choose Application Server Components dialog, select **IBM HTTP Server V1.3.6** on the right-hand side. Select **Configure administrative domain with default...**. This step is important because the standard installation does not install the sample servlets or configure an EJB container or servlet engine.

   You will now see Figure 15.

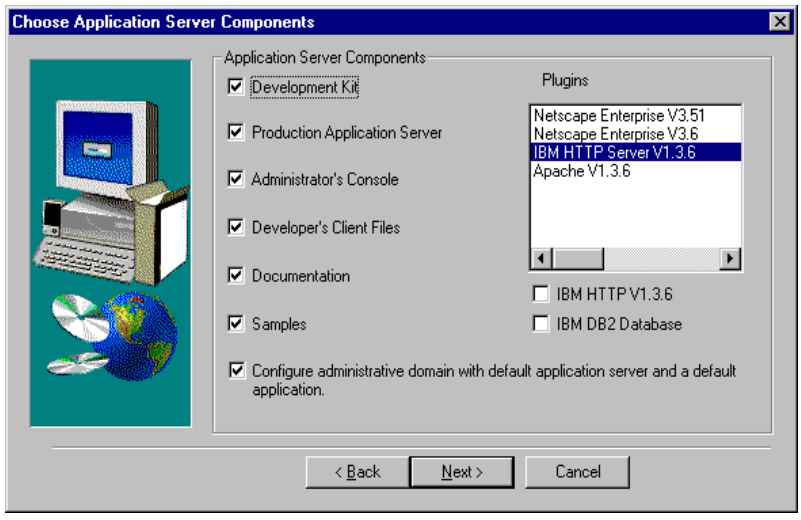


Figure 15. Choose Application Server Components

9. Click **Next**. If necessary, shut down all Web servers you plan to run with WebSphere Application Server and proceed.

10. The system should find any level of IBM JDK that is installed.
11. You are now prompted with the Security/Database Options page, which configures the database access and user IDs. The database access is important to WebSphere Application Server, because the product stores the configuration of the server in a set of specially defined tables in the database.

    Note: If you use an invalid user ID to install WebSphere, it will not successfully register the WebSphere Administrative Server to the NT services database. Fill in the user ID, security password, and confirming password; then specify the user ID and password for the database. For the database values, if you have already installed DB2 UDB, ensure that you specify the Username and Password specified when installing DB2 UDB. Then click **Next**.

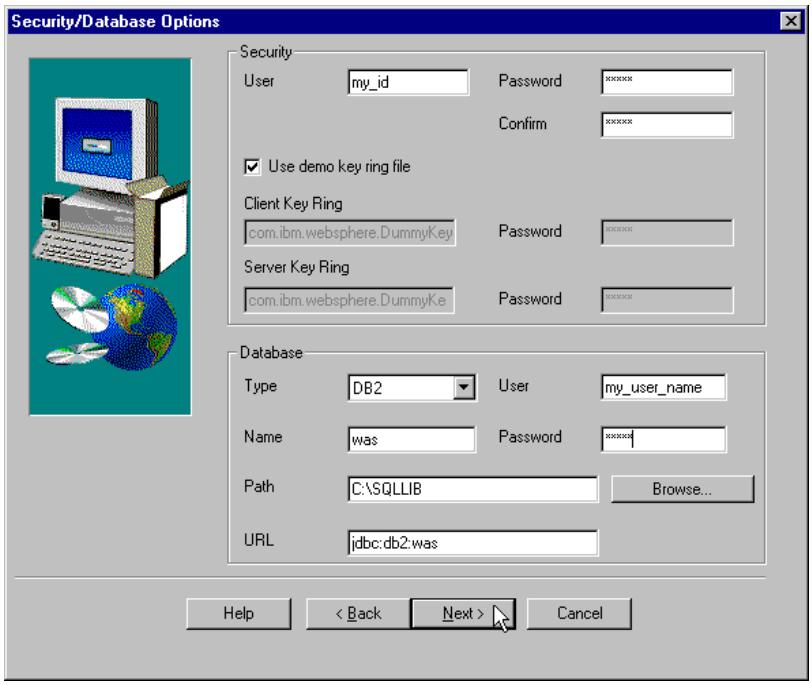


Figure 16. Security/Database Options

12. Click **Next** again to begin the installation.
13. Near the end of the installation, you are prompted for the location of the IBM HTTP Server configuration file. The installation tool should successfully find the file for you.
14. Click **OK**, and it will finish updating the files and installing.
15. The next panel points you to the README file and, if you are installing the samples, states where the samples have been installed and which server to use.
16. Click **Finish**, and choose to restart.

Finishing the prerequisite configuration

If you installed DB2 UDB as part of the WebSphere Application Server installation, the installation program should create a database to store the adminstrative configuration when your system starts up after rebooting. Thus, the installation program should create the WebSphere Application Server (WAS) database with its DB2 application heap size set to 256.

To ensure that the WebSphere Application Server (WAS) database exists, do the following:

1. Go to the Control Center dialog. If it did not open automatically after rebooting, open it as follows: From the Start menu, select:
   Programs -> DB2 for Windows NT -> Administrative Tools -> Control Center.
2. In the DB2 Control Center, expand the tree under **Systems**. Your DB2 databases are listed under **Databases**. Examine the list to see if WAS is in it.

Setting up the administrative configuration

If the DB2 database WAS has not yet been created, do the following:

1. From the Start menu, select: **Programs -> DB2 for Windows NT -> Command Line Processor**
2. Enter: `CREATE DATABASE WAS`
3. Wait a minute to allow time for DB2 to create the database. Enter:

   ```
   UPDATE DB CFG FOR WAS USING APPLHEAPSZ 512
   ```

4. Type `quit` to leave the command line processor (CLP), and then `exit` to finish the command prompt.
5. Restart the machine.

Testing the WebSphere installation with the snoop servlet

To start WebSphere Application Server for the first time, you need to restart the IBM HTTP Server. In our case, we stopped and restarted the IBM HTTP Server with Start and Stop the Services. Then we were able to access our server to run the snoop servlet as a sample that is delivered with WebSphere.

The Web Application Server (WAS) execution environment is shown in Figure 17 on page 30.

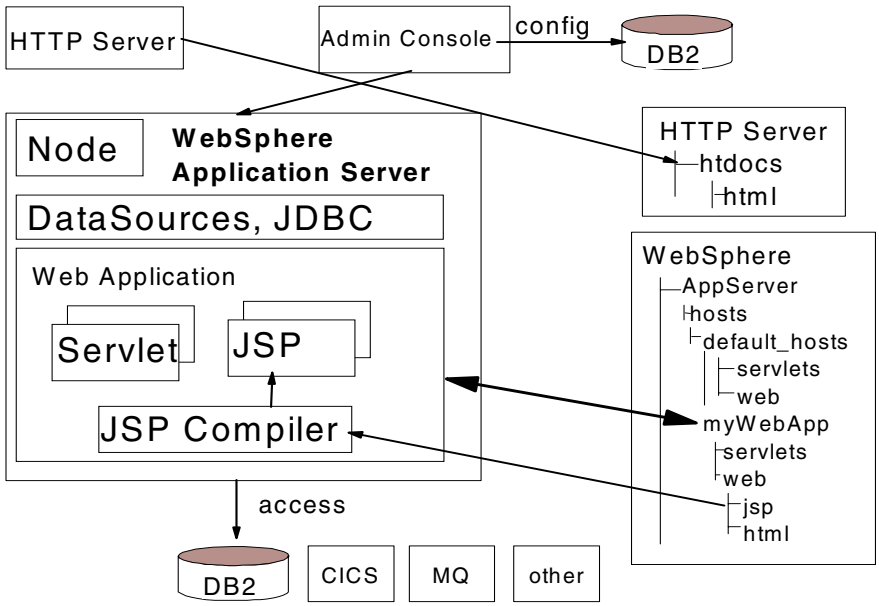


Figure 17. WAS execution environment

Here is a short description of the major components of WAS:

- WAS contains a plugin for many HTTP servers, including the IBM HTTP Server.
- WAS runs on a node (TCP/IP host name).
- JDBC drivers and DataSources describe how relational databases are accessed from servlets.
- Multiple servers, each with a servlet engine, run on the node.
- Servlets and JSPs are grouped into Web applications. A server contains multiple Web applications.
- An administrative console program maintains the configuration of the Application Server in a DB2 database. The configuration contains information about servers, Web applications, servlets, EJBs, the JSP compiler, DataSources, and other resources.
- The right side of the diagram shows the directory structure for the HTTP server and the Application Server. The HTTP Server directory contains static HTML files. The Application Server directory contains the directories for the Web applications, such as the default application and any user-defined Web applications. For each Web application, there is normally a servlets subdirectory and a Web subdirectory.
  - The servlets subdirectory contains executable code for servlets and JavaBeans.
  - The Web subdirectory contains HTML and JSP files. HTML files are served to a browser by a special file-handling servlet. JSP files are compiled into servlets the first time they are invoked.
- The code in Web applications can access enterprise resources, such as relational databases (DB2), CICS, MQSeries and others. This is normally

done using the Common Connector Framework and WebSphere connection pools.

If you need more information, see *Installing WebSphere Application Server 3.02 Advanced Edition for Windows NT Using DB2 and IBM HTTP Server* at:

```
http://www.ibm.com/software/webservers/appserv/doc/v302/install/install_guides
.html
```

4.4.3.6 DB2 Connect and DB2 Client Application Enabler

To establish a connection from the client to your VSE/ESA system for testing, you have to install DB2 Connect Version 7 Release 1 on your middle tier. DB2 Client Application Enabler is delivered together with DB2 Connect and can be installed at the same time.

Follow the default installation path and reboot the Windows NT system. After successful installation, you should have the Client Configuration Assistant; select this program item. Perform the following steps in order to specify the parameters for the connection:

1. Click **Add Database**. Continue with **Next.**
2. Select **Manual Configuration...** on the panel titled 1. Source. Continue with **Next.**
3. Specify the communication protocol **TCP/IP** to access the database server DB2 Server for VSE & VM. Select **The database physically resides on a host or AS/400 system**. Then select a connection route: **Connect directly to the server** (default), because DB2 Connect is installed on your workstation. Continue with **Next.**

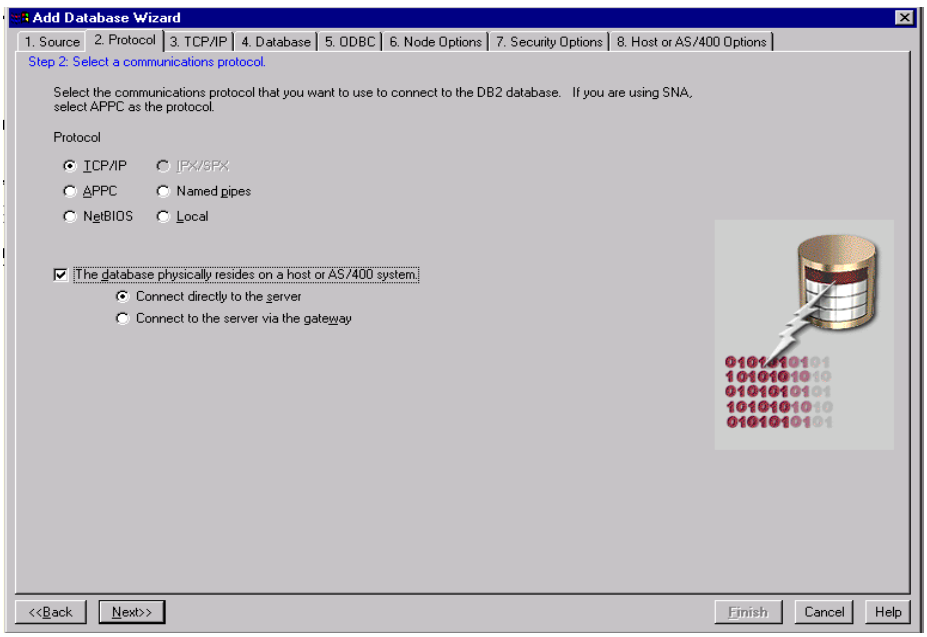


Figure 18. Add Database Wizard

4. Specify the Hostname or IP address of your VSE system on the panel titled 3. TCP/IP. In our environment, the address is 9.164.155.55.

Now you have to specify the port number used to establish a connection between DB2 Server for VSE & VM and TCP/IP for VSE/ESA. In our environment, it is 446.

5. On the screen titled 4. Database, specify the name of the DB2 database on VSE. Change the default alias name to your preferred alias name. Continue with **Next**.
6. Use the default values on the panel titled 5. ODBC. Continue with **Next**.
7. Specify the Operating system VSE on the screen titled 6. Node Options. Continue with **Next**.

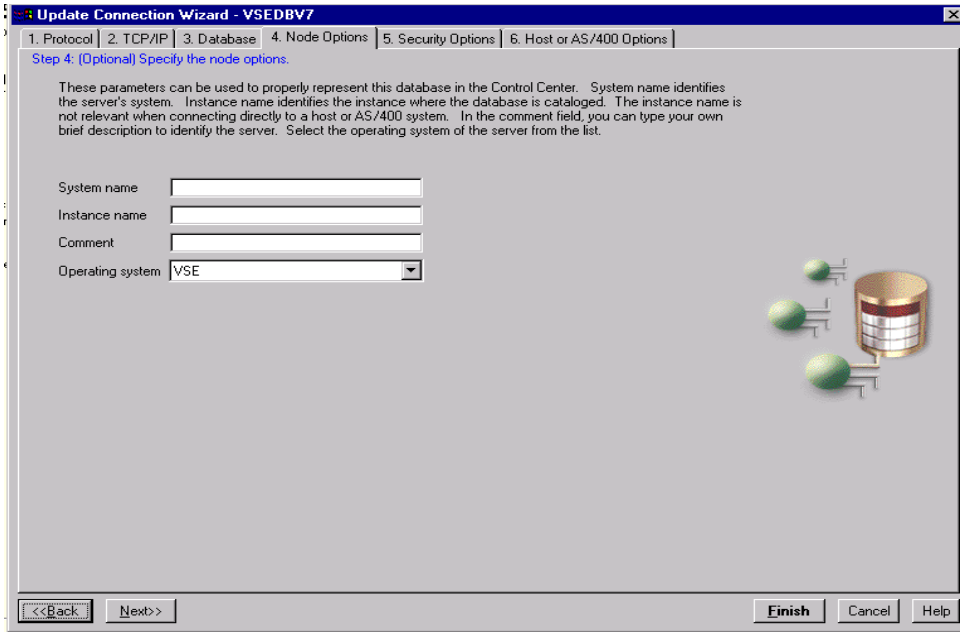


Figure 19. Node options

8. Specify the security option on the screen titled 7. Security Options. Select **Configure security options**.
9. Finish by selecting **Finish**.
10. You can directly test the connection by clicking the **Test** button.
11. Specify a user ID of a user who is authorized to access your DB2 database.

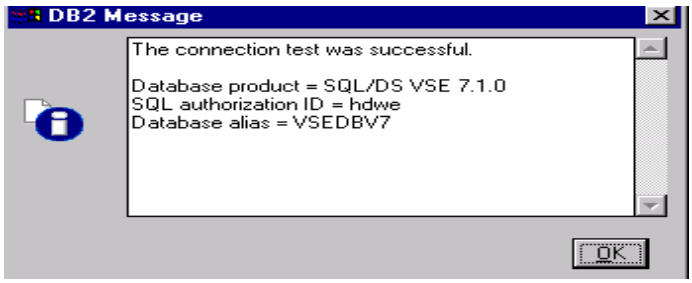


Figure 20. Successful connection message

12. The panel shown in Figure 20 will be displayed if the test is successful.

In order to establish a connection from a Java applet to DB2 Connect, the JDBC applet driver has to be started. It has to be started with a TCP/IP port number that is not yet used in your system.

The command to start the JDBC applet driver with the default port is:

```
DB2JSTRT 6789
```

Chapter 5. Building e-business connectivity

In this chapter we describe how to build connectivity to your VSE/ESA host using the different connectors available with VSE/ESA 2.5.

5.1 The Java-based connector

In order to exploit the Java-based connector you need to set up connectivity between a client and your VSE/ESA host, as well as prepare the sample Java applets for execution.

5.1.1 Design and architecture

The Java-based connector consists of a client part (the VSE Connector Client), and a server part (the VSE Connector Server). Figure 21 shows how these components fit into the overall e-business connector design for VSE/ESA.

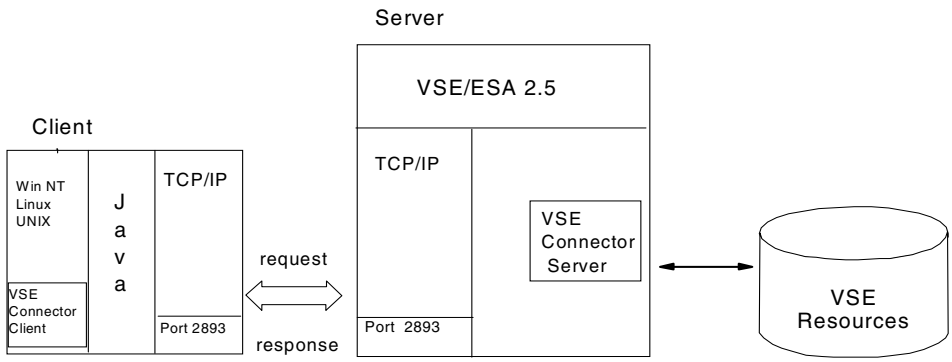


Figure 21. e-business connector design

The VSE Connector Client resides on a Web client. The VSE Connector Client has the following components:

- VSE JavaBeans
- Online documentation
- Samples
- Online programming references

The VSE Connector Server is the counterpart application on the VSE/ESA host side, and runs permanently in a VSE/ESA partition. The backbone of the client-server connection is a set of VSE JavaBeans.

To use the VSE Connector Client Java-based services you need to add an entry for a file named VSEConnector.JAR to your VSE Connector Client local classpath variable. This file consists of the VSE JavaBeans class library. Java applications or applets running on the Web client need the bean class library to provide functions like access to VSE/ESA file systems (VSE/VSAM, VSE/Librarian, VSE/POWER and VSE/ICCF) and the operator console.

5.1.2 Connector functionality

The Java-based connector allows you to:

- Access all VSE library systems
- Submit jobs to a VSE/ESA host for execution
- Handle the operator console
- Use VSE/ESA as an HTTP server

You can perform these tasks from the *client side* by using Java applications and Java applets.

5.1.3 Customizing

You need to customize the VSE Connector Client as follows before you can access VSE/ESA data.

5.1.3.1 Install and configure the VSE Connector Client

Before performing any installation activities, you should ensure that the required versions of the Java Virtual Machine (JVM) and Java Development Kit (JDK) are on your client system.

In a command prompt window type: `java -version`.

The response should be `java version "1.1.7"` or even higher for a VSE Java-based connector.

If JDK is not yet installed on your client site, you may download the latest code from:

`http://www.ibm.com/java/jdk/`

The required Java Swing Classes are included in the VSE Connector Client, which you can download from VSE/ESA.

Next, run the JDK installation step by step and check that the correct classpath has been set by the installation program or by yourself.

In a command prompt window type: `SET CLASSPATH`

The response message for the classpath definition should look like this:

```
classpath=.........;d:\jdk117\lib\classes.zip;..
```

Now you can try to compile one of the Java demos provided by the JDK package by typing the following command string from any directory path:

```
javac D:\jdk1.1.7\demo\TicTacToe\javac TicTacToe.java
```

If this runs error-free, your Java VM and JDK environment is probably set up correctly and the next step can be executed.

You can use either the IND$FILE RECEIVE command or the TCP/IP FTP utility to obtain a copy of the VSE Connector Client package file from VSE/ESA:

- Using the IND$FILE RECEIVE command, issue the command.

- Using the TCP/IP FTP utility, download the file IESINCON.W from VSE/ESA library PRD1.BASE, making sure that you use BINARY mode and that UNIX=OFF is set, especially if you are running FTP in prompt mode.

Figure 22 shows the use of the FTP under TCP/IP in prompt mode.

```
C:\tempdir>ftp 9.164.155.55
Connected to 9.164.155.55.
220-TCP/IP for VSE -- Version 01.04.00 -- FTP Daemon
    Copyright (c) 1995,2000 Connectivity Systems Incorporated
220 Service ready for new user.
User (9.164.155.55:(none)): res2
331 User name okay, need password.
Password:
230 User logged in, proceed.
ftp> cd\                                    <==== set UNIX=OFF
200 Command okay.
ftp> binary                                 <==== set BINARY transfer mode
200 Command okay.
ftp> cd prd1
250 Requested file action okay, completed.
ftp> cd base
250 Requested file action okay, completed.
ftp> get iesincon.w
200 Command okay.
150-File: PRD1.BASE.IESINCON.W
    Type: Binary Recfm: FB Lrecl:    80 Blksize:   80
    CC=ON UNIX=OFF RECLF=OFF TRCC=OFF CRLF=ON
150 File status okay; about to open data connection
226-Bytes sent:    5,350,960
    Records sent:     66,887
    Transfer Seconds:     18.22 ( 290K/Sec)
  File I/O Seconds:      6.73 ( 870K/Sec)
226 Closing data connection.
5350960 bytes received in 18,81 seconds (284,52 Kbytes/sec)
ftp> quit
C:\tempdir>ren iesincon.w install.class        <==== RENAME
```

Figure 22. FTP Get command

Figure 23 shows the use of the IND$FILE RECEIVE command.

```
RECEIVE C:\tempdir\install.class h: IESINCON W (FILE=LIB BINARY -
    NOCRLF L=PRD1 S=BASE

(h: = your hostsession ID)
```

Figure 23. IND$FILE RECEIVE command

Remember to rename the file IESINCON.W to install.class, if you have not already done so. As all Java-related files are case sensitive, you should ensure that the new file name is in lower case.

Before starting the installation program, you must ensure that the current directory is part of the local classpath environment. To have the current directory path in front of the default classpath, enter the following command:

```
set classpath=.;%classpath%;
```

Now type: `java install` and press Enter.

The installation process begins, and you are guided through various installation menus.

Figure 24. Components selection menu

From the Select Components to Install menu, select all three options to install the VSE JavaBeans (VSEConnector.jar), the online documentation and sample code for developing your own Java programs that can work in conjunction with Java-based connectors.

On Windows and OS/2, shortcuts are created to access the documentation. The main HTML file is VSEConnectors.html.

After installing the VSE Connector Client, you will need to start a Web browser and open the file VSEConnectors.html in the newly created home directory. You can access all samples and source code in the online documentation.

Before you can run any of the examples, or even your own Java program that uses VSE JavaBeans, you must have the VSEConnector.jar file in your local classpath variable.

To accomplish this, once again in a command prompt window, type: `SET CLASSPATH`

The response message for the classpath definition should now look like this:

```
classpath=.........;c:\vsecon\VSEConnector.jar;......
```

5.1.4 Using the connector to access data

Now that you have set up the connector, you can use it to access data.

5.1.4.1 Java applets

Java applets are Java programs that run inside the Java Virtual Machine of a Web browser. The typical processing flow would be as follows:

1. The client's Web browser requests an HTML page from the VSE HTTP Server running on the VSE/ESA host.
2. The VSE HTTP Server sends the Web page to the client's Web browser.
3. The client's Web browser reads an <applet> tag and requests the applet code, stored in one or more JAR files, and the VSE JavaBeans class library (VSEConnector.jar) from the VSE HTTP Server.
4. The VSE HTTP Server sends the applet.JAR and VSEConnector.JAR to the client's Web browser.
5. The client's Web browser runs the applet.
6. The applet uses the VSE JavaBeans class library (VSEConnector.JAR) to build a connection to the VSE Connector Server.
7. The end user may request data from a VSE/ESA host. In a 2-tier environment, an applet has access to VSE/POWER, VSE/VSAM, VSE/ICCF and VSE/Librarian data.

Before setting up the sample applet, you should examine the online documentation as follows:

1. Start the Web browser and open the VSEConnectors.html on the client's home path.
2. Click the **Applets** entry in the left frame of the page.
3. A submenu appears saying Applets - An Overview.
4. Read the information, particularly regarding the VSE-related restrictions, and scroll down to the three links to applet-related examples. Click **An applet to maintain VSE/VSAM maps and views**.

Here you will find the detailed information for our sample applet called VsamMappingApplet. This applet may be used to map VSAM records from the standard format to a relational structure, which is required for VSE JavaBeans to access VSAM data.

These steps will guide you in setting up the environment for the sample applet VsamMappingApplet:

1. Establish the HTTP daemon on the VSE/ESA host by issuing the TCP/IP command:
   ```
   DEFINE HTTPD,ID=HTTPD_ID1,ROOT=HTTPLIB.ROOT
   ```
2. Create the VSAMMAP.HTML file that will be loaded when a Web browser connects to this VSE/ESA host. This file is shown in Figure 25 on page 40.

```
<head>
<title>VSAM Data Mapping Example Applet</title>
</head>
<body>
<h2>VSAM Data Mapping Applet</h2>
This applet can be used to create and maintain VSAM maps and views.
Please logon to your VSE host. When the connection is established, a
list of VSAM catalogs is displayed.
<p>
<center>
<applet code="com.ibm.vse.samples.VsamMappingApplet" width=440
height=420 archive="applets.jar, vsecon.jar">
</applet>
</center>
</body>
</html>
```

Figure 25. VSAMMAP.HTML

Remember that the Web browser has to load the Java applet code from the VSE/ESA host. At present, the Java applet samples provided by VSE Connector Client are available on the client side only.

Before you can start the Java applet, you will need to do the following:

1. Compile the two Java sample source files VsamMappingApplet.java and VsamAppletListener.java to bytecode format (file extension: class). This is only necessary when you change the files, since these files are already compiled.
2. Create a .JAR file where both compiled files (.class files) are combined into a single file (now named applets.jar).
3. Send three files to HTTPLIB.ROOT library on the VSE/ESA host:
   - vsammap.html (in ASCII mode)
   - applets.jar (in binary mode)
   - vsecon.jar (in binary mode)

As you may notice, the file name of VSEConnector.jar has been truncated to VSECON.JAR, because the VSE Librarian supports short file names only.

Note: Steps 1 to 3 can be executed from a single file called DEPLOY.BAT, which you can copy from the online documentation (in the section “How to deploy the applet”). The DEPLOY.BAT file is described in Figure 26 on page 41.

Before starting DEPLOY.BAT, ensure that the library and sublibrary names are pointing to HTTPLIB and ROOT, respectively.

Also ensure that the execution of this BAT file is independent of any root directories by coding the following as the first line:

```
CD C:\vsecon\samples
```

The lower case a: in the CALL SEND string represents the session ID where you are logged on to your CICS terminal.

```
CD C:\vsecon\samples
call javac com\ibm\vse\samples\VsamMappingApplet.java
call javac com\ibm\vse\samples\VsamAppletListener.java
del applets.jar
call jar c0fv applets.jar com\ibm\vse\samples\VsamMappingApplet.class -
com\ibm\vse\samples\VsamAppletListener.class
call send vsammap.html a: vsammap html (file=lib l=httplib s=root crlf -
uncond
call send applets.jar a: applets jar (file=lib l=httplib s=root binary -
nocrlf uncond
call send c:\vsecon\VSEConnector.jar a: vsecon jar (file=lib l=httplib -
s=root binary nocrlf uncond
```

Figure 26. DEPLOY.BAT file

After executing the DEPLOY.BAT file, you should check that the three new members exist in the HTTPLIB.ROOT sublibrary on the VSE/ESA host.

On the client side, start the Web browser and type the IP address of the VSE/ESA host and the name of the sample HTML into the location entry field:

```
http://9.164.155.55/VSAMMAP.HTML
```

Press Enter. The VSAM Data Mapping Applet should appear as shown in Figure 27.

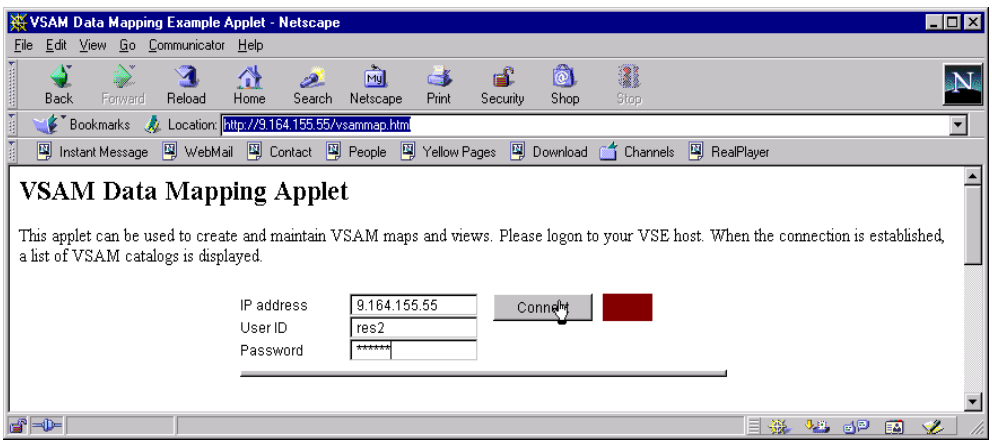


Figure 27. VSAM Data Mapping Applet

Type your logon data into the entry fields and click the **Connect** button.
It takes a little while to get the first submenu, from where you can go further step by step by selecting the **Catalog**, **Cluster**, **Maps** and so on as shown in Figure 28 on page 42.

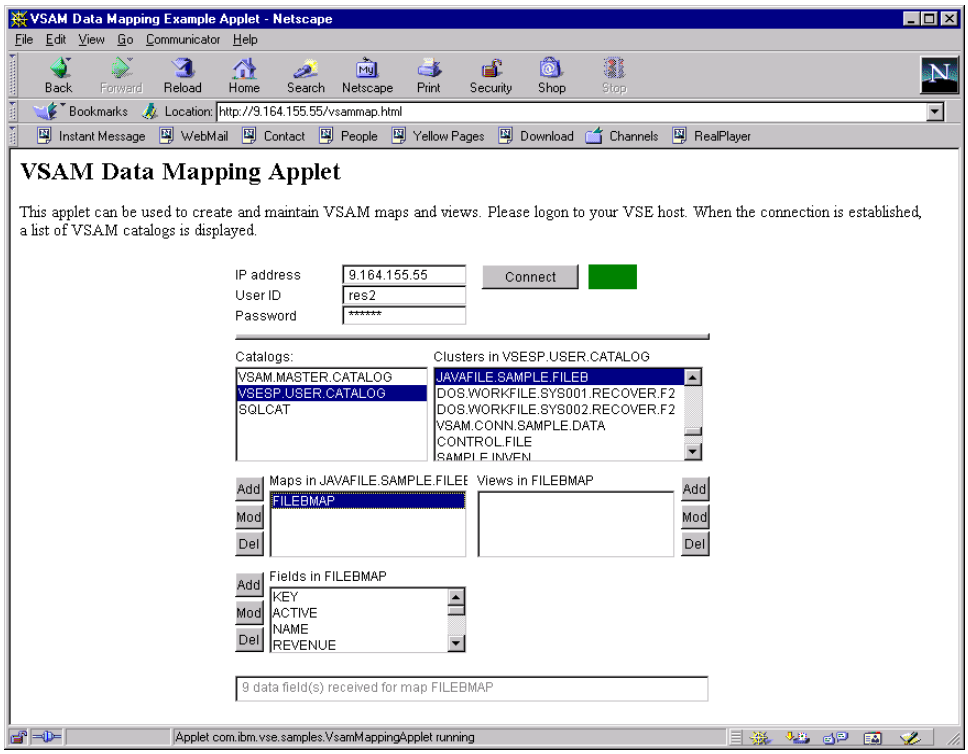


Figure 28. VSAM Data Mapping Applet

5.1.4.2 Displaying VSAM data with the Java applet vsamdisa

The Java applet vsamdisa is an example that shows how to display records from a VSAM file through a map. The applet vsamdisa displays all records from a VSAM file. When running the vsamdisa example, the input is similar to the following:

```
 x:\vsecon\samples>java com.ibm.vse.samples.vsamdisa
Please enter your VSE IP address: 9.164.155.55
Please enter your VSE file name:JAVAFILE.SAMPLE.FILEA
Please enter your VSE map name: FILEAMA1
Please enter your VSE user ID: hdwe
Please enter password:xxxxx
```

The applet vsamdisa is listed in Appendix B.2, “Java source: VSE/VSAM access and display VSE/VSAM records” on page 95. You can use this applet to test your installation and setup in the 2-tier environment.

5.1.4.3 Plugin development for the Java-based connector

The Java-based connector can be extended by user-written "plugins". There are two possibilities:

1. You can extend the VSE JavaBeans class library with your own plugins. In this case, your plugin can only request services that are already provided by the VSE Connector Server.
2. You write a plugin that consists of a client part and a server part. In this case, you can extend the server's capabilities by additional functionality needed by your client plugin.

The client side part consists of one or more JavaBeans that implement the Java class VSEPlugin which is part of the package com.ibm.vse.connector. The server side part of a user-defined plugin (VSE/ESA Connector Server Plugin) is a VSE phase written in C for VSE/ESA. This phase is loaded during the VSE Connector Server startup and has to implement a well-defined interface which enables the VSE Connector Server to setup and call the plugin.

In general, a server plugin consists of a set of callback routines that are called by the VSE Connector Server in a given sequence. Some plugin functions are called during the startup of the server, and others are called when a request is received for a given plugin. A cleanup function is called when the server terminates.

For more information, refer to the online plugin development document that is included in the Java-based connector package with VSE/ESA 2.5.

5.1.4.4 Java servlets

Servlets are used in the middle tier of a 3-tier environment. In this way the HTTP sessions are used between the Web client and the middle tier for sending and receiving data. Socket connections are used between the middle tier and the VSE/ESA host for sending and receiving data.

In our environment, the middle tier is represented by the IBM HTTP Server and the IBM WebSphere Application Server. The servlets run in the Java Virtual Machine of the IBM WebSphere Application Server and use the VSE JavaBeans to connect to the VSE Connector Server.

Servlets more closely resemble Common Gateway Interface (CGI) scripts or programs than applets, in terms of functionality.

The servlet process flow

Servlets implement a common request/response paradigm for handling messaging between the client and the server. The Java servlet defines a standard interface for the handling of these request and response messages between the client and server.

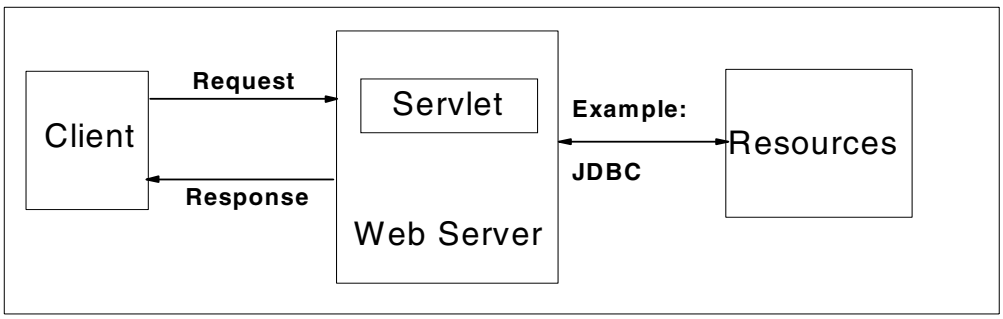


Figure 29. Client-to-servlet process flow

Figure 29 shows the process flow:

1. The client sends a request to the server.
2. The server sends the request information to the servlet.

3. The servlet builds a response and passes it to the server. That response is dynamically built, and the content of the response usually depends on the client's request. External resources may also be used.
4. The server sends the response back to the client.

The servlet life cycle

A client of a servlet-based application does not usually communicate directly with a servlet, but requests the servlet services through a Web server or a Web application server that invokes the servlet through the Java servlet API. The server's role is to manage the loading and initialization of the servlet, the servicing of the request, and the unloading or destroying of the servlet.

Typically, there is one instance of a particular servlet object at a time in the Web server's environment.The Web server is responsible for handling the initialization of the servlet when the servlet is first loaded into the environment, where it remains active for the life of the servlet.

Each client request to the servlet is handled via a new thread against the original instance object. The Web server is responsible for creating the new threads to handle the requests. The Web server is also responsible for the unloading or reloading of the servlets. This might happen when the Web application is brought down, or the underlying class file for the servlet changes, depending on the underlying implementation of the servlet. The life cycle of a servlet is expressed in the Java servlet API in the *init service* (*doGet* or *doPost*), and *destroy* methods of the servlet interface.

Servlet development

In this section we describe servlet development and servlet deployment. The key parts of the servlet are:

1. The `package` declaration defines this servlet as a part of a Java package.
2. The `import` gives us access to other Java packages; for example the import of `java.io` is done so that we have access to the standard I/O classes. The `class` declaration extends the `HttpServlet` class to make our class an HTTP servlet.
3. The heart of the servlet is the implementation of the service method for the handling of the request and response objects of the servlet.

Figure 30 on page 45 shows an extract from our servlet.

```
package com.ibm.redbook.servlet;

import java.io.*;

import java.util.*;

import com.ibm.redbook.utils.*;

import javax.servlet.*;

import javax.servlet.http.*;


public class VSAMDisplayer extends HttpServlet {


private String propfile = "";

private VSE vse = null;

private int err = 0;


  public String getServletInfo() {


    return "VSAM DIsplayer - displays VSAM data from a VSE-server";

  }


  public void init() throws ServletException {

...

...
```

Figure 30. VSAMDisplayer servlet (extract)

The complete servlet is listed in Appendix B.3, “Java source: VSAMDisplayer servlet” on page 99. For further information, see the IBM Redbook *Servlet and JSP Programming with IBM WebSphere Studio and VisualAge for Java*.

Adding a servlet to WebSphere

Once the servlet has been written, it needs to be registered with WebSphere before use. You can generally use one of the task wizards to help you through the process of adding the new servlet (resource) to your WebSphere setup. The task wizards can be found on the **Tasks** tab in the administrative console.

1. Start WebSphere and log in to the WAS administrative console.
2. Click the **Tasks** tab, select **Configuration** and then select **Configure a Web application.**
3. Click the green **Start Task** button.

4. The wizard appears in the right-hand pane and prompts you for the name of the new Web application (servlet name). Make sure you deselect **Enable File Servlet.**
5. Click **Next**. You will be prompted to select a servlet engine.
6. Our configuration has only one node with one application server (the Default Server), therefore we can select only the **servletEngine** belonging to the default server.

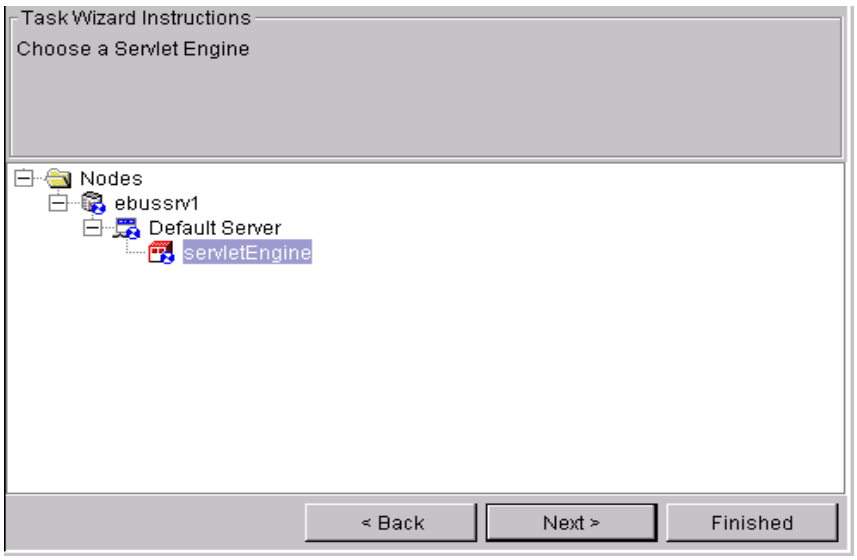


Figure 31. Choose a servlet engine

7. Click **Next**.
8. The next step prompts you for the virtual host and Web application Web path. Select **default_host** for the virtual host. Type in the Web path, for example, /sample . The URL for servlet is:

   `http://yourserver/sample`

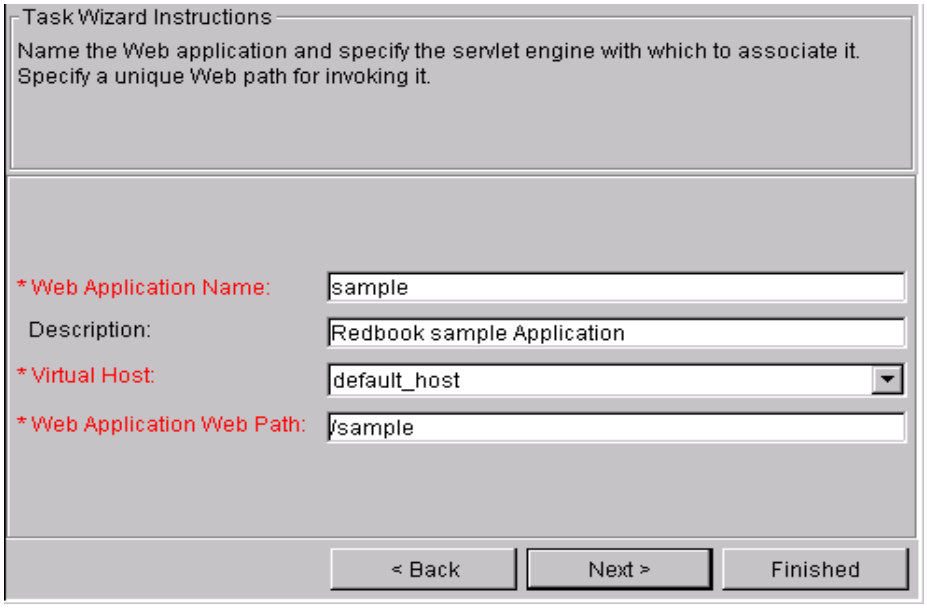


Figure 32. Specify the servlet engine

9. Click **Next.**

10. The next step prompts you for the document root and the classpath (for servlets) of the Web application. The document root is where all your document files used in this Web application are stored; in our case this is d:\WebSphere\AppServer\hosts\default_host\sample**.**

    The directory is not automatically created for you, so you will have to do this as a manual step afterwards. For now, specify the classpath for the Web application, at first your application (in our case sample.jar), and then the VSEConnector.jar, which contains the connector classes.

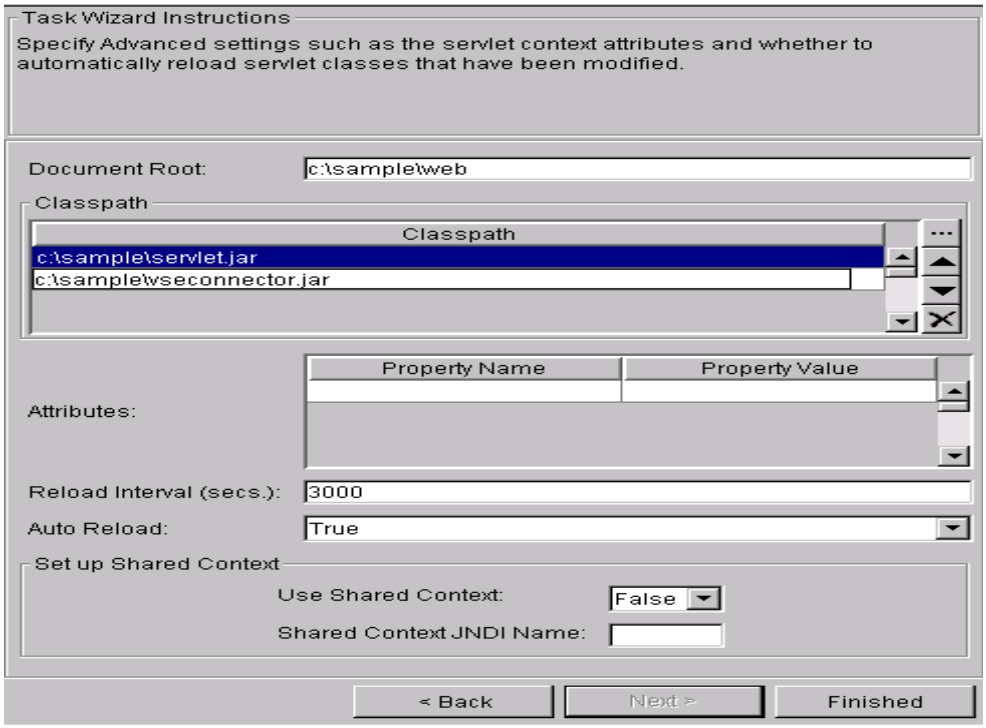


Figure 33. Specify the classpath

11. Click **Finished**. Check the messages pane for a confirmation, as shown in Figure 34:

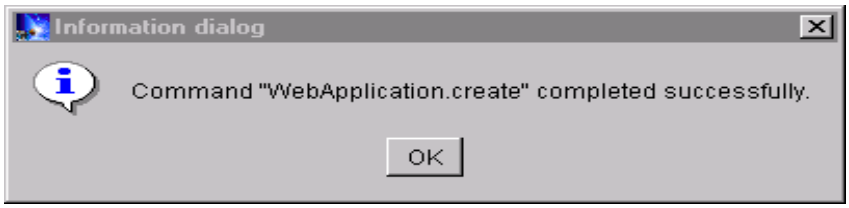


Figure 34. WebSphere completion message

Defining servlets

We set up the Web application in a way that servlets can be invoked by class name. We define the servlet in WebSphere Application Server by selecting the sample Web application in the **Topologies** pane and selecting **Create ->Servlet** from the context menu. We enter VSAMSample as the servlet name and select **sample** as the Web application.

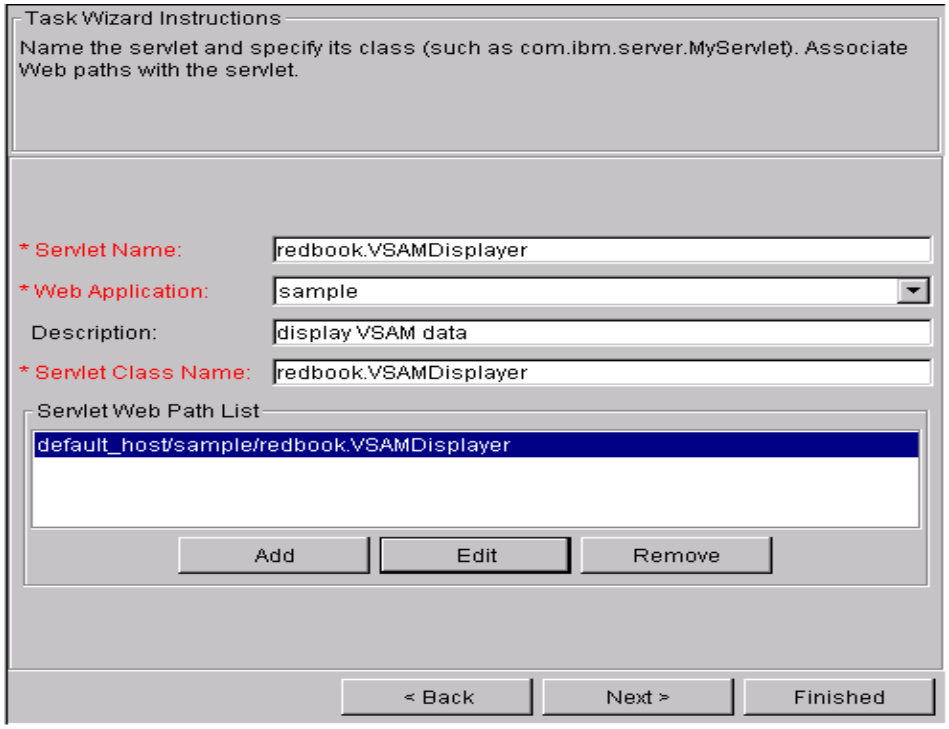


Figure 35. Add a servlet

Click **Add** and enter `/sample/redbook.VSAMDisplayer` as the servlet Web path. Click **Create** to define the servlet.

5.1.4.5 Enterprise JavaBeans (EJBs)

Enterprise JavaBeans are write once, run anywhere components which contain only the business logic of enterprise solutions. They exist in a multi-tier architecture: the client, the EJB server, and persistent storage.

There are two types of EJB:

- Entity beans, which are long-lived beans existing across many client sessions. Entity beans directly represent data, usually one table row in a relational database or, as shown in the VSAM EJB example, one VSAM record that is mapped to a relational structure. An entity bean provides methods to access and change the columns (data fields) of the row (record).
- Session beans, which are tied to the lifetime of a given client session. They encapsulate temporary data and provide business logic to access and update data in a permanent database, usually by accessing an entity bean.

EJBs do not directly interface to the application server's Java Virtual Machine. Instead, they communicate with a container that is provided by the application server. The container manages access to resources from multiple EJBs on that server and provides services for transaction management. The container allows the developer to separate business logic, data access, and presentation logic.

EJBs have some limitations:

- They cannot access the local file system of the Web server platform.
- There is no way to implement callback mechanisms, because EJBs are always single-thread programs. This is a disadvantage, especially for long-running actions like all kinds of requests that retrieve lists of objects. There is no way to stop a running action, nor to begin processing the result list before the complete action has finished.

In a VSE environment, EJBs can be used to:

- Represent data in a relational database (DB2)
- Represent data in non-relational data stores, like DL/I or VSE/VSAM

5.2 The DB2 connector

The DB2 connector enables you to access VSE host data from a Web browser using Java applets. This section discusses how to set up and execute these functions.

5.2.1 Design and architecture

Figure 36 shows a possible 3-tier configuration using the DB2-based connector. The figure shows the connection paths that provide access to VSAM, DL/I, or DB2 data on the host.

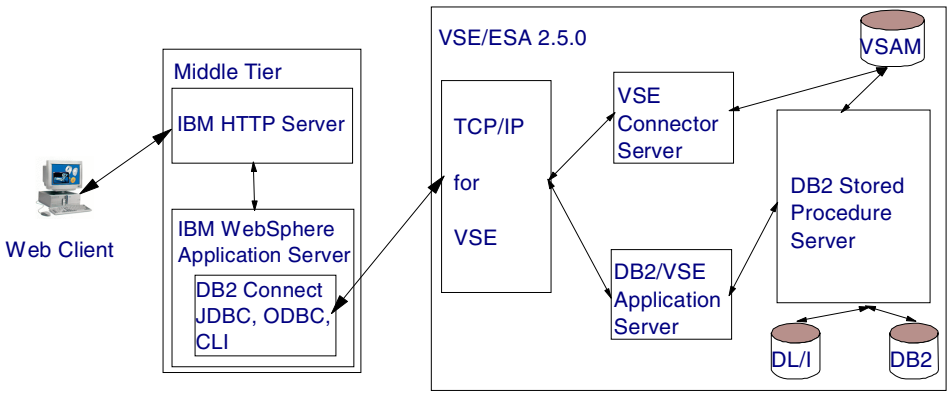


Figure 36. Possible 3-tier configuration using the DB2-based connector

The DB2 connector consists of a client part (DB2 Connect) running on the middle tier and the DB2 server which runs on the host system (DB2 Server for VSE & VM). When the host DB2 server has been installed with the DRDA capability, a TCP/IP connection can be established between the middle tier (DB2 Connect) and the host system (DB2 Server for VSE & VM) through TCP/IP on the VSE/ESA host. This then allows access to the host enterprise data from a client connected via TCP/IP to the middle tier. The client can then, using a standard Web browser, access VSE host data by way of Java applets.

5.2.2 Connector functionality

The DB2 Family of products run on IBM and non-IBM hardware. They also execute under IBM and non-IBM operating systems. If the DB2 database resides on a host system, DB2 Connect is required to interface with this database from a client application. DB2 Connect provides transparent access to data stored in an DB2 Server for VSE & VM host database, and with the use of stored procedures, can access VSAM and DL/I database files. This functionality provides access to relational and non-relational enterprise data resident on the host through standard relational interfaces such as JDBC or ODBC.

Access to DL/I databases through the DB2-based connector and stored procedure server is accomplished mainly through the use of the active CICS-DL/I system. When the new AIBTDLI call is made from a DB2 stored procedure, the request is processed by DLZMPX00. It is recommended that this module be loaded in the SVA. The DLZMPX00 module establishes an XPCC connection to the CICS-DL/I mirror task to process the request. Responses and data are passed back and forth through this XPCC connection, making the data available to the DB2 stored procedure user program.

Host data stored in VSAM files can be accessed from a DB2 stored procedure by way of the new VSAM Call Level Interface (CLI) function. This uses MAPS and VIEWS of the VSAM clusters to “map” record layouts of native VSAM files. For more details, see the online programming reference in VSE Connector Client.

5.2.3 Customizing

Before you can begin customizing the DB2-based connector, the DB2 Server for VSE & VM server must be installed with DRDA code. Documentation for this procedure is included in 4.3.3, “DB2 Server for VSE V7 installation notes” on page 19.

To access DL/I databases from the DB2-based connector through a DB2 stored procedure, you need to ensure the following steps have been completed:

- DL/I for VSE/ESA 1.11 or later has been installed.
- The PTF for APAR PQ39683 has been applied if using DL/I 1.11.
- Your CICS-DL/I system has *all* databases (DBDs) defined in the FCT that you wish to access from DB2 stored procedures.
- The CICS-DL/I system should have all PSBs to be used by the DB2 stored procedures defined in the DL/I nucleus module DLZNUCxx.
- The CICS-DL/I mirror program DLZBPC00 must be authorized to access all required PSBs for any databases you wish to access from DB2 stored procedure programs.
- The CICS-DL/I system should have an active MPS system available.
- The DL/I termination task exit (DLZBSEOT) must be loaded into the SVA.

It will be helpful if at least one CICS system has been configured to use the ISQL interface for DB2 administration tasks. You will need to ensure, however, that at least one CICS system has been configured to access the DB2 Server for VSE and has journaling active.

If you have performed a new install of DB2 Server for VSE & VM, ensure you have at least the supplied DB2 sample database restored and configured. This can be done as part of the install process for DB2. Or you can use the skeleton provided in ICCF library 59 (SKDB2VAR) and follow the documentation for the creation of the sample database in steps 3 to 3.6 in the *VSE/ESA e-business Connectors User's Guide*.

Next you should complete the setup of the DB2 stored procedure server. Documentation for the stored procedure server customization can be found in Chapter 11 of *DB2 Server for VSE & VM Database Administration*. Alternatively, you can use the skeleton JCL members supplied in ICCF library 59 and documented in the *VSE/ESA e-business Connectors User's Guide*.

1. Tailor SKDB2SPS JCL following the CATALOG SPSERV01. A statement in accordance to your site/environment configuration standards. Submit the job to catalog the DB2 stored procedure server skeleton JCL. An example of a tailored SKDB2SPS JCL member is shown here:

```
* $$ JOB JNM=DB2SPSCA,CLASS=0,DISP=D
* $$ LST CLASS=Q,DISP=D
// JOB DB2SPSCA CATALOG STORED PROCEDURE SERVER JOB
// EXEC LIBR,PARM='MSHP'
   ACCESS SUBLIB = PRD2.DB2710
CATALOG SPSERV01.A        EOD=&&          REPLACE=YES
. $$ PUN JNM=SPSERV01,DISP=I,CLASS=R
// JOB SPSERV01    START DB2 STORED PROCEDURE SERVER 01
// OPTION NODUMP,NOSYSDUMP
* // EXEC PROC=ARIS71SL
// ASSGN SYS098,SYSPCH
// LIBDEF *,SEARCH=(PRD2.DB2STP,PRD2.DB2710,PRD2.SCEEBASE,PRD1.BASE)
   ON $RC > 0 GOTO END
// EXEC PGM=ARISPRC,SIZE=(ARISPRC,256K)
/.END
/*
/&
&&
/*
/&
* $$ EOJ
```

2. After this job has completed, you need to define the stored procedure server to DB2. To do this, you can either use ISQL or run a batch job. Following is an example using ISQL:

   Log on to ISQL and enter:

   ```
   create pserver spserv01 autostart yes
   ```

   The response from ISQL is:

   ```
   ARI0500I SQL processing was successful.
   ARI0505I SQLCODE = 0 SQLSTATE = 00000 ROWCOUNT = 0
   ```

   Following is an example using batch JCL:

   ```
   * $$ JOB JNM=PSERVER,CLASS=0,DISP=D
   * $$ LST CLASS=W,DISP=H,DEST=(,xxxxx)
   // JOB PSERVER
   // LIBDEF PROC,SEARCH=(PRD2.DB2710)
   ```

```
// EXEC PROC=ARIS71PL *--DB2 PROD.LIBRARY ID PROC
// EXEC PROC=ARIS71DB *--DB2 DATABASE ID PROC
// EXEC ARIDBS,SIZE=AUTO
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW;
CREATE PSERVER SPSERV01 AUTOSTART YES;
/*
/&
* $$EOJ
```

3. If you used the supplied LIBDEF statement in SKDB2SPS, you will need to catalog all stored procedures into library PRD2.DB2STP.
4. All stored procedures *must* be compiled with the RENT option.
5. Definitions for your stored procedures need to be created. You can use the sample supplied in the SKCRESTP ICCF member to create the required definitions for the supplied samples.

   If you plan on using the supplied samples, you should tailor and run the SKCRESTP job. A return code of 6 is acceptable from this job. However, the output should be checked to ensure that only the DROP commands incurred a non-zero return code. All other SQL commands should complete with a return code of zero.
6. Next you can run the SKCPSTP (Compile/link C/VSE Stored procedures for VSAM data access) and the SKVSSAMP (Define and load sample VSAM cluster) jobs. Check the output from each of these jobs. If you are running SKVSSAMP for the first time, you can expect MAX.RETURN CODE=8 caused by the IDCAMS DELETE for the sample VSAM file failing. All other steps should complete with MAX.RETURN CODE=0.

   If you wish to use the delete, update and insert VSAM samples, you will need to execute SKCPSTP for each of the required actions. Simply change the PHASE JCL statement and the INFILE parameter on the C/VSE compile to each of the required samples. If you use all four supplied samples, you will need to run SKCPSTP four times. The samples' names are: VSAMSEL, VSAMDEL, VSAMINS, VSAMUPD.
7. If you want to use the DB2-based connector support samples to access DL/I databases, you may wish to install the sample stored procedures and definitions. The sample stored procedure definitions can be created by running job SKDLISTP supplied in ICCF library 59. A MAX.RETURN CODE =6 is acceptable for this job, though the output should be checked to ensure that only a non-zero SQLCODE and SQLSTAT were received for DROP commands.

   Sample stored procedures, written in COBOL using DL/I, are supplied in member SKDLICMP in ICCF library 59. If you wish to use any or all of these samples, this JCL should be tailored and submitted. If you do not have DL/I databases available for accessing but you wish to use this support via the DB2-based connector, you can install the sample DL/I database to use for demonstration purposes. The job(s) for creating a sample DL/I database are in member SKDLISMP in ICCF library 59. Tailor and submit this JCL to your VSE host machine. Following is a console log from the creation process:

```
BG 0001 1Q47I   BG STJDBGEN 04217 FROM (RES1) ,TIME=13:26:17
 BG 0000 // JOB STJDBDGN GENERATE DBDS FOR SAMPLE DATA BASE
         DATE 08/16/2000, CLOCK 13/26/17
 BG 0000 EOJ STJDBDGN  MAX.RETURN CODE=0000
         DATE 08/16/2000, CLOCK 13/26/34, DURATION   00/00/17
 BG 0000 // JOB STJPSBGN GENERATE PSBS FOR SAMPLE DATA BASE
```

```
        DATE 08/16/2000, CLOCK 13/26/34
 BG 0000 EOJ STJPSBGN  MAX.RETURN CODE=0000
        DATE 08/16/2000, CLOCK 13/26/40, DURATION   00/00/05
 BG 0000 // JOB STJACBGN GENERATE ACBS FOR SAMPLE DATA BASE
        DATE 08/16/2000, CLOCK 13/26/40
 BG 0000 EOJ STJACBGN  MAX.RETURN CODE=0000
        DATE 08/16/2000, CLOCK 13/26/42, DURATION   00/00/02
 BG 0001 1Q47I   BG STJDBLOD 04218 FROM (RES1) , TIME=13:26:42
 BG 0000 // JOB STJPREOR PRE-REORGANIZATION
        DATE 08/16/2000, CLOCK 13/26/42
 BG 0000 DLZ968I    BEGIN EXECUTION DLZURPR0   08/16/2000   13:26:43

 BG 0000 DLZ961I    0001  DBIL=STDIDBP

 BG 0000 DLZ961I    0002  OPTIONS=(NOPUNCH,STAT,SUMM)

 BG 0000 DLZ966I    NORMAL PROGRAM TERMINATION
 BG 0000 EOJ STJPREOR  MAX.RETURN CODE=0000
        DATE 08/16/2000, CLOCK 13/26/43, DURATION   00/00/01
 BG 0000 // JOB STJDFINV DEFINE CLUSTER FOR SAMPLE DATA BASE
        DATE 08/16/2000, CLOCK 13/26/43
 BG 0000 EOJ STJDFINV  MAX.RETURN CODE=0008
        DATE 08/16/2000, CLOCK 13/26/45, DURATION   00/00/01
 BG 0000 // JOB STJLDCST LOAD SAMPLE DATA BASE
        DATE 08/16/2000, CLOCK 13/26/45
 BG 0000 1T20I  SYS010 HAS BEEN ASSIGNED TO X'FEC' (TEMP)
 BG 0000 1T20I  SYS011 HAS BEEN ASSIGNED TO X'FEE' (TEMP)
 BG 0000 EOJ STJLDCST  MAX.RETURN CODE=0000
        DATE 08/16/2000, CLOCK 13/26/47, DURATION   00/00/02
 BG 0000 // JOB STJPRRES PREFIX RESOLUTION
        DATE 08/16/2000, CLOCK 13/26/47
 BG 0000 DLZ968I    BEGIN EXECUTION DLZURG10   08/16/2000   13:26:48

 BG 0000 ILU321I J1 STJPRRES SORT  COMPLETE, INSERT 12, DELETE 12, IN 0,
OUT
 BG 0000 ILU321I J1 STJPRRES SORT  COMPLETE, INSERT 9, DELETE 9, IN 0,
OUT 0
 BG 0000 DLZ966I    NORMAL PROGRAM TERMINATION

 BG 0000 EOJ STJPRRES  MAX.RETURN CODE=0000
        DATE 08/16/2000, CLOCK 13/26/51, DURATION   00/00/03
 BG 0000 // JOB STJPRUPD PREFIX UPDATE
        DATE 08/16/2000, CLOCK 13/26/51
 BG 0000 DLZ968I    BEGIN EXECUTION DLZURGP0   08/16/2000   13:26:52

 BG 0000 DLZ966I    NORMAL PROGRAM TERMINATION   00000009

 BG 0000 EOJ STJPRUPD  MAX.RETURN CODE=0000
        DATE 08/16/2000, CLOCK 13/26/52, DURATION   00/00/00
 BG 0001 1Q34I   BG WAITING FOR WORK
```

8. To access DL/I data via the DB2-based connector, you need to also customize a CICS-DL/I system. The following customization details assume you are going to use a CICS TS system for DL/I access control.

   a. Customize your CICS TS system according to Part 6 of the document *DL/I DOS/VS Resource Definition and Utilities* and the CICS-DL/I tables

requirements. Then read the section “Migrating to DL/I VSE 1.11 and CICS TS for VSE 1.1” in the *DL/I 1.11 Release Guide*.

b. Define your DL/I databases for the supplied sample DL/I database (STDIDBP), to the CICS TS FCT. Assemble the modified FCT and verify a successful assembly.

c. Provide labels to the databases you wish to access either in the CICS startup JCL or in the VSE standard label area.

d. Create or modify the DLZNUCxx (using DLZACT) by including all DL/I online programs and PSBs you wish to use. Ensure the CICS-DL/I mirror program is authorized to use PSB STBICLG, which is used to access the sample database.

e. Account for an increase in the number of mirror tasks running on the CICS TS system. You may want to increase the value specified for MAXTASK and CMAXTASK in the DLZACT generation.

f. Activate an MPS system.

9. Now ensure that DB2 Server for VSE & VM is executing in a VSE partition with DRDA support and stored procedure support. Sample JCL for this is supplied in ICCF library 59 member SKDB2STR. You need to have TCP/IP started prior to executing the DB2 server job.

10. Finally, if you have not already done so, install DB2 Connect on your middle-tier platform and configure to access DB2 data stored on the host. Define the host DB2 database to DB2 Connect. Installation and customization of the DB2 Connect product is described in 4.4.3.3, “DB2 Universal Database Version 7 for Windows NT” on page 24. Documentation for this step is also available in the document *DB2 Connect User’s Guide* or from the Web at:

    `http://www.ibm.com/db2`

Once customization is complete, you are ready to access host data from a client connection through the middle-tier platform.

5.2.4 Using the connector to access data

Once you have installed and customized the DB2-based connector, you should be able to access your VSE data.

5.2.4.1 The DB2-based connector

In the following section we discuss the installation requirements and implementation of how to access VSAM data files using the supplied DB2-based stored procedures and sample Java applets.The sample uses the 3-tier environment, as use of the DB2-based connector is not possible in a 2-tier environment.

To use the supplied Java applet samples, you will need to have performed the steps previously described in 4.3, “Installation overview” on page 14 and 4.4, “Customizing the prerequisites” on page 22. You will also need to have installed the following on your middle-tier system:

- An HTTP server (for example, IBM HTTP Server)
- A Web application server (for example, WebSphere Application Server)
- DB2 Connect V7
- ODBC, JDBC or CLI

A TCP/IP connection needs to be established between DB2 Connect on your middle tier and the DB2 Server for VSE & VM on your host system. The middle tier will also require the Java applet code from the samples supplied in member IESINCON.W in library PRD1.BASE. These can be installed in any directory on the middle-tier system. Ensure you also have the VSEConnector.jar file installed in the same directory, as the client will require access to the VSE JavaBeans class.

For the following example, you should have the sample applet code and DB2Index.html file in your httpd/htdocs directory, published on your Web server.

You can now start the VSAM applet from a Web browser by using the URL prefixed by your Web server's address, followed by the DB2INDEX.HTML file. The URL used for this example was:

```
http://9.164.167.81/db2vsam/db2index.html
```

The applet sample code was installed on the Web server in directory httpd/htdocs/db2vsam.

When connected, you will be presented with the screen shown in Figure 37 in your Web browser.

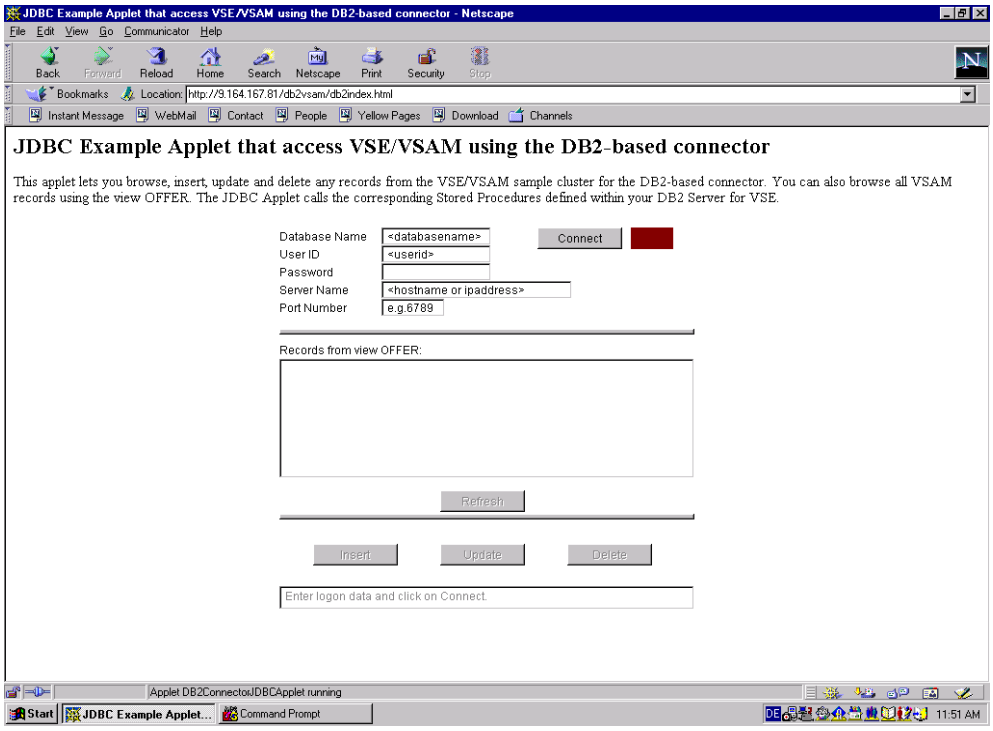


Figure 37. Applet used to access VSAM data through DB2-based connector

In the Database Name field, enter the name of the DB2 Server for VSE database as defined to DB2 Connect on your middle-tier system. You can use either the DB2 database name defined or an alias, if you have created one. Enter your user ID and password for the DB2 database on the host VSE system in their respective fields.

In the Server Name field, enter either the IP address or domain name of the server running DB2 Connect. Supply the port number DB2 Connect has been configured to use in the Port Number field. Click the **Connect** button to process the connection request to DB2 Connect and on to DB2 Server for VSE on the host. The DB2 for VSE stored procedure server will start in the designated host VSE partition if no previous executions of stored procedures have been actioned.

If you have loaded the sample VSE Connector VSAM file (VSE.CONN.SAMPLE) and defined its associated map (USEDCARS) and view (OFFER) using the supplied ICCF member SKVSSAMP in library 59, the following screen will be displayed in your Web browser:

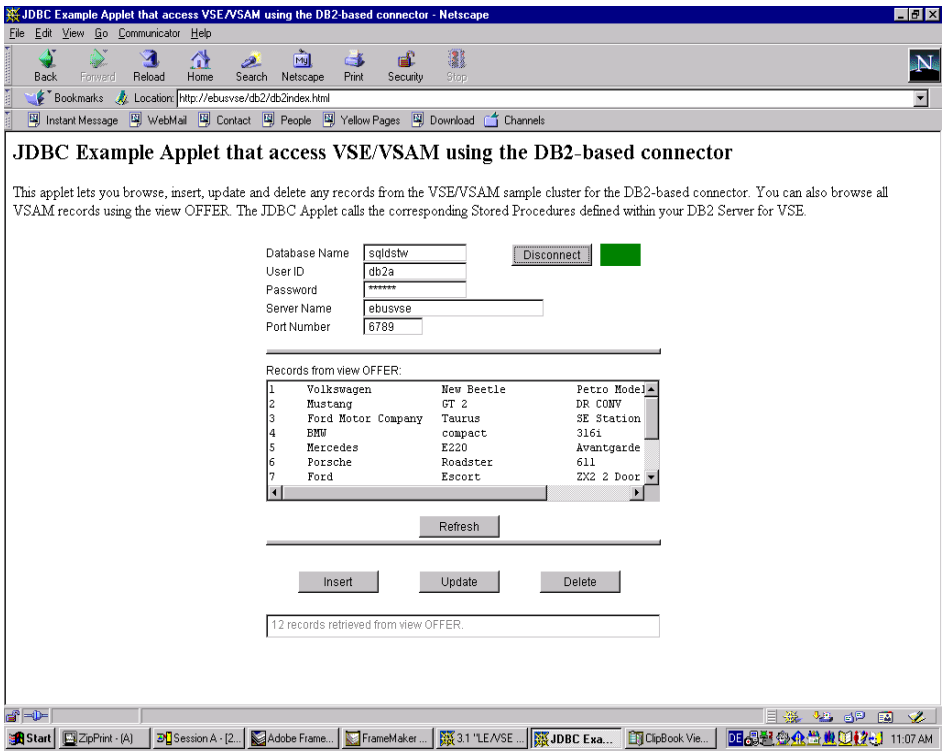


Figure 38. Using DB2-based connector and stored procedure server for record retrieval

If all three supplied VSAM access stored procedure programs have been compiled, linked and cataloged into the DB2 stored procedure library, you will be able to use each of the INSERT, UPDATE and DELETE buttons within the applet to modify the VSAM file.

To do this first decide, from the displayed list of records, which record you want to modify or delete. If you decide to insert a new record, first find a free row number to use. Next, click the button for the desired request (for this example we used **Update**). After clicking the button, a pop-up window will appear. Following is a screen shot of the Update function pop-up window:

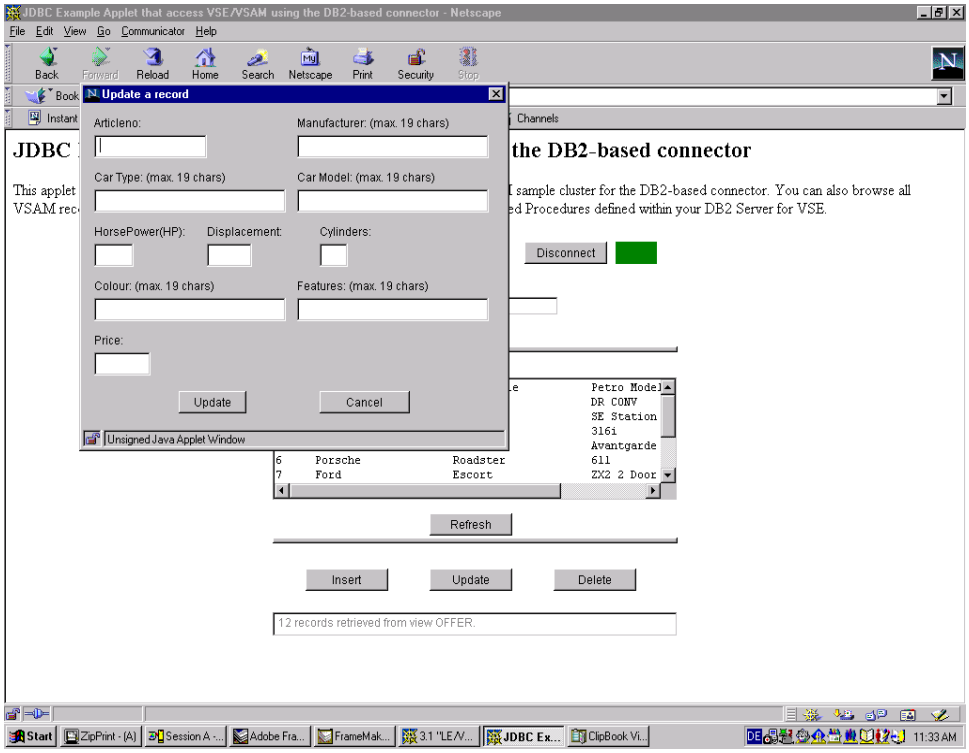


Figure 39. Update button pop-up window

We requested that the record on the seventh row be changed from Ford to Ford Motor Company by entering the number 7 into the Articleno field. In the Manufacturer field we entered Ford Motor Company. Then we clicked **Update** to process the request. To view the updated record, we clicked **Refresh** to instruct the applet to retrieve all the records from the VSAM file again.

The applet display window appeared:

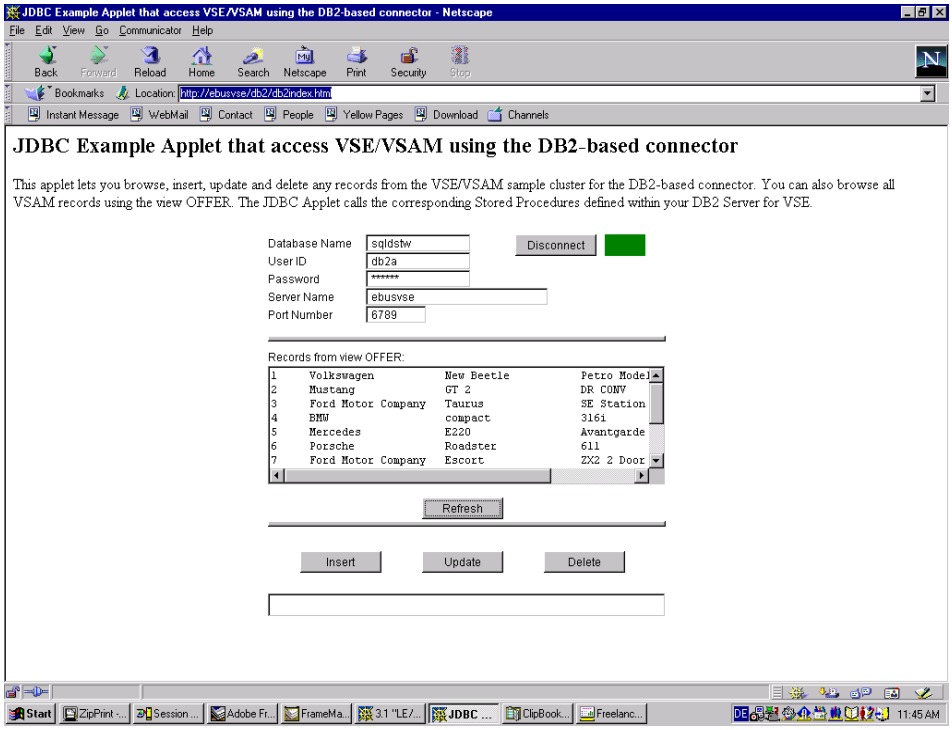


Figure 40. Updated VSAM record using the Java-based applet

Note that the record on the seventh row now has a manufacturer of Ford Motor Company. A similar process can be followed to exercise the Insert and Delete functions of the applet.

5.3 CICS connectivity in a 3-tier environment

In a 3-tier configuration, CICS connectivity is provided by the CICS Transaction Gateway. This allows CICS applications to be accessed from Java Web-based front-ends. CICS Transaction Gateway provides many capabilities, including:

- Enabling a Java gateway application, which is usually stored on the middle tier, to communicate with CICS applications running under CICS TS for VSE/ESA through the ECI or EPI
- Enabling the CICS Java class library to be used for communication between a Java gateway application and a Java application
- Allowing you to use a Web browser as an emulator for a 3270 CICS application on the CICS TS for VSE/ESA server, via a terminal servlet
- Allowing you to use a set of Java EPI beans to create Java front-ends for existing CICS 3270 applications without any programming effort

For more details, refer to an article in the *VSE/ESA Software Newsletter, Issue 19* which you can find at:

`http://www.s390.ibm.com/ftp/vse/docs/vsenew19/vsedtsw.pdf`

You can also refer to the CICS Web site:

```
http://www.ibm.com/software/ts/cics
```

The IBM Redbook *e-business Solutions for VSE/ESA* also contains detailed information on implementing and using these features.

5.4 CICS connectivity in a 2-tier environment

In a 2-tier configuration, the following enhancements to CICS TS for VSE/ESA are provided:

- CICS Web Support - this new set of services enables direct access to CICS applications and transactions from Web browsers.
- Extensions to the CICS application programming interface - these allow you to:
  - Build formatted areas (for example, HTML pages) in a CICS program and send them back to the Web browser that requested them.
  - Read and process HTML headers and data sent from a Web browser-initiated CICS transaction.
- CICS 3270 Bridge - this new function provides the ability to run existing 3270-based transactions without a 3270 terminal.

For more details, refer to an article in the *VSE/ESA Software Newsletter, Issue 19* which you can find at:

```
http://www.s390.ibm.com/ftp/vse/docs/vsenew19/vsedtsw.pdf
```

You can also refer to the CICS Web site:

```
http://www.ibm.com/software/ts/cics
```

The IBM Redbook *CICS TS New Functions* also contains more detailed information on implementing and using these features.

5.4.1 CICS Web Support

You can invoke a CICS application from a Web browser by entering a URL in the form:

```
http://www.your.url:portnumber/converter/alias/program
```

where:

- portnumber is a port number on which CICS listens for requests (configured by you)
- converter is the name of a user-written program used to decode and encode data from and to the Web browser
- alias is the transaction ID of the alias transaction
- program is the name of your CICS application program to be invoked

Provided that your CICS and TCP/IP environments have been set up correctly, the CICS Web Support will process an incoming request from a Web browser, translate it, and then pass the required COMMAREA to your CICS application program.

You can choose either to pass the output from this CICS program back to the converter (which would encode the data before sending it back to the requesting Web browser), or you can use the new CICS Web API to build the HTTP headers and data directly in your CICS program.

5.4.2 Extensions to the CICS application programming interface

There are new commands in two areas:

- `EXEC CICS DOCUMENT`
- `EXEC CICS WEB`

The `EXEC CICS DOCUMENT` commands allow you to create and then manipulate formatted data areas in your CICS program. For the purposes of this interface, these data areas would be HTML pages, although you could construct many other types of formatted output. The data areas are known as *documents*, and can contain text and binary components. They are defined as DOCTEMPLATES to your CICS system definition file.

The `EXEC CICS WEB` commands allow you to handle HTTP requests and responses from your program and to take action based on this analysis in your program.

5.4.3 The 3270 Bridge

This new function allows you to invoke 3270-based applications from a non-3270 source. Typically this would be from a Web browser, but it could also be from MQSeries, for instance. The 3270 Bridge function is especially suitable where you need to get a CICS application “Web-enabled” quickly. No changes are necessary to the application program; if your application uses BMS maps, then these need to be reassembled using new, supplied macros. The 3270 Bridge also supports non-BMS 3270 data streams.

Other macros are also supplied that enable you to customize the screen layout when output is being transmitted to a Web browser, rather than to a “green screen”.

The 3270 Bridge is invoked using a URL in the form:

```
http://www.your.url:portnumber/cics/cwba/dfhwbtta/transid
```

where:

- portnumber is a port number on which CICS listens for requests (configured by you)
- cics is a fixed value, indicating that no converter program should be used
- cwba is a fixed value, representing the name of the alias transaction ID
- dfhwbtta is a fixed value, representing the name of the IBM supplied 3270 Bridge program
- transid is the ID of the transaction that you wish to execute

The 3270 Bridge program intercepts all 3270 terminal I/O requests and emulates a 3270 terminal transparently to the 3270 transaction being executed.

5.5 MQSeries-based connector

You can use the MQSeries Client for Java to communicate from a Web browser to legacy applications running on VSE/ESA.

5.5.1 Design and architecture

The MQSeries Client for Java accesses the MQSeries queue manager by establishing a client connection through the Internet. The client can be installed either on a local hard disk, or on a Web server.

5.5.2 Connectors functionality

The MQSeries Java client enables Web browsers and Java applets and applications to issue calls and queries to MQSeries giving access to VSE/ESA-based applications over the Internet without the need for any other MQSeries code on the client machine.

Installation on a Web server has the advantage that MQSeries client applications can be downloaded and run on machines which do not have the MQSeries Client for Java installed locally. When the applet is downloaded from a Web server, it can only communicate with an MQSeries queue manager running on the same machine as the Web server.

Note: If you intend to use the MQSeries Java client to connect to an MQSeries server running on VSE/ESA 2.5, you should note carefully the following restriction:

MQSeries Version 2.1.0 does not support Java clients, because Java classes attempt to open a queue manager as an object on connection. However, under MQSeries Version 2.1.1 (which is available under VSE/ESA 2.5), you can connect to a Java client, because an MQSeries queue manager on VSE/ESA can be opened as an object.

5.6 Sample connector application

This section provides you with an overview of VSE/ESA Navigator and the Java application JDataMig. The Java application JDataMig is our self-written product.

5.6.1 VSE/ESA Navigator

The VSE/ESA Navigator is a Java application that is implemented on the basis of the VSE JavaBeans class library. This class library is part of the VSE/ESA e-business connectors.

Like the VSE Connector Client, the VSE/ESA Navigator is provided through a single Java installation class file. This file can be downloaded from the VSE Home page at:

```
http://www.s390.ibm.com/products/vse/support/vseconn/vsenavi.htm
```

You have to download the file install.class in binary to any Java-enabled workstation. To install the Navigator client, enter the following from a command prompt:

```
java install
```

Before running the VSE/ESA Navigator, make sure that the current directory is contained in your classpath environment variable.

Since the VSE/ESA Navigator is implemented on the basis of the VSE JavaBeans class library, the VSEConnector.jar file must be available on your workstation and be contained in your classpath.

5.6.1.1 VSE/ESA Navigator startup

The Navigator root directory contains a number of files that can be used to bring up the application on different platforms.

- RUN.BAT is intended for Windows.
- RUN.CMD is intended for OS/2.
- JRE.BAT is intended for OS/2 or Windows with JRE.
- RUN.SH is intended for UNIX/Linux.

There are two icons provided, one for Windows and one for OS/2.

5.6.1.2 Completing the installation

After you finish the installation process, desktop objects (icons) should have been created on your computer. For some platforms (Windows and OS/2), we always create two program objects to start up the application:

- java.exe should be used for a “normal” Java installation.
- jre.exe should be used for a Java Runtime Environment (JRE) installation.

The manner in which the VSE/ESA Navigator main class file is called depends on whether we are using a java.exe or a jre.exe. For instance, JRE requires the classpath to be specified as an argument when calling JRE. You will need to experiment to find out which object works best on your system.

5.6.1.3 The VSE/ESA Navigator GUI

The Navigator’s graphical user interface (GUI) behaves very much like file managers that you know from other platforms, such as Windows Explorer. The difference is that it includes host-based VSE file systems and provides host-specific functions like generate and submit VSE jobs, list VTOC, retrace products, display host CPU activity, and many more.

Each icon displays a pop-up menu when being right-clicked. Whenever a function needs host access, you are prompted to log on to the host if you are not already logged on. When you use the functions under pull-down menu **Functions**, you will be prompted to specify the host on which you want to execute this function. The same functions will show up in the host's pop-up menu.

When you use this pop-up menu, there will be no prompt for the host. The **Configuration** pull-down menu offers several functions to configure the client. For example, the **Look and feel** menu choice offers several ways to dynamically change the appearance of the main window and dialog boxes.

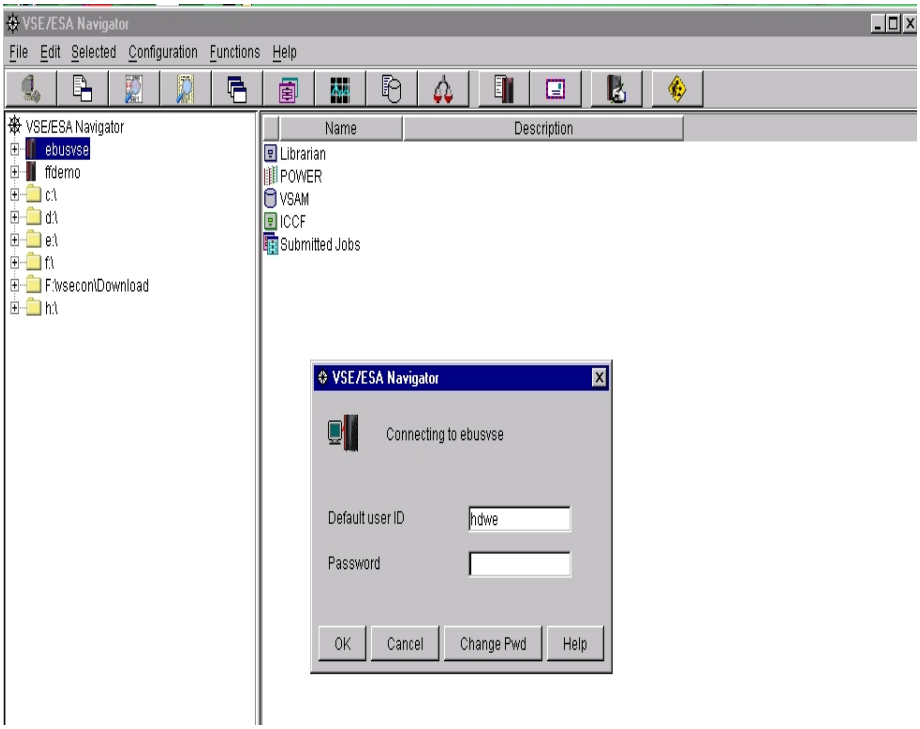


Figure 41. VSE/ESA Navigator log on

All client-related settings are stored in two configuration files in the Navigator's root directory:

- VSENavigator.cfg
- Filetypes.cfg

These files are plain ASCII text files and can be edited by the user.

The VSE/ESA Navigator menu bar and tool bar

This part allows you to select the functions of the VSE/ESA Navigator. You can select the desired function either by using the menu bar, or by pushing one of the tool bar buttons. The menu bar consists of several menu items:

- File
- Edit
- Selected
- Configuration
- Functions
- Help

The **Selected** menu item allows you to select object specific functions for the currently selected object. The tool bar consists of several tool bar buttons. By pressing the toolbar button, you select the function. Some of these functions can also be found in the **Functions** menu.

The VSE/ESA Navigator tree

This tree, shown in Figure 42, is located at the left side of the main window. It allows you to navigate through the VSE/ESA hosts and their file system hierarchy. You can define the hosts by using the **Configure Hosts Dialog** function. This can be found in the **Configuration** menu.

You can also define local directories which are displayed below the hosts. This is useful for downloading or uploading files from or to these directories. You can define the local directories by using the **Configure Paths Dialog** on the **Workdirectories** tab. For each host there are the following file systems:

- Librarian
- POWER
- VSAM
- ICCF
- Submitted Jobs

The Submitted Jobs component is a logical component and is not physically located on the host. With this component you can manage the execution of jobs.

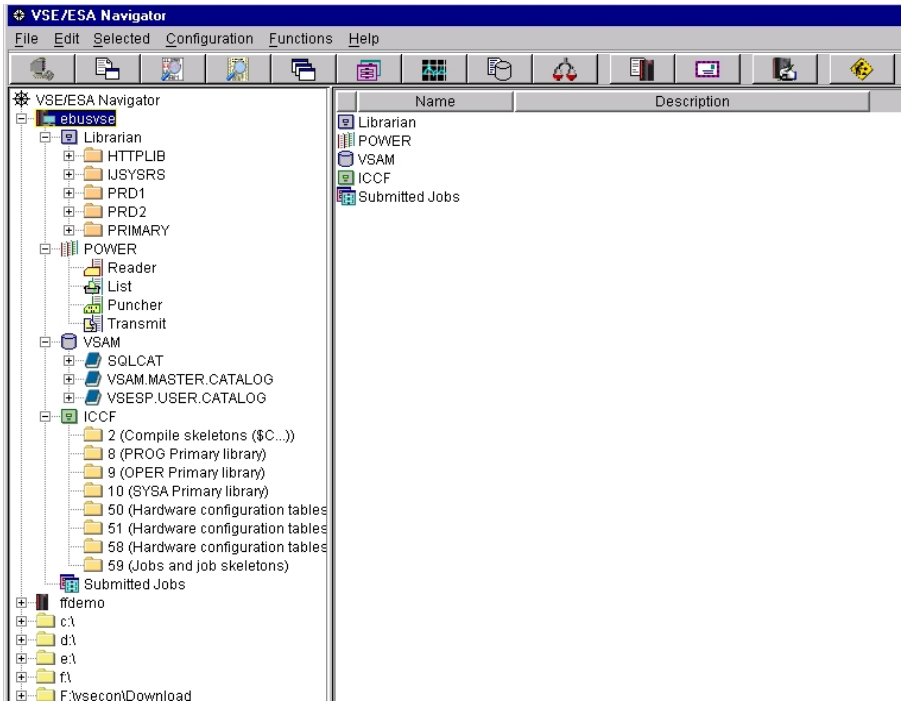


Figure 42. VSE/ESA Navigator tree

5.6.1.4 Some selected dialogs using VSE/ESA Navigator

The following two figures contrast the difference in the VSE/ESA system activity screens using the VSE/ESA Navigator tool and the traditional 3270 screen using the Interactive User Interface.

Figure 43 on page 66 shows the function Display system activity, monitoring static partitions with VSE/ESA Navigator.

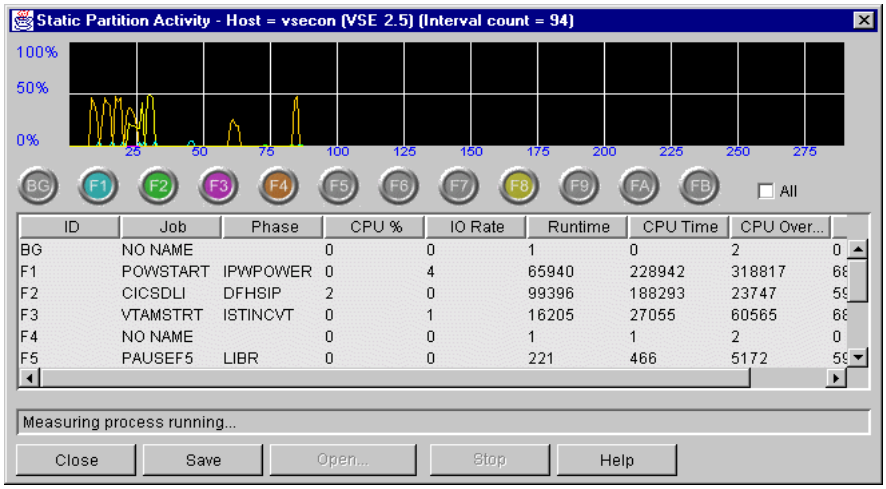


Figure 43. VSE /ESA Navigator system activity

Figure 44 shows the same function Display system activity, monitoring static partitions using the 3270 layout.

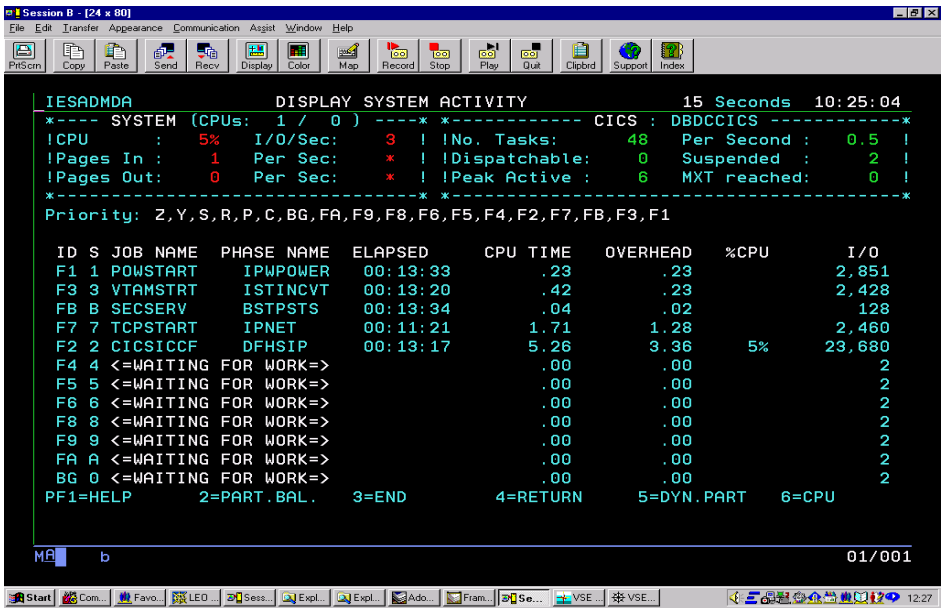


Figure 44. VSE/ESA system activity

Figure 45 on page 67 shows the function Monitor free and used VSAM catalog space with VSE/ESA Navigator.

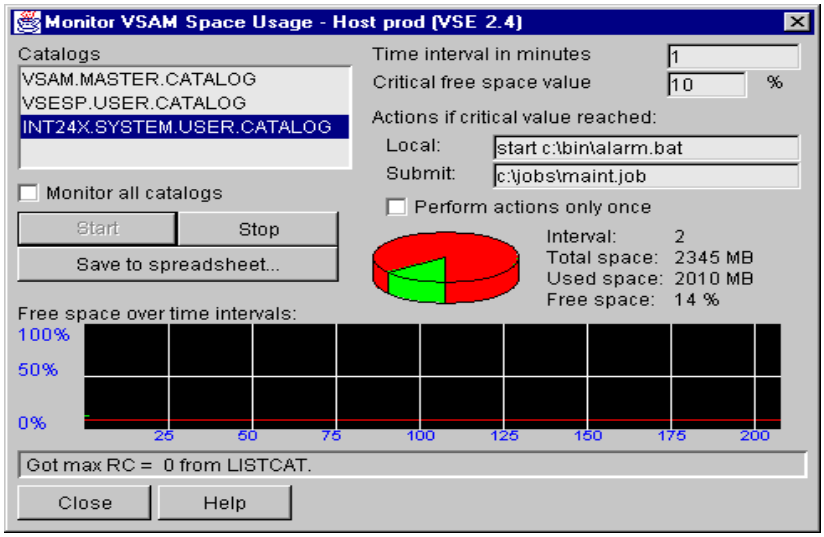


Figure 45. VSE/ESA Navigator VSAM space monitoring

5.6.2 JDataMig (Java data migration application)

At this point, we have created a Java application which can access VSAM data and replicate the VSAM data in a local or a remote DB2 database. The JDataMig application is a command line version which handles many of the steps needed for the data migration:

- Logging on to the VSE system
- Selecting the VSAM catalog, VSAM cluster, and VSAM mapping file
- Selecting the database network driver
- Logging on to the DB2 instance
- Selecting the DB2 database
- Creating and/or dropping the DB2 table
- Cleansing the data, when necessary
- Copying the data from VSE/VSAM to DB2

Figure 46 on page 68 shows the configuration file called sample.cfg that can be used for the JDataMig application. To execute it, issue the command:

```
x:\JDataMig>run -f sample.cfg batch
```

```
# vse-config
vse.host=9.164.155.55
vse.user=mla1
vse.pwd=hallo

# db/2-config
db2.driver=2
db2.database=ebusnt
db2.user=pool36
db2.pwd=ebusvse
db2.host=ebussrv1
db2.port=6789

# migration-config
vsam.catalog=VSESP.USER.CATALOG
vsam.cluster=JAVAFILE.SAMPLE.FILEA
vsam.map=FILEAMAP

# migration-options
# is it allowed to kill the table/file, if it exists
options.droptable=true
# only valid for db/2: if not dropping the table, is it allowed to clear it?
options.cleartable=true
```

Figure 46. The configuration file sample.cfg for JDataMig

The JDataMig application is based on the VSE Java-based connector and can be used to access non-relational VSAM data sources and relational data such as DB2. Using this tool, you can work with your non-relational data while others access the same data as a relational database. The source of the JDataMig application is listed in Appendix B.1, “Java source: JDataMig” on page 91.

Chapter 6. Operation

In this chapter we provide a brief overview of typical operation and administration commands, as well as hints and tips on troubleshooting. This should help you to get the VSE e-business connectors environment up and running more easily.

6.1 Administration

The information in this section is based upon our experience with VSE e-business connectors.

6.1.1 TCP/IP commands

START The START command restarts the TCP/IP engine after it has been stopped with the STOP command.

STOP The STOP command causes the TCP/IP engine to temporarily stop dispatching pseudo tasks, or to suspend attempts to activate a specific communication link.

QUERY The publication *TCP/IP for VSE Commands, Release 1.4* contains details and descriptions of all QUERY parameters. Figure 47 shows a sample of the QUERY HTTPDS command to see the currently used setup for the HTTP daemon; for instance, the root library where the HTML and Java class files reside.

```
F7-0100 IPN300I Enter TCP/IP Command
100 QUERY HTTPDS
F7 0098 IPN508I (( TCP/IP HTTP Daemons ))
F7 0098 IPN509I  ID: HTTPD_ID1 Port:   80  Driver: HTTPD
F7 0098 IPN510I Confine: OFF Root: HTTPLIB.ROOT.
```

Figure 47. QUERY HTTPDS command

To get the current IP address of VSE/ESA TCP/IP on the host site, you can use the TCP/IP command QUERY SET on the VSE/ESA console; see Figure 48.

```
msg f7
AR 0015 1I40I  READY
F7 0100 IPN300I Enter TCP/IP Command
F7-0100
100 QUERY SET
F7 0098 IPN451I (( TCP/IP Set Information ))
F7 0098 IPN452I  System ID: 00
F7 0098 IPN453I  IP Address: 9.164.155.55  Submask: 255.255.224.0
F7 0098 IPN454I  All Bound Time: 30000  Link Retry Time: 18000
```

Figure 48. QUERY SET command and response

IPCONFIG The IP address on the client site can be displayed with the TCP/IP command `IPCONFIG` from a command prompt window, as shown in Figure 49.

```
C:\>ipconfig
Windows NT IP Configuration
Token Ring adapter AT25LE1:
IP Address. . . . . : 9.164.188.226
Subnet Mask . . . . . : 255.255.224.0
Default Gateway . . . . . : 9.164.181.1
```

Figure 49. IPCONFIG command and response

PING The `PING` command (PING IP address of partner's TCP/IP) permits testing of a communication link without involving clients and servers on each end. You may issue a `PING` command from a VSE host to another server, or vice versa.The responses will show whether the remote TCP/IP can be accessed or not; see Figure 50 and Figure 51.

```
100 ping 9.164.188.226
F7 0098 TCP910I  Client manager connection Established.
F7 0098 TCP915I  PING
F7 0098 TCP910I  PING Ready:
F7 0098 TCP915I  SET HOST= 9.164.188.226
100 ping 9.164.188.226
F7 0098 TCP910I  Client manager connection Established.
F7 0098 TCP915I  PING
F7 0098 TCP910I  PING 1 was successful, milliseconds: 00011.
F7 0098 TCP910I  PING 2 was successful, milliseconds: 00010.
F7 0098 TCP910I  PING 3 was successful, milliseconds: 00010.
F7 0098 TCP910I  PING 4 was successful, milliseconds: 00010.
F7 0098 TCP910I  PING 5 was successful, milliseconds: 00010.
F7 0098 TCP910I  PING Ready:
F7 0098 TCP915I  QUIT
```

Figure 50. PING command and response on VSE/ESA console

```
C:\>ping 9.164.155.55
Pinging 9.164.155.55 with 32 bytes of data:
Reply from 9.164.155.55: bytes=32 time=10ms TTL=253
Reply from 9.164.155.55: bytes=32 time=10ms TTL=253
Reply from 9.164.155.55: bytes=32 time=10ms TTL=253
Reply from 9.164.155.55: bytes=32 time=10ms TTL=253
```

Figure 51. PING command and response on server site

6.1.2 VSE Connector Server commands

Ensure that the VSE Connector Server job STARTVCS is up and running. Reply `HELP` to the MSG response to bring up the help menu; refer to Figure 52 on page 71.

```
msg r1
AR 0015 1I15I READY
R1-0045 IESC1046I ENTER VSE CONNECTOR SERVER COMMANDS
45 HELP
R1 0045 IESC1043I HELP COMMAND
R1 0045 HELP|?                                      PRINTS THIS MESSAGE
R1 0045 STATUS {ALL|CONFIG|CLIENTS|PLUGINS|VSAM} PRINTS STATUS INFORMATION
R1 0045 SENDMSG <USER(S)> <MESSAGE TEXT>          SENDS A MESSAGE TO A USER
R1 0045 SHUTDOWN {NOPROMPT}                       SHUTS DOWN THE SERVER
R1 0045 SETTRACE <TRACEFILE> <TRACELEVEL>         SET TRACING ON/OFF
R1 0045 STOP CLIENT <CLIENT-ID|ALL>               STOPS THE SPECIFIED CLIENT
R1 0045 CLOSE VSAM <SLOT-ID|ALL>                  CLOSES A VSAM CLUSTER
```

Figure 52. VSE Connector Server HELP menu

To enter a command that the VSE Connector Server should process, use this command syntax:

```
msg <jobname>,data=<command>
```

where:

- `<jobname>` is the actual name of the VSE Connector Server startup job.
- `<command>` is one of the command strings shown in Figure 52.

For example:

```
msg startvcs,data=shutdown
```

6.1.3 DB2 connector commands

Following is a description of operator console commands that can be helpful when using the DB2-based connector from a VSE host DB2 application server.

SHOW CONNECT Used to display the status of all or selected users connected to the application server by remote DRDA support.

SHOW ACTIVE Displays only the active users connected to the VSE host DB2 application server.

SHOW PSERVER * Displays the status of all DB2 procedure servers controlled by the application server. This is useful for determining the status of defined stored procedure servers. Figure 53 shows a sample console log.

```
F4-0004 ARI0062A SQLDS :
              Enter a DB2 Server for VSE operator command.
4 show pserver *
F4 0004 Status of stored procedure server(s)    2000-08-29  14:58:42
F4 0004 PSERVER  GROUP            PROCNAME          AUTHID   STATUS   IMPL
F4 0004 SPSERV01                                             STARTED.   Y
F4 0004 ARI0065I Operator command processing is complete.
```

Figure 53. Console log output from SHOW PSERVER command

The status of STARTED indicates that this stored procedure server is currently running in a VSE partition. A status of STARTING is normal for a stored procedure server that has not yet been activated.

6.1.4 IBM WebSphere Application Server

To start the server, run the administrative console, which is how you manage WebSphere Application Server 3.0. The basic administrative console is split into three tabs:

1. Tasks
2. Types
3. Topology

If you start the Administrator's Console, you should see a screen similar to Figure 54:

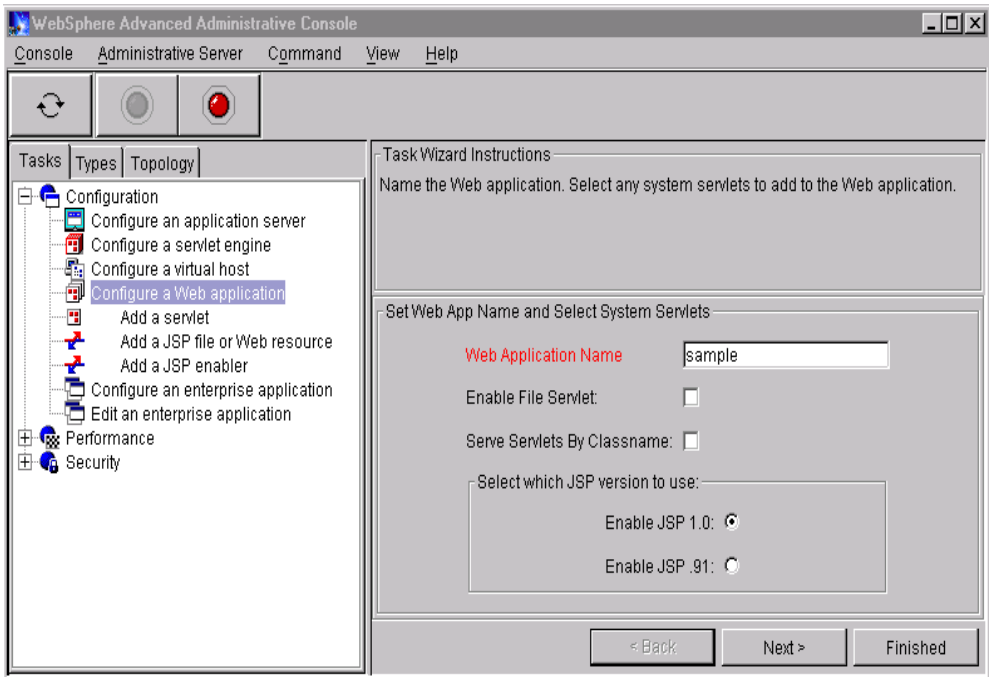


Figure 54. WebSphere Advanced Administrative Console

If you click **Topology**, your host name should be listed. Expand the view of that node, and you should see an entry called Default Server.

The WAS console is the main part of the WAS administration; this is where you define the EJBs, the servlet, and so on.

6.2 Troubleshooting

In this section we provide hints and tips on common problems.

6.2.1 Classpath checking

The Java classpath environment variable contains path information similar to that for the PATH variable used with Windows and OS/2. The classpath contains the paths to your Java class library, which can be:

- Standard workstation directories
- Java archives (.jar files)
- Compressed ZIP files (.zip files)

You can check the currently used classpath by opening a command prompt window and issuing the `SET CLASSPATH` command, as shown in Figure 55.

```
C:\>SET CLASSPATH
CLASSPATH=C:\lpex\bin;C:\jdk1.1.7\lib\classes.zip;c:\vsecon\Navigator.jar;
c:\vsecon\VSEConnector.jar;
```

Figure 55. SET CLASSPATH command

When you install new class libraries like the VSE Connector Client, you must ensure that the name of the directory where the new classes reside is part of the classpath variable. Depending on your operating system, examine either CONFIG.SYS (OS/2) or AUTOEXEC.BAT (Windows 95/98) for the set classpath entry.

On Windows NT, you need to select **System Properties**. On the Environment tab, click the name of the variable you want to change.

Modify the string in value textfield and click **Set --> Apply** to make a permanent change and an immediate update.

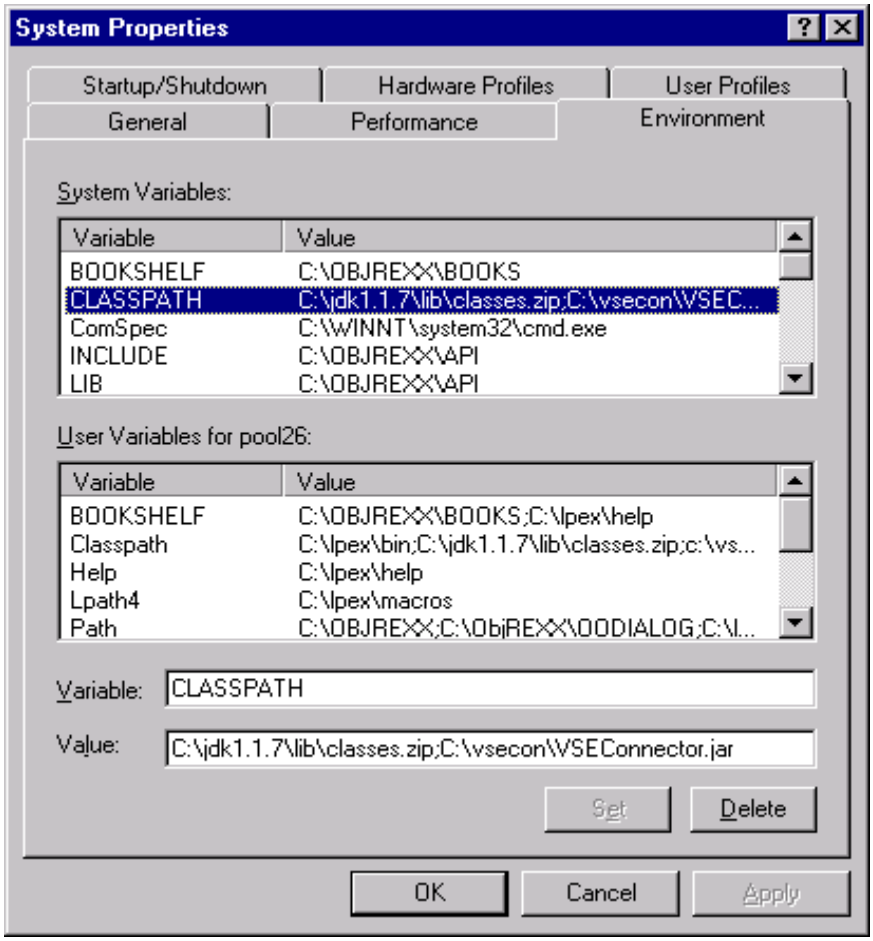


Figure 56. Windows NT environment panel

To examine the currently used classpath, issue the `set classpath` command from a command prompt window. This will show you the classpath values that are defined in the system environment and have been active since system start; see Figure 56.

You may change the path temporarily by setting your own user directory in front of the permanent classpath by the following command:

```
set classpath=.;%classpath%;
```

The dot after `set classpath=` represents your user directory.

This classpath is valid as long as the command prompt window is active. When you leave this session, the temporary change is also gone.

6.2.2 Netscape - and other Web browser support

Ensure that your Netscape browser version is 4.7. or higher.

The archive attribute in the HTML <applet> tag is Netscape-specific. For example, it is not supported by Microsoft Internet Explorer 3, although both IE4 and IE5 support this tag.

6.2.3 Checks when downloading phases in binary mode

When downloading a file from the VSE/ESA host by using the TCP/IP FTP utility in command prompt mode, ensure that you set binary mode on (enter `IMAGE` or `BINARY`) so that you are running in UNIX=OFF mode.

Open a command prompt window and establish a TCP/IP FTP connection to the VSE/ESA TCP/IP server. After answering the prompts for user and password, enter the required FTP settings. Issue the command `BINARY` to switch from ASCII to BINARY mode. The command `type` shows the actual setup for this.

Additionally, you should ensure that you have set UNIX=OFF. You may switch between both modes by entering `CD \` (for UNIX=OFF) or `CD /` (for UNIX=ON).

6.2.4 TCP/IP port number 2893

In TCP/IP, a port is a 16-bit number used to communicate between TCP and a higher-level protocol or application. The port numbers are divided into three ranges: the well-known ports, the registered ports, and the dynamic and/or private ports.

The well-known ports are those from 0 through 1023.

The registered ports are those from 1024 through 49151.

The dynamic and/or private ports are those from 49152 through 65535.

Refer to `http://www.area51.dk/network/port-numbers` for a more detailed description of port number assignments. Here you will find the following entries for VSE connectors:

- vseconnector 2893/tcp VSECONNECTOR
- vseconnector 2893/udp VSECONNECTOR

Note that you should *not* change the port number (2893) for VSE connectors.

6.2.5 Java, javac, and classpath

Each Java source program (.java file) must be compiled into bytecode (that is, a .class file). This can be done with javac or any other Java compiler.

javac is a base component of Java Development Toolkit (JDK™) and may be started from a command prompt window.

The compiler needs the classpath information to find all the classes to be included into the compiled bytecode. Ensure that one of the classpath entries points to the JDK home directory and its packed classfiles:

C:\jdk1.1.7\lib\classes.zip.

You may also set any directory to the current classpath by calling javac with parameter -classpath. For example:

```
javac -classpath.;C:\users\dac\classes;C:\tools\java\classes ...
```

-classpath specifies the path javac uses to look up classes that are needed to run javac, or that are being referenced by other classes you are compiling. It overrides the default as well as the classpath environment variable, if it is set.

6.2.6 JDK 1.1.7 or higher fix

The Java Development Kit (JDK™) 1.1.8 is the latest maintenance release that corrects a significant number of known problems in the JDK 1.1.7B and later releases.

You can display the current JDK release by issuing the following command from the command prompt window:

```
JAVA -VERSION (or JAVA -FULLVERSION)
```

6.2.7 Changing the application heap size

The DB2 database WebSphere Application Server (WAS) does not work with the application heap size of 256. Change the application heap size to 512 using the DB2 for Windows NT command line processor:

```
UPDATE DB CFG FOR WAS USING APPLHEAPSZ 512
```

6.2.8 IBM HTTP Server error 1069

You could get the following error message:

```
Error 1069: The service did not start due to a logon failure
```

If this happens, click **Startup** on the Services dialog box and change the password to match your Windows NT password.

6.2.9 IBM HTTP Server error 2186

You could get the following error message:

```
Error 2186:The service is not responding to the control function
```

If this happens, verify the following:

- The JDK that was installed/referenced during the HTTP server/WebSphere installation is active.
- The WebSphere administrative server service is started.

6.2.10 The WebSphere administrative server does not start (Windows NT)

Ensure that the IBM HTTP Server is running (in the Services dialog). If the IBM HTTP Server is not running, use the **Start** option in a Services dialog to start the server.

Check that WebSphere administrative server has been registered in the Services dialog. Open the Control Panel and select **Services**. If you scroll down, you should see IBM WS AdminServer. Start the service by selecting **IBM WS AdminServer**, and then selecting **Start**.

If the server is slow to start or does not start successfully, examine the last line in the \WebSphere\AppServer\logs\tracefile.log. If the trace file says *server is open for e-business*, the server has started.

Hint: You can control the server from a command line or batch file using the following commands:

```
net start "IBM WS AdminServer"
net stop "IBM WS AdminServer"
```

6.3 Performance information

This chapter describes some performance considerations in relation to VSE/VSAM and DB2 Server for VSE & VM. For more information, see the document *VSE/ESA 2.5 Performance Considerations* on the VSE home page at:

```
http://www.s390.ibm.com/vse
```

6.3.1 VSE Java-based connector - share option 4 impact

The VSE Java-based connector for VSAM, as shown in Figure 57, requires share option 4 for VSAM e-WRITEs if the file is open in CICS.

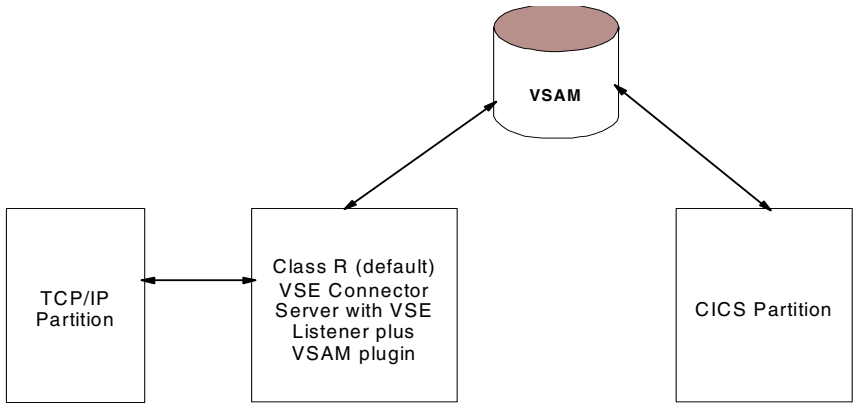


Figure 57. VSE JavaBeans connectors (VSAM)

If you need to update a VSAM file using the connector, the file must be defined with share option 4. However, VSAM share option 4 has an overhead and thus should be carefully evaluated as each READ must be done from disk. The pathlength is longer for processing UPDATE requests, and internal Control Area locks (CA- locks) may degrade performance.

A different approach might be to use a connector in such a way that the VSAM file writes take place in the file owning partition (typically CICS), thus avoiding the use of share option 4.

In any case, either try to avoid using VSAM share option 4, or reduce the performance overhead by one of the following means.

Performance hints for VSAM updates

- Use BIM-VSHARE (from BI Moyle Assoc., Inc.) in CICS as a VSAM "traffic policeman".

  VSAM batch I/Os are intercepted and sent via XPCC to a VSHARE component running under the file owning CICS. For further details, refer to:

  ```
  http://www.bimoyle.com
  ```

- Use SYSB-II for VSE (from H&W Computer Systems, Inc.).

  VSAM batch I/Os are intercepted and sent via XPCC to a CICS command level transaction in the file owning CICS. For further details, refer to:
  `http://www.hwcs.com`

- Use BSI OPTI-Cache (from Barnard Software, Inc.) as a VSAM Global Resource Pool.

  This solution avoids physical I/O overhead for share option 4. For further details, refer to:

  `http://www.bsiopti.com`

6.3.2 VSE DB2-based connectors

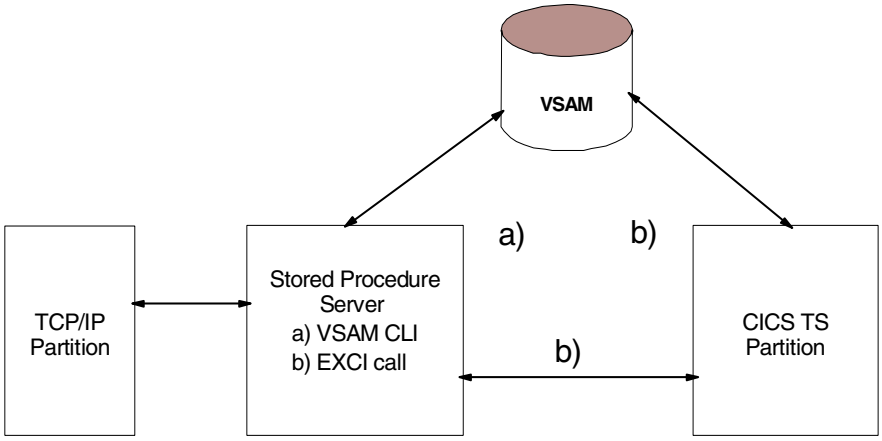
For the new VSE DB2-based connectors, only a fixed amount of data can be passed back at one time from VSAM or DL/I. Variable data lengths are not supported.

VSE stored procedure connectors are better suited for a fixed set or for smaller outputs. For more information, refer to:

- *Getting Started with DB2 Stored Procedures: Give Them a Call through the Network*, SG24-4693 (an IBM Redbook)
- *DB2 Server for VSE & VM Database Administration*

Figure 58 shows you two possibilities for VSAM file updates:

1. You can use the VSAM Call Level Interface (CLI) in a DB2 stored procedure to access and update VSAM data as if it were relationally organized. In this case, the VSAM file *must* be defined with share option 4 in VSE/ESA 2.5 (see route a in Figure 58). For more details, refer to *VSE/ESA e-business Connectors User's Guide,* SC33-6719.
2. You can write a program using an EXCI call from a batch partition, which starts a CICS TS transaction to access and update a VSAM file "owned" by CICS TS. In this case, the VSAM file does not need to be defined with share option 4 (see route b in Figure 58). For more details, refer to *CICS Transaction Server for VSE/ESA External CICS Interfaces (EXCI) Guide*, SC33-1669.



SHAROPT 4 required for UPDATEs, except access via EXCI

Figure 58. DB2-based connector

Chapter 7. Benefits

In this chapter we discuss the benefits of implementing VSE e-business connectors.

7.1 Value to the customer

VSE e-business connectors on VSE/ESA V2.5.0 deliver information or data in an easy way. The connectors close the gap between the VSE high availability platform and the new e-business technology.

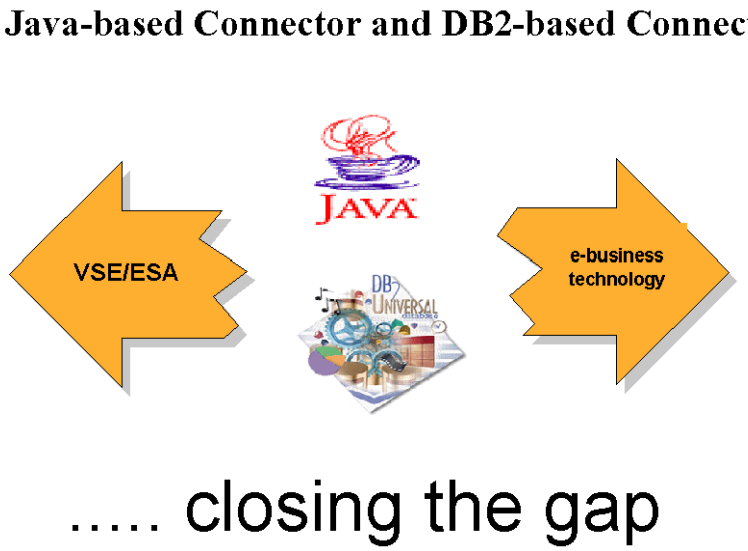


Figure 59. VSE/ESA e-business Connectors benefits

Core applications (CICS, COBOL, VSAM) typically run on the VSE/ESA system and are critical to a company's operations. They are expected to remain in production for many years to come and usually represent an enormous investment of resources.

On the other hand, e-business applications are typically based upon common standards such as TCP/IP, HTML, XML, Secure Sockets Layer (SSL), and Secure Electronic Transaction (SET), and could include both server and client code written in Java. These e-business applications could access relational data locally or remotely, and interface with end users via a standard Web browser.

The VSE/ESA e-business connectors provide you with the resources to extend your core applications to e-business applications. The VSE/ESA 2.5.0 environment is now very flexible and powerful with the functionality of the VSE/ESA e-business connectors.

There are two significant benefits of the VSE/ESA e-business connectors:

- Reducing IT costs
- Increasing the S/390 image

Now you can use the best from different platforms, as the VSE e-business Connector Client runs on various platforms including Windows, UNIX and Linux.

7.2 Range of applications

The aim of this section is to provide a brief overview of some VS/ESA e-business connectors concepts in a real-life environment.

The applications discussed in this chapter are based on flexible e-business technologies, running together with a powerful VSE/ESA V2.5.0 system. The solutions in this section are based on the Java application JDataMig, which is described in 5.6, “Sample connector application” on page 62. The source of this application is listed in Appendix B.1, “Java source: JDataMig” on page 91.

7.2.1 VSE/ESA e-business connectors and e-business technology

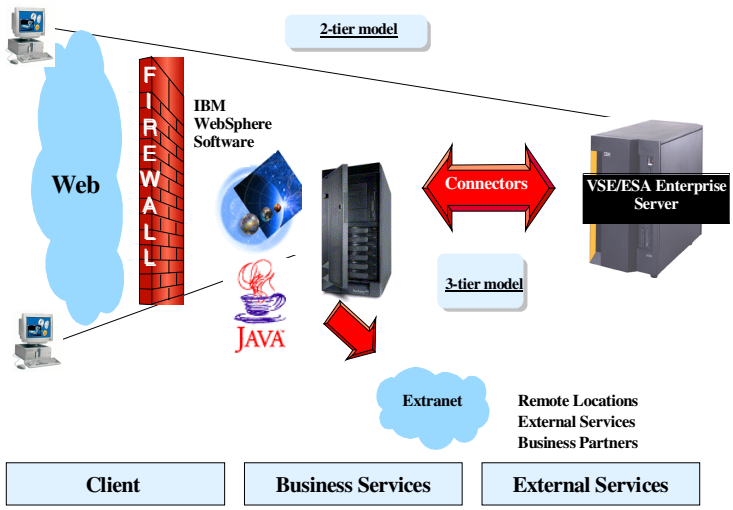


Figure 60. e-business technology

Figure 60 shows two typical scenarios using VSE/ESA e-business connectors in a two-tier or three-tier model. Using the three-tier model, we are able to exploit the JDataMig application to prepare our data for input to IBM DB2 Intelligent Miner for Data.

7.2.2 VSE/VSAM dataset in the Business Intelligence environment

IBM Business Intelligence solutions help you capitalize on the massive amounts of customer data your company acquires, so you can keep your customers loyal and stave off the competition, transforming your enterprise into a nimble, customer-focused, information-driven e-business.

In our Business Intelligence environment, the operational data is stored in a VSE/VSAM dataset on VSE/ESA V2.5.0. We replicated the data in a small Data Mart (Departmental Data Warehouse) on a Windows NT system. The data

access to VSE/ESA works with the Java-based connector and the Java application JDataMig. The application JDataMig prepares the data and exports it in the data format of IBM DB2 Intelligent Miner for Data.

Next we analyzed the data with the IBM DB2 Intelligent Miner for Data. The following steps are needed to build the Data Mart and for the IBM Intelligent Miner to process the data:

1. VSAM file mapping, with VSE/ESA Navigator (VSE/ESA Navigator is described in 5.6.1, “VSE/ESA Navigator” on page 62)
2. Running the JDataMig application to replicate the data and transform the data
3. Analyzing the data with the IBM DB2 Intelligent Miner for Data

The Intelligent Miner statistical functions provide various statistical and forecasting methods to support your business decisions. The following statistical functions are provided:

- Bivariate Statistics
- Factor Analysis
- Linear Regression
- Principal Component Analysis
- Univariate Curve Fitting

We selected Bivariate Statistics to analyze the VSE/VSAM data. In the result screen shown in Figure 61 you see the pattern of distribution. You also see the VSAM fields and the cluster characteristics. In this way you can interpret the revenue in the different countries.

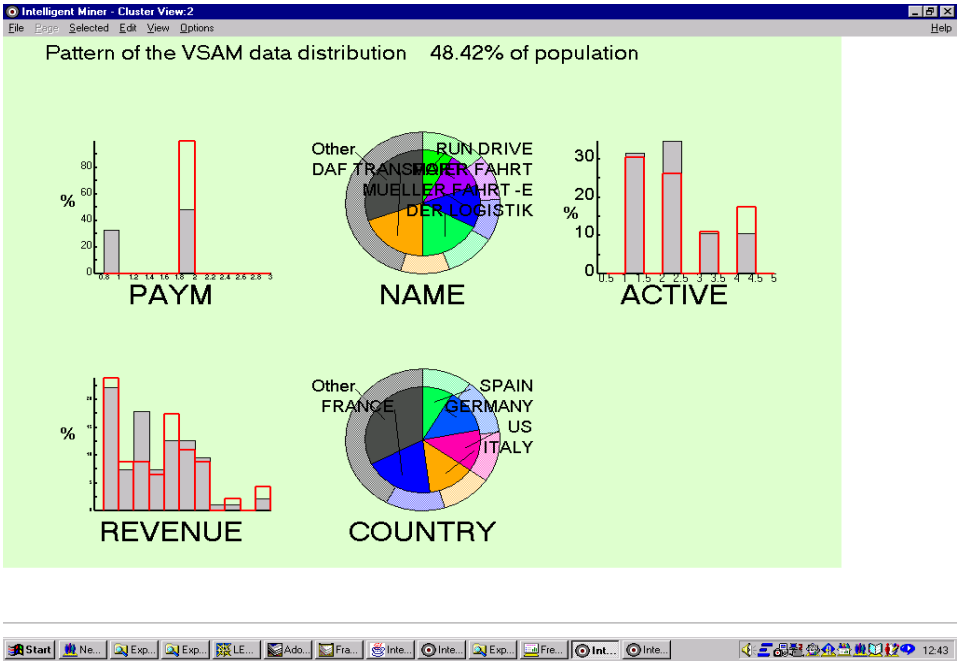


Figure 61. Results screen of the Bivariate Statistics

The results of the Bivariate Statistics in Figure 61 show that:

- 48% of customers effect payment in time.
- 30% of the 48% are active customers.
- The revenue of the 30% customers is less than 1 Million Deutschmark.
- Payment in time by French customers was better than in other countries.

DB2 IBM Intelligent Miner for Data is the best decision support system for Business Intelligence, and because Intelligent Miner is not defined for a specific industry, it helps you increase your business and reduce your risk.

7.2.3 VSE/VSAM dataset as input for Lotus 1-2-3

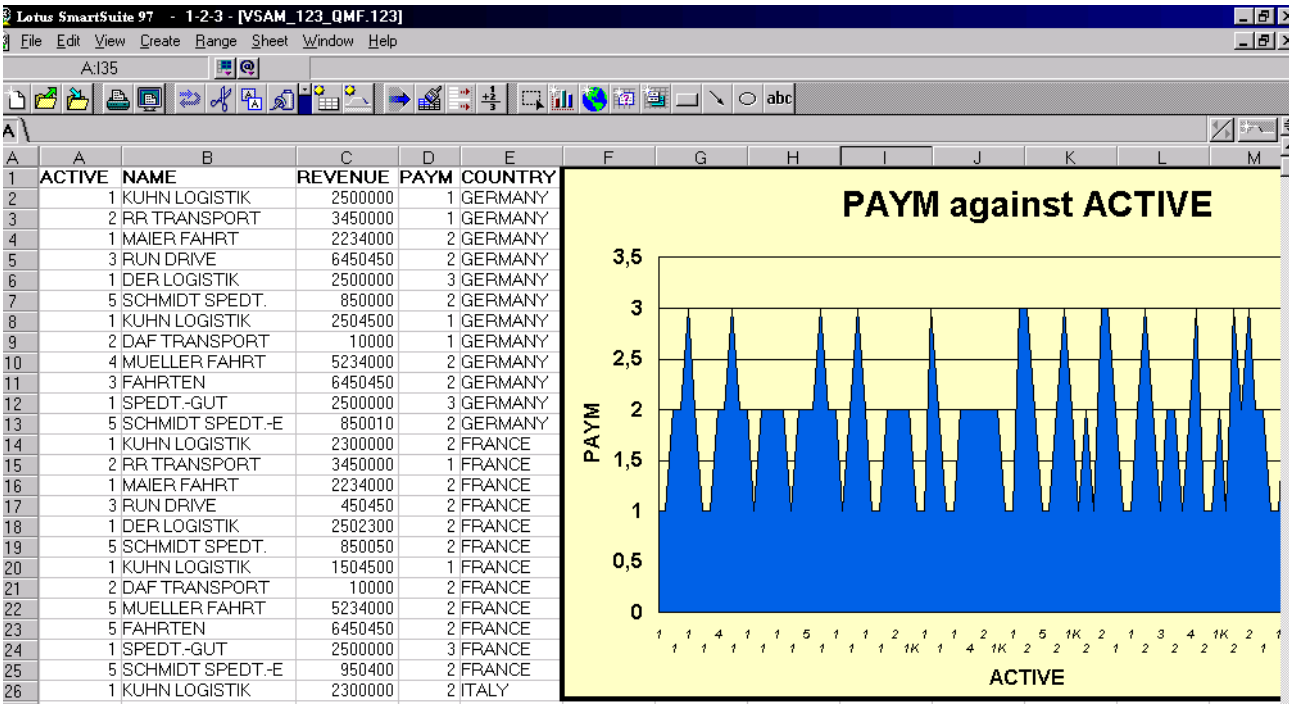
If you need to create a financial analysis or want to present the output in multiple graphic forms, you can use the spreadsheet of Lotus 1-2-3.

The Lotus 1-2-3 solution describes another way to analyze the VSAM dataset. In this scenario we also use the Java application JDataMig.

The operational data is stored in a VSE/VSAM dataset on VSE/ESA V2.5.0, and once again we access the data using JDataMig.

The following steps are necessary for the Lotus 1-2-3 application:

- Map the VSAM file with VSE/ESA Navigator.
- Run the JDataMig application to access the VSAM data and migrate to the IBM DB2 UDB for Windows NT.
- Predefine a QMF query (QMF for Windows V6.1).
- Use the Data Snap-in for QMF (which is available for download at no charge from the Resource Center for QMF at `http://www.rocketsoftware.com/qmf`) to import the data into Lotus 1-2-3.



	A	B	C	D	E
1	ACTIVE	NAME	REVENUE	PAYM	COUNTRY
2	1	KUHN LOGISTIK	2500000	1	GERMANY
3	2	RR TRANSPORT	3450000	1	GERMANY
4	1	MAIER FAHRT	2234000	2	GERMANY
5	3	RUN DRIVE	6450450	2	GERMANY
6	1	DER LOGISTIK	2500000	3	GERMANY
7	5	SCHMIDT SPEDT.	850000	2	GERMANY
8	1	KUHN LOGISTIK	2504500	1	GERMANY
9	2	DAF TRANSPORT	10000	1	GERMANY
10	4	MUELLER FAHRT	5234000	2	GERMANY
11	3	FAHRTEN	6450450	2	GERMANY
12	1	SPEDT.-GUT	2500000	3	GERMANY
13	5	SCHMIDT SPEDT.-E	850010	2	GERMANY
14	1	KUHN LOGISTIK	2300000	2	FRANCE
15	2	RR TRANSPORT	3450000	1	FRANCE
16	1	MAIER FAHRT	2234000	2	FRANCE
17	3	RUN DRIVE	450450	2	FRANCE
18	1	DER LOGISTIK	2502300	2	FRANCE
19	5	SCHMIDT SPEDT.	850050	2	FRANCE
20	1	KUHN LOGISTIK	1504500	1	FRANCE
21	2	DAF TRANSPORT	10000	2	FRANCE
22	5	MUELLER FAHRT	5234000	2	FRANCE
23	5	FAHRTEN	6450450	2	FRANCE
24	1	SPEDT.-GUT	2500000	3	FRANCE
25	5	SCHMIDT SPEDT.-E	950400	2	FRANCE
26	1	KUHN LOGISTIK	2300000	2	ITALY

Figure 62. Lotus 1-2-3 spreadsheet

After you have imported the data into the Lotus 1-2-3 spreadsheet, you can process the data as normal, producing a screen similar to the one shown in Figure 62. It is also possible to convert the VSAM data as a comma separated values (csv) file for MS Excel.

7.2.4 System management with VSE/ESA Navigator

The VSE/ESA Navigator tool provides a powerful new way to perform system management tasks using a Java application executing on a remote workstation. You can download the VSE/ESA Navigator tool to your workstation and customize it as described in 5.6.1, “VSE/ESA Navigator” on page 62.

Although there are many functions available in VSE/ESA Navigator, the following are particularly relevant for system management:

- Display host CPU activity
- Retrace products
- Monitor submitted jobs

7.2.5 Summary

The new Java-based and DB2-based connectors for VSE/ESA deliver a powerful new set of interfaces enabling you to integrate existing VSE applications into many areas of IT on other platforms. We have seen the use of VSE/ESA Navigator as a Java application, as well as the JDataMig application to prepare VSAM or DL/I data for use in remote DB2, or even PC-based, applications.

Appendix A. Setup and configuration

The listings in this appendix show the initialization member we used to configure TCP/IP for VSE/ESA, and the JCL to create the required library and sublibrary in VSE.

A.1 TCP/IP for VSE/ESA V1.4 customizing

This section contains a listing of the initialization member EBUSVSE.L stored in sublibrary PRD2.CONFIG that we used for our environment. You also find a sample initialization member IPINIT00.L in PRD1.BASE.

For editing, use the VSE Workdesk tool TCP/IP for VSE/ESA Configuration Support on your workstation or edit the member directly.

For more details, refer to the IBM Redbook *Getting Started with TCP/IP for VSE/ESA 1.4* and to the publication *TCP/IP for VSE/ESA - IBM Program Setup and Supplementary Information*.

```
* ------------------------------------------------
*  EBUSVSE.L
*
*  TCP/IP FOR VSE/ESA INITIALIZATION MEMBER
*  CREATED Tue Jul 25 12:58:35 2000
*  BY TCP/IP FOR VSE/ESA CONFIGURATION SUPPORT
*  REVISION 1.1.4 - JULY 1999, APAR PQ28003
*
* ------------------------------------------------
*  NETWORK
* ------------------------------------------------
*
SET IPADDR              =  9.164.155.55
SET MASK                =  255.255.224.0
SET MAX_SEGMENT         =  700
*
* ------------------------------------------------
*  PERFORMANCE
* ------------------------------------------------
*
SET ALL_BOUND           =  30000
SET WINDOW              =  8192
SET TRANSFER_BUFFERS    =  20
SET MAX_BUFFERS         =  100
SET RETRANSMIT          =  100
SET DISPATCH_TIME       =  30
SET REDISPATCH          =  10
*
* ------------------------------------------------
*  LINKS
* ------------------------------------------------
*
DEFINE LINK,ID=VMCTCA,TYPE=CTCA,DEV=E00
SET LINK_RETRY          =  18000
*
* ------------------------------------------------
*  ADAPTER
```

```
* ------------------------------------------------
*
*
* ------------------------------------------------
*  ROUTES
* ------------------------------------------------
*
DEFINE ROUTE,ID=VMROUTE,LINKID=VMCTCA,IPADDR=0.0.0.0, -
           GATEWAY=9.164.186.75
SET GATEWAY             =  ON
*
* ------------------------------------------------
*  TELNET DAEMONS
* ------------------------------------------------
*
DEFINE TELNETD,ID=EBUSVSE_TN,TARGET=DBDCCICS,TERMNAME=TN,PORT=23, -
           LOGMODE=S3270,LOGMODE3=D4B32783,LOGMODE4=D4B32784, -
           LOGMODE5=D4B32785,COUNT=4,IPADDR=0.0.0.0
SET TELNETD_BUFFERS     =  0
SET CONNECT_SEQUENCE    =  OFF
*
* ------------------------------------------------
*  MENUS FOR TELNET DAEMONS
* ------------------------------------------------
*
*
* ------------------------------------------------
*  FTP DAEMONS
* ------------------------------------------------
*
DEFINE FTPD,ID=EBUSVSE_ID,PORT=21,COUNT=4,UNIX=YES,TIMEOUT=36000, -
           ZERO_ERR=NO
*
* ------------------------------------------------
*  HTTP DAEMONS
* ------------------------------------------------
*
DEFINE HTTPD,ID=HTTPD_ID1,ROOT='HTTPLIB.ROOT',PORT=80,CONFINE=NO
*
* ------------------------------------------------
*  GOPHER DAEMONS
* ------------------------------------------------
*
*
* ------------------------------------------------
*  LINE PRINTER DAEMONS
* ------------------------------------------------
*
*
* ------------------------------------------------
*  EVENTS
* ------------------------------------------------
*
*
* ------------------------------------------------
*  FILE SYSTEM
* ------------------------------------------------
*
```

```
DEFINE FILESYS,LOCATION=SYSTEM,TYPE=PERM
*
* ------------------------------------------------
*  FILES
* ------------------------------------------------
*
DEFINE FILE,TYPE=POWER,PUBLIC='POWER',ALLOWSITE=NO,LRECL=80
DEFINE FILE,TYPE=LIBRARY,DLBL=HTTPLIB,PUBLIC='HTTPLIB',ALLOWSITE=NO, -
          LRECL=80
DEFINE FILE,TYPE=LIBRARY,DLBL=PRD1,PUBLIC='PRD1',ALLOWSITE=NO,LRECL=80
DEFINE FILE,TYPE=LIBRARY,DLBL=PRD2,PUBLIC='PRD2',ALLOWSITE=NO,LRECL=80
*
MODIFY FILE,PUBLIC='POWER'
MODIFY FILE,PUBLIC='HTTPLIB'
MODIFY FILE,PUBLIC='PRD1'
*
* ------------------------------------------------
*  NETWORK FILE SYSTEM (NFS)
* ------------------------------------------------
*
*
* ------------------------------------------------
*  MESSAGE SETTINGS
* ------------------------------------------------
*
SET PING_MESSAGE        =  ON
*
* ------------------------------------------------
*  SECURITY SETTINGS
* ------------------------------------------------
*
SET SECURITY            =  OFF
SET SECURITY_ARP        =  OFF
SET SECURITY_IP         =  OFF
SET ISOLATION           =  OFF
*
* ------------------------------------------------
*  OPERATION AND STARTUP RELATED SETTINGS
* ------------------------------------------------
*
SET CONSOLE_HOLD        =  OFF
SET RECORD              =  OFF
*
* ------------------------------------------------
*  PERMITTED USER ID
```

A.2 Define the HTTP library

The following job is a sample to define the library and sublibrary that we used for the HTTP daemon (see 4.3.2.3, “TCP/IP for VSE/ESA customizing” on page 16).

```
* $$ JOB JNM=LIBDHDW,CLASS=0,DISP=D,NTFY=YES
 // JOB  DEFINE HTTP-LIB
 // EXEC IDCAMS,SIZE=AUTO
  DEFINE CLUSTER ( -
```

```
        NAME (VSE.HTTPLIB.LIBRARY                          ) -
        CYL    (10 1  ) -
        SHAREOPTIONS (3) -
        RECORDFORMAT (NOCIFORMAT        ) -
        VOLUMES (SYSWK2 ) -
        NOREUSE -
        NONINDEXED -
        TO (99366  )) -
        DATA (NAME (VSE.HTTPLIB.LIBRARY.@D@                      ) ) -
        CATALOG (VSESP.USER.CATALOG                         )
  IF LASTCC NE 0 THEN CANCEL JOB
 /*
 // OPTION STDLABEL=ADD
 // DLBL HTTPLIB,'VSE.HTTPLIB.LIBRARY',,VSAM,                          X
               CAT=VSESPUC,DISP=(OLD,KEEP)
 /*
 // EXEC IESVCLUP,SIZE=AUTO
 A VSE.HTTPLIB.LIBRARY                          HTTPLIB VSESPUC OLD KEEP
/*
 // EXEC LIBR,PARM='MSHP'
        DEFINE LIB=HTTPLIB   REPLACE=YES
        DEFINE SLIB=HTTPLIB.ROOT
 /*
 /&
 * $$ EOJ
```

Appendix B. Sample source files

This appendix contains samples of the source of the programs that we used in this redbook:

1. JDataMig (see 5.6.2, “JDataMig (Java data migration application)” on page 67)
2. vsamdisa.java (see 5.1.4.2, “Displaying VSAM data with the Java applet vsamdisa” on page 42)
3. VSAMDisplayer servlet (see 5.1.4.4, “Java servlets” on page 43)

These examples are available for download at the VSE/ESA Home page at:
`http://www.s390.ibm.com/vse/`

As more samples are developed, they will also be available at the VSE/ESA Home page.

B.1 Java source: JDataMig

This source file is the major part of the JDataMig application.

```
package com.ibm.redbook.datamig.cmdline;

import java.io.*;
import java.util.*;
import java.text.*;

import com.ibm.redbook.datamig.core.*;

public class JDataMigLite {

public static void main(String[] args) {

JDataMigLite my = new JDataMigLite();
my.doIt(args);


private SourceSystem source = new SourceSystem();
private boolean sconnect = false;
private DestinationSystemDB2 dest = null;
private boolean dconnect = false;

private String ConfFile = "";
private boolean batchMode = false;

public JDataMigLite() {}

public void doIt(String[] args) {

log("***********************************************");
log("** IBM DataMigrator for Java/VSE Lite Edition **");
log("***********************************************");
log(" ");

processArgs(args);
if(!ConfFile.equals("")) {
try {
```

```
ConfigFileReader conf = new ConfigFileReader(ConfFile);
conf.readConfigFile();
this.source = conf.getSource();
this.dest = conf.getDestinationDB2();
sconnect = true;
}
catch (Exception e) {
log("!! an error occured:" + e.toString());
}
} else {
this.source.ActionType = this.source.ACTION_EXP_DB2;
}

while(!sconnect) {
readVSEParams();
}
while(!dconnect) {
readDB2Params();
}

if(source.getCatalogName().equals("")) {
source.setCatalogName(readln("enter VSAM-catalog to use (l = don't know, show
me a list)"));
if(source.getCatalogName().equalsIgnoreCase("l")) {
try {
log("listing catalogs");
source.listCatalogs();
}
catch (Exception e) {
log("!! an error occured:" + e.toString());
}
for(int i=0; i<source.noOfCatalogs(); i++) {
log(i + " " + source.getCatalogEntry(i).getFileID());
}
source.useCatalogEntryAt(readint("enter the number of the catalog to use"));
}
}

if(source.getClusterName().equals("")) {
source.setClusterName(readln("enter VSAM-cluster to use (l = don't know, show
me a list)"));
if(source.getClusterName().equalsIgnoreCase("l")) {
try {
log("listing clusters");
source.listClusters();
}
catch (Exception e) {
log("!! an error occured:" + e.toString());
}
for(int i=0; i<source.noOfClusters(); i++) {
log(i + " " + source.getClusterEntry(i).getFileID());
}
source.useClusterEntryAt(readint("enter the number of the cluster to use"));
}
}

if(source.getMapName().equals("")) {
source.setMapName(readln("enter map to use (l = don't know, show me a list)"));
```

```
if(source.getMapName().equalsIgnoreCase("l")) {
try {
log("listing maps");
source.listMaps();
}
catch (Exception e) {
log("!! an error occured:" + e.toString());
}
for(int i=0; i<source.noOfMaps(); i++) {
log(i + " " + source.getMapEntry(i).getName());
}
source.useMapEntryAt(readint("enter the number of the map to use"));
}
}

log("VSAM catalog: " + source.getCatalogName());
log("VSAM cluster: " + source.getClusterName());
log("VSAM map    : " + source.getMapName());
try {
log("VSAM fields : " + source.Source.getNoOfFields());
}
catch (Exception e) {
log("!!error. one of the vsam-parms is wrong!!");
System.exit(0);
}

try {
if(!batchMode) {
System.out.print("i have everything; press any key to start migration");
System.in.read();
}

log("creating destination table");
dest.logUseStdOut = true;
dest.Source = source.Source;
dest.createTable();
source.logUseStdOut = true;
source.Destination = dest;
source.copyData();

log("done successfully. disconnecting");
source.disconnect();
dest.disconnect();
}
catch (Exception e) {
log("!! an error occured:" + e.toString());
}
}

private void processArgs(String[] args) {

for(int i=0; i<args.length; i++) {
if(args[i].equals("-f")) {
this.ConfFile = args[i+1];
log("using config-file: " + this.ConfFile);
}
if(args[i].equals("-batch")) {
this.batchMode = true;
```

```
log("i am in batch-mode. no prompts!");
}
}
}

private void readVSEParams() {

if(source==null) source = new SourceSystem();
if(source.Host.equals(""))
source.Host = readln("enter Hostname/IP-Address of VSE-System");
if(source.User.equals(""))
source.User = readln("enter UserID to use for connection");
if(source.Pwd.equals(""))
source.Pwd  = readln("enter Password for this UserID");

try {
log("trying to connect to VSE: " + source.User + "@" + source.Host);
source.connect();
sconnect=true;
log("connected to VSE");
}
catch (Exception e) {
log("!! an error occured:" + e.toString());
sconnect=false;
source.Host = "";
source.User = "";
source.Pwd = "";
}
}

private void readDB2Params() {

if(dest==null) dest = new DestinationSystemDB2();
if(dest.DriverToUse==dest.DRIVER_NONE)
dest.DriverToUse = readint("enter Driver type to use.\n 1 = App Driver\n 2 =
Net Driver");
if(dest.Database.equals(""))
dest.Database = readln("enter DB/2-Database to use");
if(dest.User.equals(""))
dest.User     = readln("enter UserID to use for DB/2");
if(dest.Pwd.equals(""))
dest.Pwd      = readln("enter Password for this UserID");
if(dest.DriverToUse==dest.DRIVER_NET) {
if(dest.Host.equals(""))
dest.Host = readln("enter Hostname to connect to");
if(dest.Port==0)
dest.Port = readint("enter Portnumber (e.g. 6789)");
}

try {
log("trying to connect to DB/2: " + dest.User + "@" + dest.Database);
dest.connect();
dconnect=true;
log("connected to DB/2");
}
catch (Exception e) {
log("!! an error occured:" + e.toString());
dconnect=false;
```

```
dest.Database = "";
dest.User = "";
dest.Pwd = "";
}
}

private String readln(String print) {

if(batchMode) {
log("batch mode: insufficent information. quit.");
System.exit(0);
}

byte[] input;
String out = "";

while(out.equals("")) {
System.out.print(print + " (q = quit): ");
try {
input = new byte[100];
System.in.read(input);
}
catch (Exception e) {
return "";
}
out = new String(input).trim();
if(out.equalsIgnoreCase("q")) System.exit(0);
}

return out;
}

private int readint(String print) {

return Integer.parseInt(readln(print));
}

private void log(String text) {

System.out.println(text);
}
}
```

B.2 Java source: VSE/VSAM access and display VSE/VSAM records

This source file, vsamdisa.java (see 5.1.4.2, “Displaying VSAM data with the Java applet vsamdisa” on page 42), is based on the VSAMDisplayExample.java.

The Java program is a command line version. With this program, you can see how to access VSE/VSAM datasets and how to display the VSE/VSAM records. The Java program accesses the VSESP.USER.CATALOG by default.

The input parameters for this Java program are:

- VSE IP address:
- VSE file name : (Sample : JAVAFILE.SAMPLE.FILEA)

- VSE map name : (Sample : FILEAMAP)
- VSE user ID:
- Password:

```
/*********************************************************************** */
/*           VSE/ESA VSAM variable Display Sample Code  ****************** */
/* MODULE NAME : vsamdisa.java                             ********** */
/* RELEASE : VSE/ESA Version 2.5.0                        ********** */
/* COMPILER : ************************************************************ */
/*      Java 1.1.8 ********************************************************* */
/*  REDBOOK-TEAM ********************************************************** */
/*********************************************************************** */
package com.ibm.vse.samples;

import java.lang.*;
import java.net.*;
import java.io.*;
import java.util.*;

/* Import IBM Common Connector Framework (CCF) classes */
import com.ibm.connector.internal.*;

/* Import VSE Connector classes */
import com.ibm.vse.connector.*;

public class vsamdisa
{
   /**
    */
   public static void main(String argv[]) throws IOException
   {
      VSEConnectionSpec spec;
      VSESystem system;
      VSEVsam vsam;
      VSEVsamCatalog catalog;
      VSEVsamCluster cluster;
      VSEVsamRecord record, newRec=null;
      VSEVsamMap map;
      RecordListener rl;
      byte[] inputArray;
      String ipAddr, userID, password;
      Vector vRecords;
      String RecNum, newField="";

      String catName = "VSESP.USER.CATALOG";
      String fileName = "";
      String mapName = "";

      /* Prompt for IP address, user ID and password. We have to remove */
      /* trailing blanks from the input strings before using them. */
      inputArray = new byte[15];
      System.out.println("Please enter your VSE IP address:");
      System.in.read(inputArray);
      ipAddr = new String(inputArray);
      inputArray = new byte[30];
      System.out.println("Please enter your VSE file name :");
```

```
      System.in.read(inputArray);
      fileName = new String(inputArray);
      inputArray = new byte[15];
      System.out.println("Please enter your VSE map name :");
      System.in.read(inputArray);
      mapName = new String(inputArray);
      inputArray = new byte[8];
      System.out.println("Please enter your VSE user ID:");
      System.in.read(inputArray);
      userID = new String(inputArray);
      System.out.println("Please enter password:");
      System.in.read(inputArray);
      password = new String(inputArray);
      mapName = mapName.trim();
      fileName = fileName.trim();
      userID = userID.trim();
      password = password.trim();
      ipAddr = ipAddr.trim();

      /* Create connection specification. The connection spec */
      /* holds information about the physical host connection and is */
      /* stored permanently in the Common Connector Framework (CCF) */
      System.out.println("Creating connection and VSE system ...");
      try {
      spec = new VSEConnectionSpec(InetAddress.getByName(ipAddr),
                                          2893,userID,password);
      }
      catch (UnknownHostException e)
      {
         System.out.println("Unknown host : " + e);
         return;
      }
      spec.setReapTime(5);
      spec.setMaxConnections(5);
      spec.setMinConnections(0);
      spec.setUnusedTimeout(10000);

/* Stay logon with this user for lifetime of this connection */
      spec.setLogonMode(true);

      /* Create VSE system instance with this connection */
      system = new VSESystem(spec);
      if (!(system.getConnectionManager()
            instanceof com.ibm.connector.connectionmanager.ConnectionManager))
      {
         system.setConnectionManager(
                  new
com.ibm.connector.connectionmanager.ConnectionManager());
      }

      /* Get file system from host */
      System.out.println("Getting records from " + fileName + "...");
      vsam = system.getVSEVsam();

      /* Step 1: Create a listener that gets notified when objects */
      /* are retrieved from the host */
      rl = new RecordListener();
      vsam.addVSEResourceListener(rl);
```

```
      /* Step 2: Get VSAM records from host using the given map */
      map = new VSEVsamMap(system, catName, fileName, mapName);
      cluster = new VSEVsamCluster(system, catName, fileName);
      cluster.addVSEResourceListener(rl);
      cluster.selectRecords(map);
      cluster.removeVSEResourceListener(rl);

      /* Get vector containing all records from listener */
      vRecords = rl.getRecords();

      /* Step 3: Display records ... */
      int numMapFields = map.getNoOfFields();
      System.out.println("Records in file " + fileName + " :");
      for (int k=0;k<vRecords.size();k++)
      {
         System.out.println("Record " + k + ":");
         record = (VSEVsamRecord)(vRecords.elementAt(k));
         for (int i=0;i<numMapFields;i++)
         {
            try {
               if (map.isFieldPartOfPrimaryKey(i))
                  System.out.println(map.getFieldName(i) + " (Key) : " +
record.getField(i).toString());
               else
                  System.out.println(map.getFieldName(i) + " : " +
record.getField(i).toString());
            }
            catch (Exception e)
            {
               System.out.println("writeRecord() : " + e);
            }
         }
         System.out.println("------------------------------");
      }


/* Step 4: Add a new flight to the cluster ... */
 boolean done = false;
      while (!done)
      {
         inputArray = new byte[map.getFieldLength(0)];
         System.out.println("(Please enter any key to quit)");
         System.in.read(inputArray);
         RecNum = new String(inputArray);
         RecNum = RecNum.trim();
         if ((new Integer(RecNum).intValue()) < 0)
            return;
      }
   }
}
```

A screen shot of the result screen is shown in Figure 63 on page 99:

Figure 63. Display a VSAM record

B.3 Java source: VSAMDisplayer servlet

Following is the complete code for the servlet referred to in 5.1.4.4, “Java servlets” on page 43:

```
package com.ibm.redbook.servlet;


import java.io.*;

import java.util.*;

import com.ibm.redbook.utils.*;

import javax.servlet.*;

import javax.servlet.http.*;


public class VSAMDisplayer extends HttpServlet {


private String propfile = "";

private VSE vse = null;

private int err = 0;


  public String getServletInfo() {


    return "VSAM DIsplayer - displays VSAM data from a VSE-server";

  }
```

```
  public void init() throws ServletException {
  super.init();
  /*getServletContext().log("try to get properties");
  propfile = getInitParameter("propfile");
  getServletContext().log("try to read " + propfile);
  Properties prop = new Properties();
  try {
prop.load(new FileInputStream(new File(propfile)));
vse = new VSE(prop.getProperty("vse.host"),
 Integer.parseInt(prop.getProperty("vse.port")),
 prop.getProperty("vse.user"),
 prop.getProperty("vse.pwd"));
if(!vse.connect()) {
getServletContext().log("unable to connect");
} else {
getServletContext().log("connected successfully");
}
  }
  catch(Exception e) {
  getServletContext().log("error during init: " + e.toString());
  e.printStackTrace();
  }*/
  }

  public void doGet(HttpServletRequest request, HttpServletResponse response)
throws IOException {
    performTask(request,response);
 }public void doPost(HttpServletRequest request, HttpServletResponse response)
throws IOException {
performTask(request,response);
}
public void performTask(HttpServletRequest req, HttpServletResponse response)
throws IOException {
```

```
  response.setContentType("text/html");
  ServletOutputStream out = response.getOutputStream();
  out.println("<html><head><title>VSAM Displayer</title></head><body>");
  out.println("<form action='/redbook/servlet/VSAMDisplayer' method='post'>");
  this.err=0;
  HttpSession sess = req.getSession(true);
  this.vse = (VSE)sess.getValue("vseobj");
  if(this.vse==null) this.vse = new VSE();

  int page = Utils.string2int(req.getParameter("page"), 0);
  switch(page) {
  case 0:
//this.vse = new VSE();
sendForm1(out);
break;
  case 1:
if(req.getParameter("ip").equals(""))
sendError(out, "an Host or IP is needed");
 if(req.getParameter("user").equals(""))
sendError(out, "specify a user");
if(req.getParameter("pwd").equals(""))
sendError(out, "specify a password");
if(this.err>0)
sendForm1(out);
else {
this.vse.setHost(req.getParameter("ip"));
this.vse.setUser(req.getParameter("user"));
this.vse.setPassword(req.getParameter("pwd"));
if(this.vse.connect()) {
this.vse.listCatalogs();
} else {
out.println("unable to connect");
}
```

```
sendForm2(out);
}
break;
  case 2:
if(req.getParameter("catalog")==null)
sendError(out, "select a catalog");
if(this.err>0)
sendForm2(out);
else {
this.vse.setupCatalog(req.getParameter("catalog"));
this.vse.listClusters();
sendForm3(out);
}
break;
 case 3:
if(req.getParameter("cluster")==null)
sendError(out, "select a cluster");
if(this.err>0)
sendForm3(out);
else {
this.vse.setupCluster(req.getParameter("cluster"));
this.vse.listMaps();
sendForm4(out);
 }
break;
  case 4:
if(req.getParameter("map")==null)
sendError(out, "select a map");
if(this.err>0)
sendForm4(out);
else {
this.vse.setupMap(req.getParameter("map"));
this.vse.listData();
```

```
sendForm5(out);
}
break;
  default:
out.println("invalid page=" + page);
break;
  }
```

```
  sess.putValue("vseobj", this.vse);

  out.close();

}

private void sendForm1(ServletOutputStream out)

throws IOException {


out.println("<input type='hidden' name='page' value='1'>");

  out.println("<table>");

  out.println("<tr><td>Host or IP:</td><td><input type='text' size='20'
name='ip'></td></tr>");

  out.println("<tr><td>User:</td><td><input type='text' size='20'
name='user'></td></tr>");

  out.println("<tr><td>Password:</td><td><input type='password' size='20'
name='pwd'></td></tr>");

  out.println("<tr><td colspan='2' align='center'><input type='submit'
value='Connect!'></td></tr>");

  out.println("</table>");


}private void sendForm2(ServletOutputStream out)

throws IOException {


  out.println("<input type='hidden' name='page' value='2'>");

  out.println("<table>");

  out.println("<tr><td>Host or IP:</td><td>" + this.vse.getHost() +
"</td></tr>");

  out.println("<tr><td>User:</td><td>" + this.vse.getUser() + "</td></tr>");

  //out.println("<tr><td>Password:</td><td>" + this.vse.getPassword() +
"</td></tr>");

  out.println("<tr><td>Catalogs:</td><td><select name='catalog' size='5'>");

  for(int i=0; i<this.vse.noOfCatalogs(); i++) {

 out.println("<option value='" + this.vse.getCatalogEntry(i).getFileID() +
"'>" + this.vse.getCatalogEntry(i).getFileID() + "</option>");

  }

  out.println("</select>");

  out.println("</td></tr>");

  out.println("<tr><td colspan='2' align='center'><input type='submit'
value='List Clusters'></td></tr>");
```

```
  out.println("</table>");

  }


  private void sendForm3(ServletOutputStream out)

throws IOException {


  out.println("<input type='hidden' name='page' value='3'>");

  out.println("<table>");

  out.println("<tr><td>Host or IP:</td><td>" + this.vse.getHost() +
"</td></tr>");

  out.println("<tr><td>User:</td><td>" + this.vse.getUser() + "</td></tr>");

  //out.println("<tr><td>Password:</td><td>" + this.vse.getPassword() +
"</td></tr>");

  out.println("<tr><td>Catalog:</td><td>" + this.vse.getCatalog().getFileID()
+ "</td></tr>");

  out.println("<tr><td>Clusters:</td><td><select name='cluster' size='5'>");

  for(int i=0; i<this.vse.noOfClusters(); i++) {

  out.println("<option value='" + this.vse.getClusterEntry(i).getFileID() +
"'>" + this.vse.getClusterEntry(i).getFileID() + "</option>");

  }

  out.println("</select>");

  out.println("</td></tr>");

  out.println("<tr><td colspan='2' align='center'><input type='submit'
value='List Maps'></td></tr>");

  out.println("</table>");

  }


private void sendForm4(ServletOutputStream out)

throws IOException {

  out.println("<input type='hidden' name='page' value='4'>");

  out.println("<table>");

  out.println("<tr><td>Host or IP:</td><td>" + this.vse.getHost() +
"</td></tr>");

  out.println("<tr><td>User:</td><td>" + this.vse.getUser() + "</td></tr>");

  //out.println("<tr><td>Password:</td><td>" + this.vse.getPassword() +
"</td></tr>");
```

```
  out.println("<tr><td>Catalog:</td><td>" + this.vse.getCatalog().getFileID()
+ "</td></tr>");

  out.println("<tr><td>Cluster:</td><td>" + this.vse.getCluster().getFileID()
+ "</td></tr>");

  out.println("<tr><td>Maps:</td><td><select name='map' size='5'>");

  for(int i=0; i<this.vse.noOfMaps(); i++) {

  out.println("<option value='" + this.vse.getMapEntry(i).getName() + "'>" +
this.vse.getMapEntry(i).getName() + "</option>");

  }

  out.println("</select>");

  out.println("</td></tr>");

  out.println("<tr><td colspan='2' align='center'><input type='submit'
value='Show data'></td></tr>");

  out.println("</table>");

  }


  private void sendForm5(ServletOutputStream out)

throws IOException {


  out.println("<input type='hidden' name='page' value='0'>");

  out.println("<table>");

  out.println("<tr><td>Host or IP:</td><td>" + this.vse.getHost() +
"</td></tr>");


  out.println("<tr><td>User:</td><td>" + this.vse.getUser() + "</td></tr>");

  //out.println("<tr><td>Password:</td><td>" + this.vse.getPassword() +
"</td></tr>");

  out.println("<tr><td>Catalog:</td><td>" + this.vse.getCatalog().getFileID()
+ "</td></tr>");

  out.println("<tr><td>Cluster:</td><td>" + this.vse.getCluster().getFileID()
+ "</td></tr>");

  out.println("<tr><td>Map:</td><td>" + this.vse.getMap().getName() +
"</td></tr>");

  out.println("</table>");

  out.println("<table>");

  out.println("<tr>");

  for(int i=0; i<this.vse.getMap().getNoOfFields(); i++) {
```

```
  out.println("<th>" + this.vse.getMap().getFieldName(i) + "<br>" +

 VSE.getFieldTypeText(this.vse.getMap().getFieldType(i)) +

 "(" + this.vse.getMap().getFieldLength(i) + ")<br>" +

 (this.vse.getMap().isFieldPartOfPrimaryKey(i) ? "PK" : "") +

 "</th>");

  }

  out.println("</tr>");

  for(int i=0; i<this.vse.noOfData(); i++) {

  out.println("<tr>");

  for(int j=0; j<this.vse.getMap().getNoOfFields(); j++) {

 out.println("<td>" + (String)this.vse.getDataEntry(i).getField(j).toString()
+ "</td>");

  }

  out.println("</tr>");

  }

  out.println("</table>");

  out.println("<tr><td colspan='2' align='center'><input type='submit'
value='New Query'></td></tr>");

  out.println("</table>");

  }


  private void sendError(ServletOutputStream out, String err)

throws IOException {

this.err++;

  out.println("<font color='red'>Error: " + err + "</font><br>");

  }

}
```

Appendix C. Special notices

This publication is intended to help customers, business partners, and IBMers to install, configure and deploy e-business connectors in the VSE/ESA environment. The information in this publication is not intended as the specification of any programming interfaces that are provided by VSE/ESA Version 2 Release 5. See the PUBLICATIONS section of the IBM Programming Announcement for VSE/ESA Version2 Release 5 more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

IBM ®	AIX
AS/400	AT
CICS	CT
DB2	DB2 Connect
DB2 Universal Database	Distributed Relational Database Architecture
DRDA	Enterprise Storage Server
Intelligent Miner	MQSeries
Netfinity	Nways
OS/2	OS/390
OS/400	QMF
RS/6000	S/390
SP	System/390
VisualAge	VM/ESA
VSE/ESA	VTAM
WebSphere	Wizard
XT	400
Lotus	1-2-3
Redbooks	Redbooks Logo 

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Appendix D. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

D.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 113.

- *e-business Solutions for VSE/ESA*, SG24-5662
- *Getting Started with TCP/IP for VSE/ESA 1.4*, SG24-5626
- *CICS TS New Functions*, SG24-5997
- *Getting Started with DB2 Stored Procedures: Give Them a Call through the Network*, SG24-4693
- *Servlet and JSP Programming with IBM WebSphere Studio and VisualAge for Java*, SG24-5755

D.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at ibm.com/redbooks for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
IBM System/390 Redbooks Collection	SK2T-2177
IBM Networking Redbooks Collection	SK2T-6022
IBM Transaction Processing and Data Management Redbooks Collection	SK2T-8038
IBM Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
IBM AS/400 Redbooks Collection	SK2T-2849
IBM Netfinity Hardware and Software Redbooks Collection	SK2T-8046
IBM RS/6000 Redbooks Collection	SK2T-8043
IBM Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

D.3 Other resources

These publications are also relevant as further information sources:

- *VSE/ESA e-business Connectors User’s Guide*, SC33-6719
- *VSE/ESA Planning* , SC33-6703
- *VSE/ESA Release Guide*, SC33-6718
- *TCP/IP for VSE Commands, Release 1.4* from Connectivity Systems, Inc.
  This document is available for download at the VSE Home page at:
  http://www.s390.ibm.com/vse/
- *TCP/IP for VSE Optional Products, Release 1.4* from Connectivity Systems, Inc.
  This document is available for download at the VSE Home page at:
  http://www.s390.ibm.com/vse/

- *TCP/IP for VSE/ESA - IBM Program Setup and Supplementary Information*, SC33-6601
- *DL/I 1.11 Release Guide*, SC33-6211
- *DL/I DOS/VS Resource Definition and Utilities*, SH24-5021
- *DB2 Connect User's Guide*, SC09-2838
- *Program Directory for DB2 Server for VSE*, GI10-4954
- *DB2 Server for VSE & VM Database Administration*, SC09-2888
- *CICS Transaction Server for VSE/ESA Internet Guide*, SC34-5765
- *CICS Transaction Server for VSE/ESA External CICS Interfaces (EXCI) Guide*, SC33-1669
- *VSE/ESA Version 2.5 Performance Considerations* on the VSE Home Page at `http://www.s390.ibm.com/vse/`

D.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- `http://www.s390.ibm.com/vse/`
- `http://www.s390.ibm.com/products/vse/support/vseconn/vsecon.htm`
- `http://www.s390.ibm.com/products/vse/support/vseconn/vsenavi.htm`
- `http://www.s390.ibm.com/products/vse/support/tcpip/iestcp.htm`
- `http://www.ibm.com/software/webservers/appserv/doc/v302/install/install_guides.html`
- `http://www.ibm.com/java/jdk/`
- `http://www.ibm.com/java/jdk/download/index.html`
- `http://www.ibm.com/db2/`
- `http://www.ibm.com/software/ts/cics/`
- `http://www.ibm.com/ftp/vse/docs/vsenew19/vsedtsw.pdf`
- `http://www.area51.dk/network/port-numbers/`
- `http://www.hwcs.com/`
- `http://www.bsiopti.com/java/jdk/`
- `http://www.bimoyle.com/java/jdk/`
- `http://www.rocketsoftware.com/qmf/`

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** ibm.com/redbooks

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States or Canada	pubscan@us.ibm.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number _____

☐ Credit card number _____

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Abbreviations and acronyms

APAR	authorized program analysis report
API	application programming interface
APPC	Advanced Peer-to-Peer Communication
ASCII	American Standard Code for Information Interchange
AIX	Advanced Interactive Executive
BMS	basic mapping support
BSC	binary synchronous communication
CA	control area
CAE	Client Application Enabler
CD	compact disc
CGI	Common Gateway Interface
CLI	call level interface
CICS	Customer Information Control System
CLAW	Common Link Access to Workstation
CLP	command line processor
COBOL	Common Business Oriented Language
CORBA	Common Object Request Broker
CPU	central processing unit
CSI	Connectivity Systems, Inc.
CTG	CICS Transaction Gateway
CTCA	channel-to-channel adapter
CWS	CICS Web Support
DASD	direct access storage device
DBA	database administrator
DBD	Database description
DB2	Database 2
DB2 UDB	Database 2 Universal Database
DL/I	Data Language 1
DRDA	Distributed Relational Database Architecture
EBCDIC	Extended Binary Communication Data Interchange Code
ECI	external call interface
EJB	Enterprise JavaBean
EPI	external presentation interface
ESDS	Entry Sequenced Data Set
ESS	Enterprise Storage System
EXCI	external CICS interface
FCT	file control table
FTP	File Transfer Protocol
GIF	graphical user interface
GPS	General Print Server
GUI	graphical user interface
HLASM	high-level assembler
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
I/O	input/output
IBM	International Business Machines Corporation
ICCF	Interactive Computing and Control Facility
IP	Internet Protocol
IPL	initial program load
ISQL	Interactive Structured Query Language
ITSO	International Technical Support Organization
IUI	interactive user interface
IXFP	IBM Extended Facilities Product
JCL	job control language
JDBC	Java Database Connectivity
JDK	Java Development Kit
JRE	Java runtime environment
JSP	Java Server Page
JVM	Java Virtual Machine
KSDS	Key Sequenced Data Set
LAN	local area network
LPD	Line Printer Daemon
LPR	Line Printer Requester
LSR	local shared resources

MB	Megabyte
MQI	message queue interface
MSHP	Maintain System History Program
MPS	multiple partition support
MVS	Multiple Virtual Storage Operating System
NFS	Network File System
ODBC	open database connectivity
ORA	online resource adapter
OS/2	Operating System/2
OSA	Open Systems Adapter
OS/390	Operating System for the System/390 platform
OS/400	Operating System for the AS/400 platform
PC	personal computer
PSB	program specification block
PTF	program temporary fix
PU	physical unit
RS/6000	IBM RISC System/6000
SD	Solution Developer (formerly Independent Software Vendor (ISV))
SET	secure electronic transaction
SIT	system intialization table
SNA	Systems Network Architecture
SQL	Structured Query Language
SSL	Secure Sockets Layer
SVA	shared virtual area
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol /Internet Protocol
UDB	Universal Database
UI	user interface
URL	Uniform Resource Locator
VM	virtual machine
VM/ESA	Virtual Machine/Enterprise Systems Architecture
VSAM	Virtual Storage Access Method
VSE/ESA	Virtual Storage Extended/Enterprise Systems Architecture
VSE/POWER	Virtual Storage Extended/Priority Output Writers, Execution processor, and input Readers
VSE/VSAM	Virtual Storage Extended/Virtual Storage Access Method
VTAM	Virtual Telecommunications Access Method
WAS	WebSphere Application Server
WWW	World Wide Web
XML	Extensible Markup Language
XPCC	cross-partition communication

Index

W

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at ibm.com/redbooks
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-5950-00 e-business Connectivity for VSE/ESA
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. ibm.com/privacy/yourprivacy/

(0.2”spine)
0.17”<->0.473”

e-business Connectivity for VSE/ESA

Redbooks

VSE e-business connectors enhance interoperability

Learn to install/configure these connectors & required products

Java- and DB2-based examples

The VSE customer environment is increasingly characterized by multiple servers - including different types of servers - and the need to exploit Internet technologies. Interoperability for VSE environments is now enhanced with the addition of VSE e-business connectors in VSE/ESA 2.5.

This IBM Redbook gives you a broad understanding of the new VSE e-business connectors: the Java-based connector and the DB2-based connector.

Detailed usage examples help you install, configure, and set up these connectors and required products.

This publication complements the redbook *e-business Solutions for VSE/ESA*.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-5950-00 ISBN 0738419388