VisualAge PL/I



Remote Edit-Compile-Debug User's Guide

Version 2.1.7

VisualAge PL/I



Remote Edit-Compile-Debug User's Guide

Version 2.1.7

Note!
Before using this information and the product it supports, be sure to read the general information under "Appendix. Notices" on page 31.

Second Edition (September 2000)

This edition applies to VisualAge PL/I, Version 2.1.7, and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications by phone or fax. IBM Software Manufacturing Solutions takes publication orders between 8:30 a.m. and 7:00 p.m. Eastern Standard Time (EST). The phone number is (800) 879-2755. The fax number is (800) 445-9269.

You can also order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

A form for reader's comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation, Department HHX/H3
P.O Box 49023
San Jose, CA 95141-9023
USA

or Fax it to this U.S. number: 800-426-7773

or use the form on the Web at:

http://www.ibm.com/software/ad/rcf/

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

© Copyright International Business Machines Corporation 1995, 2000. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Remote edit-compile-debug . 1

Benefits of remote edit-compile-debug 2
Costs of remote edit-compile-debug 2
Remote editing 2
Remote compiling 3
Remote debugging 3
Comparing workstation and host concepts 4

Chapter 2. Configuring your systems . . 5

Installing and configuring the MS Loopback Adapter 5
Configuring and connecting your workstation for remote ECD 6
Verifying your connection 10

Chapter 3. Developing remotely 13

Creating an MVS project 13
Changing the source directories for your MVS project 16
Accessing host data sets 18
Mapping MVS data sets 20
Editing host data sets 21
MVS edit safeguards 23
Emulating other editors 23
Compiling host applications 23
Checking syntax on the workstation 24
Submitting jobs on the host 24
Monitoring the status of submitted jobs 25
Debugging host applications. 26
Preparing the environment for debugging . . . 26
Preparing your application for debugging . . . 27
Indicating the location of the Distributed Debugger daemon 27
Starting the remote debugging session 28
Starting your application on the host 29
Issuing host commands from the workstation . . . 29

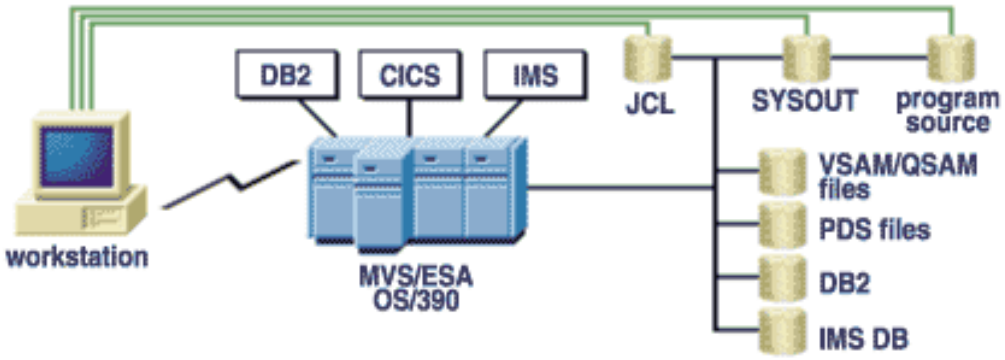
Appendix. Notices 31

© Copyright IBM Corp. 1995, 2000

Chapter 1. Remote edit-compile-debug

In remote edit-compile-debug, you work in a cooperative development environment without downloading files to the workstation.

With remote edit-compile-debug, you can develop or enhance many types of applications, including CICS, batch, TSO, or IMS Database Manager and IMS Transaction Manager applications. These applications can access many forms of data, including DB2, VSAM, DL/I, and QSAM data.



Which tasks to do where	Workstation	Host
Edit programs on the workstation. The program files are transparently retrieved from and saved to the host. These files include the source, JCL, and held job `SYSOUT`. The source remains on the host.	X	
Compile the programs. Compilations to create load modules are done on the host, but you initiate them from the workstation by submitting jobs to the host. You can do a syntax check on the workstation with the source on the host; copy files are retrieved from the host. Neither DB2 nor CICS is required on the workstation for the syntax checking.	X	X
Test and debug the compiled programs. Execution takes place on the host. Batch execution is initiated from and controlled from the workstation (CICS transactions and IMS transactions must be started from a 3270 emulator session). The debugger runs cooperatively with a GUI on the workstation and the engine on the host.		X

"**X**" indicates the primary platform for the task.

You start certain actions (such as compiling and debugging) on your workstation, and the activity is done on the host. You can also bring your source down to the workstation, edit it locally on the workstation, and compile, test, and debug on the host.

You can divide these tasks other ways between the host and the workstation. For example, on the workstation you can edit your programs and do a syntax check

© Copyright IBM Corp. 1995, 2000

and then compile remotely (initiating the host compile from the workstation) for debugging on the host. In this case, the CICS and DB2 products are not needed on the workstation for processing `EXEC CICS` and `EXEC SQL` statements, respectively.

Benefits of remote edit-compile-debug

You can save time by editing, compiling, and debugging your host applications remotely:

- Edit faster with the language-sensitive editor. It parses the source as you enter it, giving a visual presentation.
- Save time debugging. You can see compile-time error messages in your edit session after the line on which the error occurred.
- Reduce the number of compile jobs. Use the **compile** action to check syntax on the workstation, eliminating syntax errors before compiling with the host compiler.
- Reduce the number of batch debugging jobs and ease your debugging tasks. Use the graphical debugger on the workstation while your program runs on the host.
- Use your existing jobs for system test with the benefit of workstation organization.

In addition, you can use remote edit-compile-debug no matter what kind of data your application accesses. You can use it even for applications that include calls to programs in other languages and calls from assembler programs.

Costs of remote edit-compile-debug

You will need the help of host system programmers to configure communications between the host and the workstation. Network traffic will increase, and you need to define projects on the workstation to reflect the organization of the host application.

RELATED CONCEPTS
"Remote editing"
"Remote compiling" on page 3
"Remote debugging" on page 3

RELATED TASKS
Editing host files ("Editing host data sets" on page 21)
"Compiling host applications" on page 23
"Debugging host applications" on page 26

Remote editing

Remote editing consists of using the PL/I editor to edit code that resides on the host. This editor has these features that will help you shorten the development cycle:

- PL/I-sensitive editing. The editor shows PL/I constructs (called tokens in the editor), such as commands and comments, in different colors.
- Customizations to meet individual needs. For example, you can change the colors for PL/I constructs or change the functions associated with different keys. You can make the editor resemble a host editor that you are familiar with.

- Viewing of more than one file (data set) at a time. You can set up the files in a ring so that you can move easily among them. Two files, or two parts of the same file, can also occupy a split screen or separate screens.
- Powerful macro facility. Macros, written as REXX procedures and PL/I editor commands, let you create your own commands to provide additional editing functions.

This editor shortens the process of fixing the code, because you can detect many errors during editing rather than having to wait until compiling.

RELATED TASKS
Editing host files ("Editing host data sets" on page 21)
"Emulating other editors" on page 23

Remote compiling

The remote compile process is the same as the compile process that you use when developing on the host. However, for remote compiling you use the integrated graphical tools to start and monitor the compiles, and the source remains on the host. So you control the compilation from the workstation.

You also have the benefit of syntax checking without creating an object program. You can find out the errors that the compiler finds from checking syntax and correct them on the workstation before you compile for production.

RELATED TASKS
"Checking syntax on the workstation" on page 24
Submitting jobs on MVS ("Submitting jobs on the host" on page 24)
"Monitoring the status of submitted jobs" on page 25

Remote debugging

Remote debugging consists of using the remote debugger to debug code that runs in the OS/390 environments. The supported environments are batch (JES), TSO, CICS, and IMS (both IMS Database Manager and IMS Transaction Manager), and DB2 (including stored procedures). The debugging sessions are cooperative. The Distributed Debugger resides on the workstation, and Debug Tool runs on the host with your application. The workstation interface and Debug Tool communicate with each other through TCP/IP.

This debugger has these features that will help you shorten the development cycle:

- Language-sensitive debugging. You can debug multilanguage programs (such as PL/I and COBOL) and see all of the source code during the debug session.
- All actions are done by pointing and clicking. You do not need to learn long command strings.
- You can debug IMS, CICS, and batch programs interactively from the workstation, while viewing the input and output screens on the host and Distributed Debugger windows on the workstation.

RELATED TASKS
"Debugging host applications" on page 26

Comparing workstation and host concepts

If workstation concepts are new to you, but you are familiar with the host environment, the following table might help you. It provides a comparison of workstation and host concepts. In most cases, the terms are only roughly equivalent.

Workstation concept	Host concept
File: A unit of stored or processed information such as text, data, and programs.	**Member**: In OS/390 and MVS, a data set is the major unit of data storage and retrieval. A member is a partition of a partitioned data set.
Batch file (.BAT) and command file (.CMD): a file containing operating system commands organized for sequential processing. In workstation programs, command files are much like JCL. A **command file** can also function similarly to a **CLIST** (in OS/390 or MVS) or an **EXEC** (in VM).	**JCL**: Stands for job control language, which is used to identify a job to an operating system and to describe the job's requirements. A CLIST is a list of TSO commands and statements designed to perform a specific function for the user, and an EXEC is a user-written command file that contains REXX commands and execution of control statements, such as branches.
Dynamic link library (DLL) file: This file contains executable code and data that is bound to a program at load or run time, rather than during linking. The code and data in a DLL can be shared by several applications at the same time.	DLL files are much like **preloaded subprograms** in the link pack area on MVS and OS/390 or in a shared segment on VM. For PL/I, DLL files are most like dynamically called subprograms.
Environment variables: These are any number of variables that describe the way an operating system is going to run and the devices it is going to recognize. For example, in a PL/I project on the workstation, an environment variable is an operating system variable like PATH and DPATH, and other environment variables that are defined such as TMP.	**SYS1.PARMLIB** and a **STEPLIB or PARMLIB statement in JCL** perform functions similar to workstation environment variables, in that they provide global values or settings, and provide search information.
Executable (.EXE) file: This is a file that contains the executable code for a program.	**Load module, statically linked program, statically linked load module**: All or part of a computer program in a form suitable for loading into main storage for execution. A load module is usually the output of a linkage editor.
Module definition (.DEF) file: Directives to the linkage editor on how to build the executable file.	**Linkage editor input statements**: Directives to the linkage editor, such as INCLUDE and NAME, on how to build the load module.
Make: An action that invokes the make utility, a tool that automates the process of updating project files. This process includes compiling and linking programs.	Any automated way of controlling compiling and link-editing.
Workstation desktop: A graphical way of accessing your tools and files. It fills the entire screen and holds the objects with which you can interact to perform operations on the operating system.	No exact equivalent. The closest equivalent on the mainframe is menu-driven tools such as ISPF or OfficeVision.

Chapter 2. Configuring your systems

Before you can start remote edit-compile-debug of host applications from your workstation, you need to do these tasks:

- Make sure you have the appropriate software installed on the host and on your workstation. Be sure that you have installed the MS Loopback Adapter (using the Adapters page of the network properties on Windows NT).
- Have a system programmer or network administrator configure the communication software (TCP/IP, IBM HTTP Server, foreign file system server, and job monitor server) for your development environment. The remote commands server is required only if you plan to issue TSO commands from the workstation.
- Set up remote edit-compile-debug for the workstation by running the MVS connections manager.

The remote edit-compile-debug component of VisualAge PL/I communicates between the workstation and the host through the HTTP protocol. The IBM HTTP Server (formerly Domino Go Webserver) runs under UNIX System Services (formerly Open Edition) in OS/390. The server must be installed under the hierarchical file system (HFS) of UNIX System Services and configured for the remote capability of VisualAge PL/I. This server communicates with the foreign file system server through the GoWebserver API (GWAPI).

The person installing and configuring the server is usually an OS/390 system programmer familiar with OS/390 security features and with UNIX System Services. The person installing and configuring VisualAge PL/I on the workstation (the PL/I administrator) need not be an OS/390 system programmer but does need to coordinate with the OS/390 system programmer. For the convenience of the system programmer, the instructions for installing and configuring the parts that reside on OS/390 are in a program directory that comes with the product.

RELATED CONCEPTS
"Chapter 1. Remote edit-compile-debug" on page 1

RELATED TASKS
"Installing and configuring the MS Loopback Adapter"
"Configuring and connecting your workstation for remote ECD" on page 6

Installing and configuring the MS Loopback Adapter

You must install and configure the MS Loopback Adapter to be able to connect to an MVS system. You can install the MS Loopback Adapter after you install VisualAge PL/I, but it must be installed before you start the MVS connections manager for the first time.

To install the MS Loopback Adapter, follow these steps:

1. Right-click the **Network Neighborhood** icon on your Windows NT desktop. Click **Properties** and select the **Adapters** tab.
2. On the Adapters page, click the **Add** button.

© Copyright IBM Corp. 1995, 2000

3. In the Select Network Adapter window, select MS Loopback Adapter from the Network Adapter list. Click **OK**.
4. In the MS Loopback Adapter Card Setup window, click **OK** to accept the frame type.
5. In the Windows NT Setup window, specify the location of the installation files for the adapter. If the installation files are on your workstation, the directory where they reside is indicated. If not, the directory where they reside on the Windows NT CD-ROM is indicated, and you will need to insert the CD-ROM.
6. Click **Continue**. The MS Loopback Adapter will be added to the Network Adapter list. Click **Close**.

To configure the MS Loopback Adapter, do these steps:

1. In the Microsoft TCP/IP Properties window, select the MS Loopback Adapter from the Adapter list.
2. Click the button for specifying an IP address and enter the IP address, 192.168.2.1. Do not click the button to obtain an IP address from a DCHP server.
3. Move the cursor to the **Subnet Mask** field. This field should contain 255.255.255.0. Leave the **Default Gateway** blank. Click **OK**.
4. Reboot your workstation.

To verify that the MS Loopback Adapter was installed and configured, do these steps:

1. Right-click **Network Neighborhood**, then click **Properties**.
2. Click the **Services** tab. On the Services page of the Network window, click **NetBIOS Interface**, then click **Properties**.
3. In the NetBIOS Configuration window, you will see that a Lana number with the value NetBT->NDISLoop has been added.

In some cases, you can increase the speed of bringing up certain tasks by placing the Loopback Adapter last in the protocol bindings. To do this:

1. Right-click **Network Neighborhood**, and then click **Properties**.
2. Click the **Bindings** tab. On the Bindings page in the Network window, select **all protocols** in the box next to Show Bindings for.
3. Expand each protocol option by clicking the **+** sign.
4. For each protocol where MS Loopback Adapter appears, click **MS Loopback Adapter** and then click **Move Down** until each occurrence of MS Loopback Adapter is last in the list of bindings for each protocol.

Configuring and connecting your workstation for remote ECD

Before configuring the connection to an MVS system, be sure of the following:

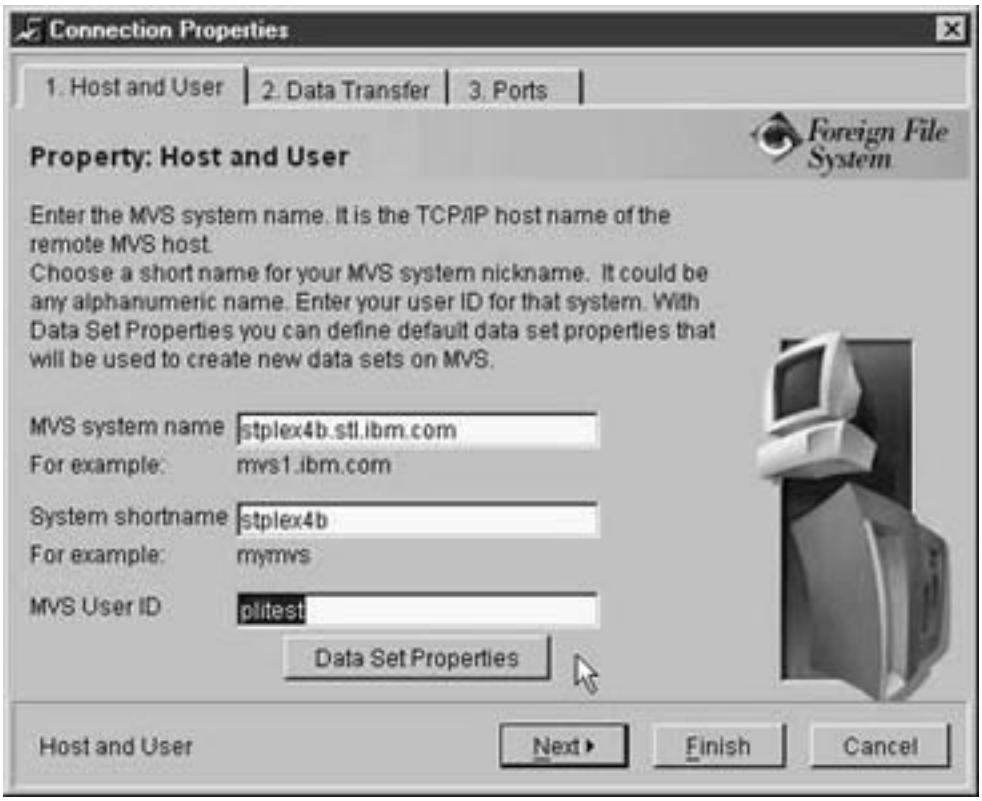
- You have installed the MS Loopback Adapter on your workstation. You will not be able to connect to the MVS system unless you have installed this adapter.
- The system programmer has configured TCP/IP, IBM HTTP Server, foreign file system server, job monitor server, and (if you plan to issue TSO commands from the workstation) remote commands server.
- The system programmer has told you the port numbers for the IBM HTTP Server (default 80) and job monitor server (default 6715) if these numbers were changed during the configuring.
- You have your user ID and password for the MVS system where your source code resides.

While you are configuring your workstation for remote ECD, you will select drive letters that VisualAge PL/I will use to attach your host files. You can specify more than one drive and more than one folder (directory) on a drive for a project. For example, you might use one drive for your source files and another drive for your test files. In a source folder, you might have different folders for your JCL and your PL/I code.

To configure the connection to an MVS system, do the following steps:

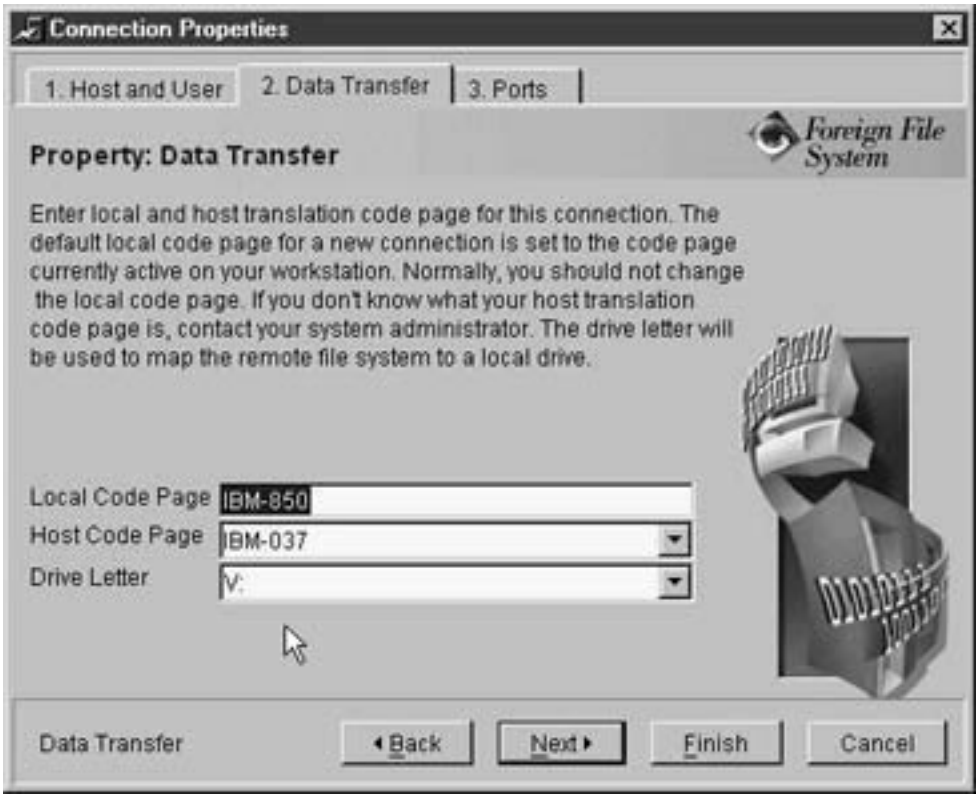
1. Click **Start** -> **Programs** -> **IBM VisualAge PLI** -> **Tools - Remote Access -> MVS Connections**.
2. On the Host and User page, enter the TCP/IP name of your MVS system. After you enter the system name and move to the next box, you will see that a system short name, a nickname for your system, is provided for you. You can either accept the default short name or specify your own. The short name can be any combination of letters and numbers. Enter your user ID for the MVS system.

   If you want to define default data set properties for new data sets that you create on MVS, click **Data Set Properties** and specify the information requested in each field. Otherwise, click **Next**.



3. On the Data Transfer page, check whether the default code pages are correct. Typically, you should not need to change the default local code page. The host code page must match the code page used on your MVS system.

   You can accept the drive selected or choose a drive letter to represent the virtual directory on your Windows workstation that VA PL/I creates for linking to the host. Click **Next**.



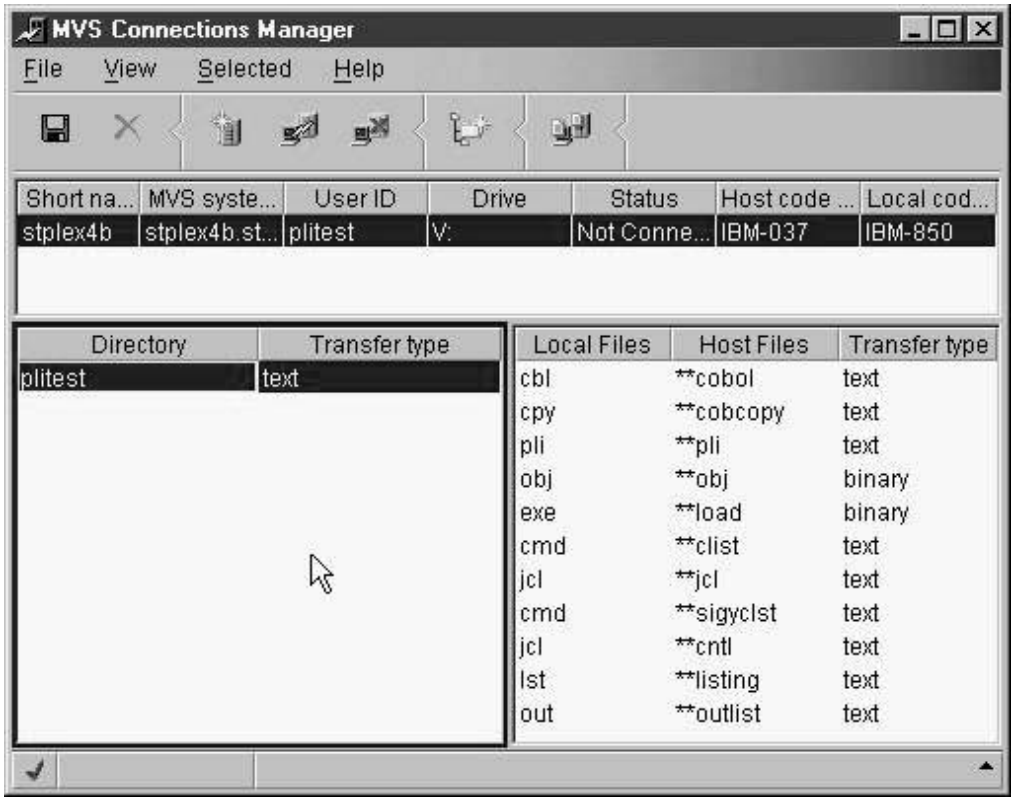
4. On the Ports page, check whether the port numbers of the IBM HTTP Server (Web port) and job monitor server (job port) are correct. Normally, you would not change the defaults. Click **Finish**.
5. In the **Directory** box in the Directory Properties window, enter the name (probably the high-level qualifier) of the host directory path. A default directory path, generated from the MVS user ID you specified earlier, is provided for you. You can either accept this default directory path, or specify your own. Do not use a PDS name for the path. For example, if you have a data set called userid.MYSRC.PLI, you can specify a path of either userid or userid.MYSRC, but do not specify a path of userid.MYSRC.PLI. Doing so will result in unexpected behavior.

   Select the type of data set (either text or binary) from the **Transfer** menu.

   If you want to to define default data set properties for new data sets that you create on MVS, click **Data Set Properties**. Otherwise, click **Finish**.



You have completed configuring one connection to your MVS system! In the MVS Connections Manager window, the information for your MVS system connection should be displayed.



6. Check the information displayed and make sure you have entered everything correctly. If you find a mistake, double-click the information you want to change to open the Connection Properties window and make changes there. You can add a directory to this drive by right-clicking on the drive and then clicking **Add Directory** and filling in the information.

   The mappings of local and host files in the lower right pane determine the file extensions that PDS members will have on the workstation. To change these, right-click an item and then click **Properties**. Or you can right-click a drive in the top pane and then click **Add Mapping**.

   When you have finished, click the **Save changes** icon at the far left of the tool bar to save the configuration of your MVS connection.
7. You are now ready to establish a connection. Select the system connection that you want to connect, and then click the **Connect All** icon. The status field should change from Not Connected to Connected.

RELATED TASKS
"Installing and configuring the MS Loopback Adapter" on page 5
"Mapping MVS data sets" on page 20
"Verifying your connection"
"Creating an MVS project" on page 13

Verifying your connection

To verify that your communications configuration is working, do the following steps:

1. In the MVS Connections Manager window, click the **Connect all** button on the toolbar. If the MVS host is available, you will be prompted for a password.
2. Enter your password for the MVS host and the user ID that you specified in the MVS connections manager. If the connection was successful, the Status column in the MVS Connections Manager window changes to Connected.

If the connection was unsuccessful, you might see the following message:

```
The host mymvs.ibm.com is not responding.

Please check that the MVS system name is valid and the host is accessible to
TPC/IP.
```

This message means that your host TCP/IP name is not valid. Check that you have entered the correct name.

You might see this message:

```
The following drives failed to connect: driveletter1, . . .
```

To find out why the drive failed to connect, you can check parts of the connection as described below. If you find that something is wrong with the host connection and you have narrowed the cause to a part on the host, contact your MVS system programmer for help.

Checking the TCP/IP connection

1. In an MS DOS command window, enter this command to ping your MVS host by name:

   ```
   C:\>ping mymvs.ibm.com
   ```
2. Check the messages that you receive from the ping command for any errors. If ping returns an error about an unknown host and the name is correct, your

TCP/IP configuration on the workstation is incorrect. If ping returns data from the host, your TCP/IP connection is operating normally.

Checking the IBM HTTP Server connection

1. In the MS DOS window, enter:

   ```
   C:\>net use V: \\ffssash\mvs1
   ```

   Here *V:* is the drive letter that you specified in the Connections properties setup, and *mvs1* is the short name that you specified for your MVS host.
2. Check the messages that you receive from the `net use` command for any errors. If the command fails, the problem could be that the MS Loopback Adapter is not installed and configured correctly. Go through the installation and configuration steps for the Loopback Adapter to make sure it is working correctly. If the connection still does not work, the IBM HTTP Server might not be started, or the foreign file system server might not be installed and running on your host system.
3. If the connection to the host was successful, disconnect from the host by using the **Disconnect all** button or by using the net use delete command:

   ```
   C:\>net use V: /d
   ```

Checking whether the IBM HTTP Server has started

1. In a Web browser, enter the following Web address:

   ```
   http://mvsipaddr:port/
   ```

   Here *mvsipaddr* is the TCP/IP name of your MVS system, and *port* is the port number of your Web server.
2. If prompted, fill in your TSO user ID and password.
3. If the server has started, you will see the Web page for the IBM HTTP Server.

Checking whether the foreign file system server is installed

1. In a Web browser, enter the following Web address:

   ```
   http://mvsipaddr:port/FFSDS
   ```

   Here *mvsipaddr* is the TCP/IP name of your MVS system, and *port* is the port number of your Web server. FFSDS must be uppercase.
2. If the server has been installed, you will see the Web page for the VisualAge foreign file system server.

RELATED TASKS

"Installing and configuring the MS Loopback Adapter" on page 5
"Configuring and connecting your workstation for remote ECD" on page 6

Chapter 3. Developing remotely

When you have finished setting up, you will be able to develop remotely by doing these tasks:

- Create an MVS project.
- Access host data sets.
- Edit host files.
- Compile on the workstation to check syntax.
- Submit batch jobs for preprocessing, compiling, and link-editing.
- Monitor the status of jobs on the host.
- Debug programs running on the host.
- Execute TSO commands on the host.

RELATED TASKS
"Creating an MVS project"
"Accessing host data sets" on page 18
Editing host files ("Editing host data sets" on page 21)
"Checking syntax on the workstation" on page 24
"Submitting jobs on the host" on page 24
"Monitoring the status of submitted jobs" on page 25
"Debugging host applications" on page 26
"Issuing host commands from the workstation" on page 29

Creating an MVS project

Before creating an MVS project, be sure of the following:

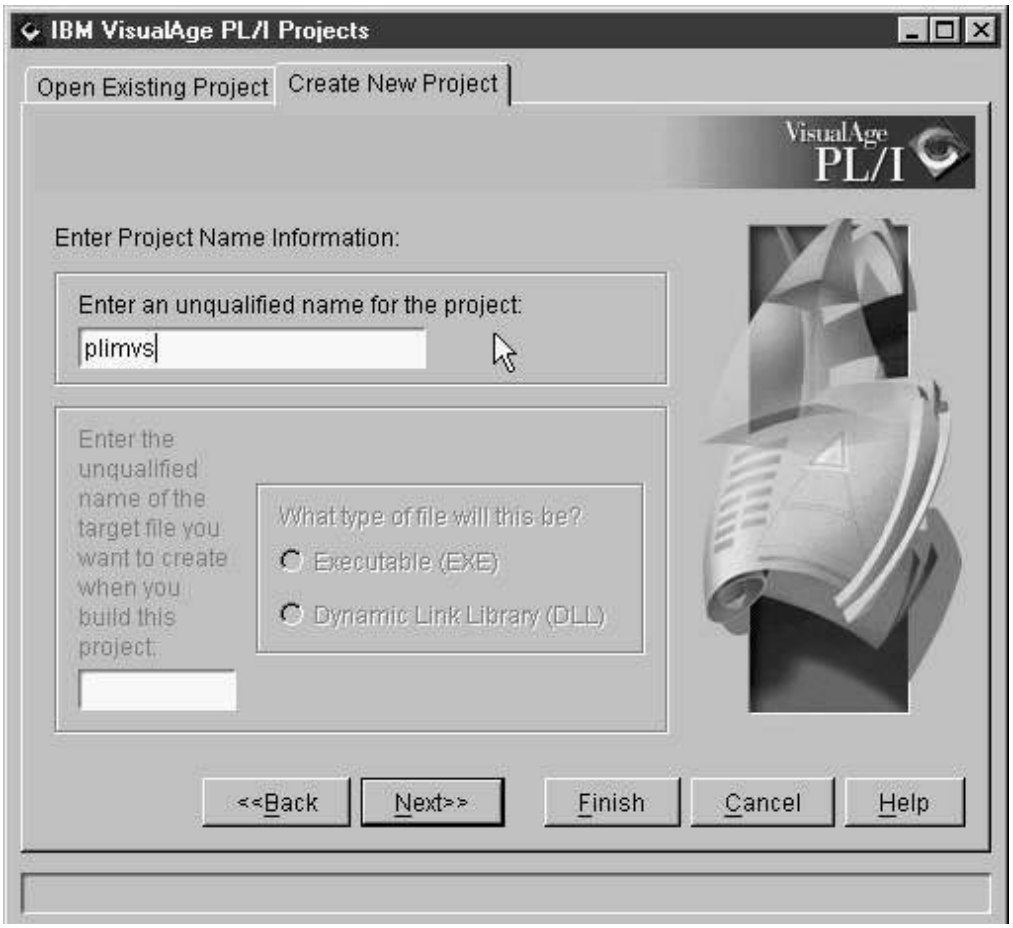
- You have configured the connection to an MVS system, and you are connected to that system.
- You know what drives you assigned to your MVS data sets when you configured your connections.
- You know the names of the data sets that you will use in your MVS project.

To create an MVS project, do the following steps:

1. Click **Start** -> **Programs** -> **IBM VisualAge PLI** -> **Accessing PLI Projects**.
2. In the PL/I Projects window, click the **Create New Project** tab.
3. Click **MVS Project**. Click **Next**. On the Create New Project page, you will enter information to identify your project and where it will reside on the workstation.

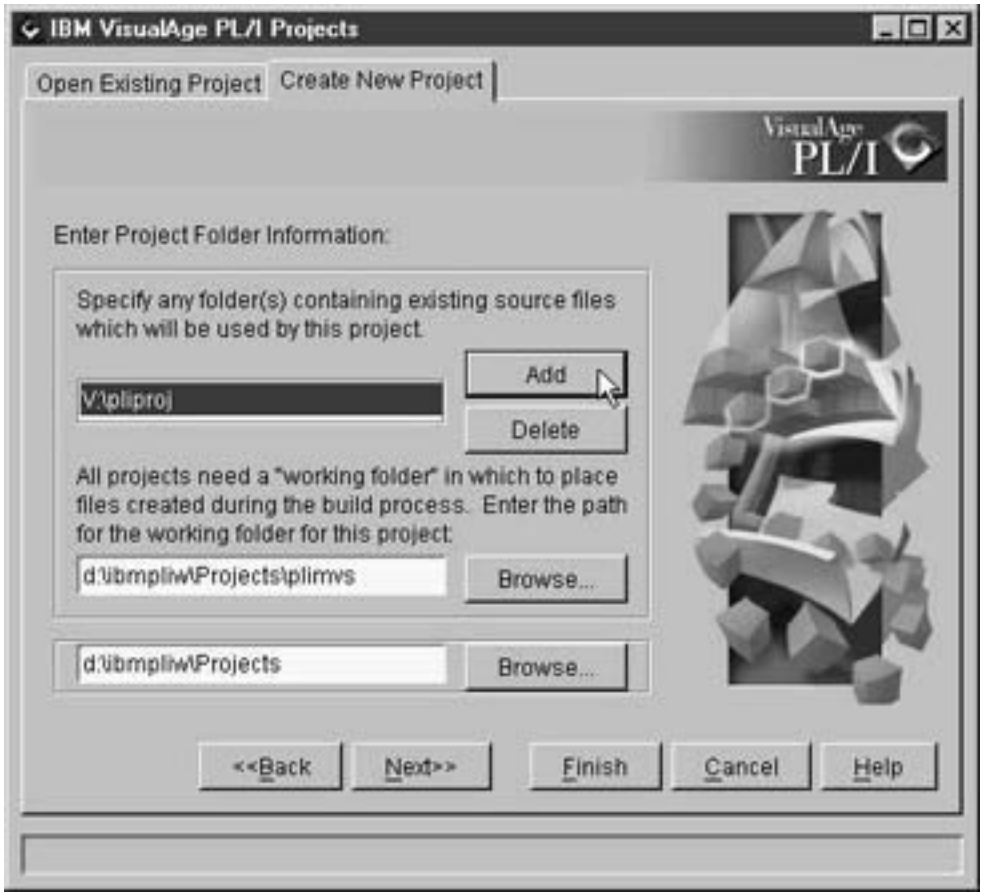
© Copyright IBM Corp. 1995, 2000

4. Enter an unqualified name for the project (such as `plimvs1`). Click **Next**.



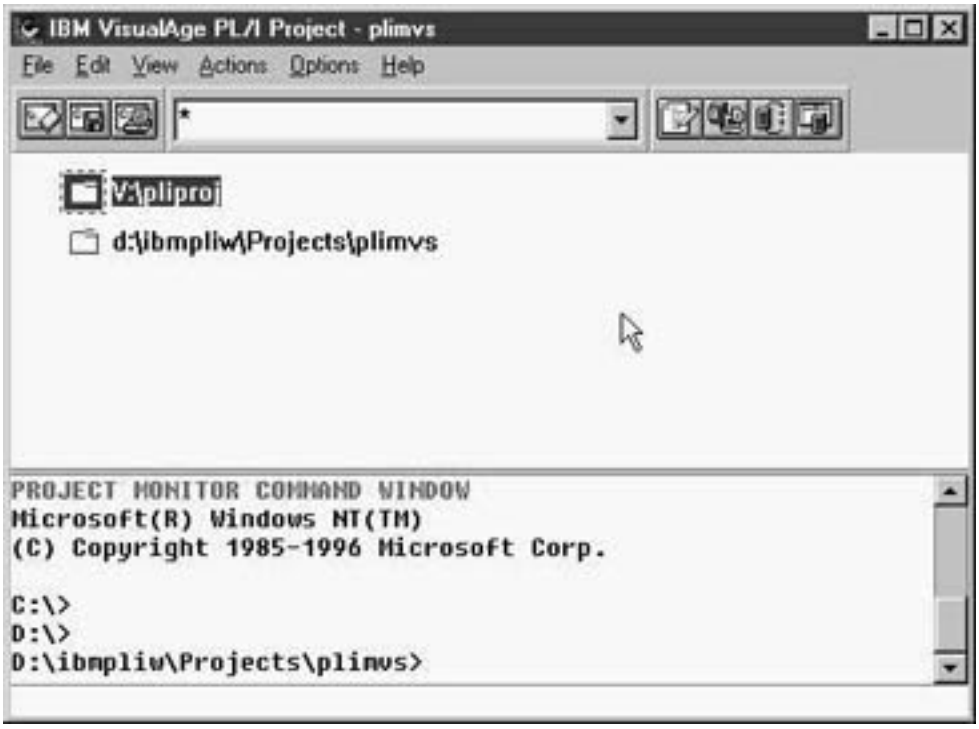
5. Enter Project Folder information. Click **Add** to specify a folder for your source files, which already exist as data sets on MVS. Choose the drive and folder that you designated for the host files when you created the MVS connection. Filling in the drive and folders for your project here lets the PL/I project environment know about your project. To add more than one folder, press the Ctrl key while you select the folders that you want to add.

   Accept the default names for the working folder that will hold your generated files and for the folder that will hold your project definition files. This information is generated from the information that you specified when you created your MVS connection. Click **Next**.



You are prompted to have the folder created. Click **Yes**.

6. Specify environment variables, or you can accept the default one. Click **Finish**. A message confirms the creation of the project, and the IBM VisualAge PL/I Project - plimvs1 window opens. You have set up the project environment for remote editing, compiling, and debugging!



RELATED TASKS

"Configuring and connecting your workstation for remote ECD" on page 6

"Changing the source directories for your MVS project"

Changing the source directories for your MVS project

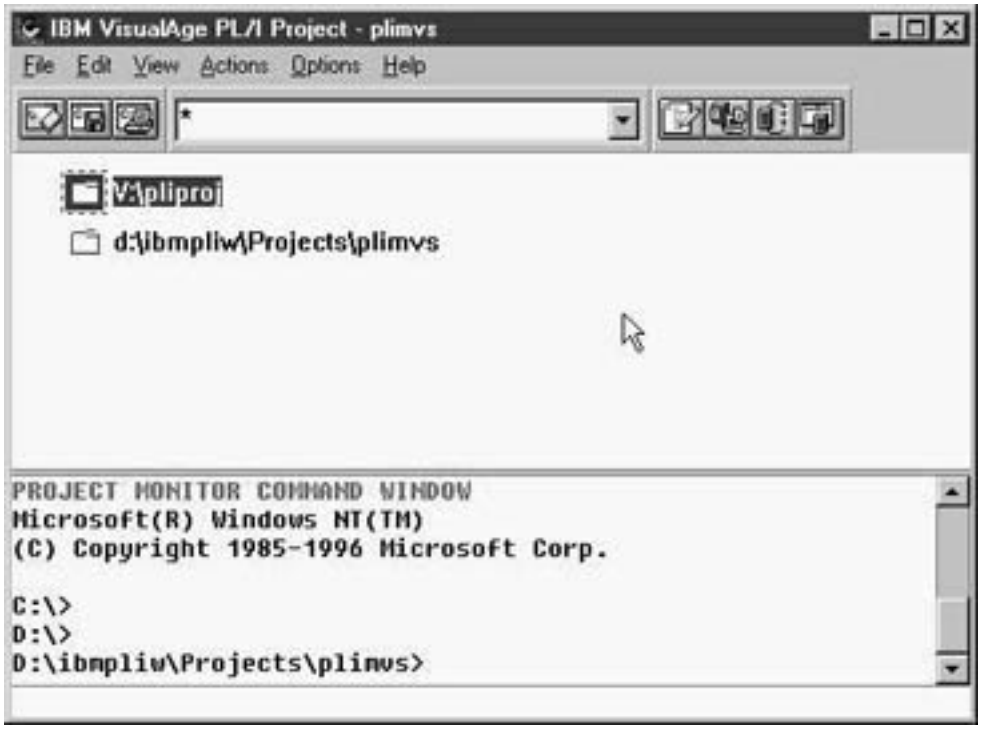
Before you change the source directories for your MVS project, be sure of the following:

- You have configured the connection to an MVS system and you are connected to that system.
- You have created an MVS project.

To change the information for the source directories for your MVS project, do the following steps:

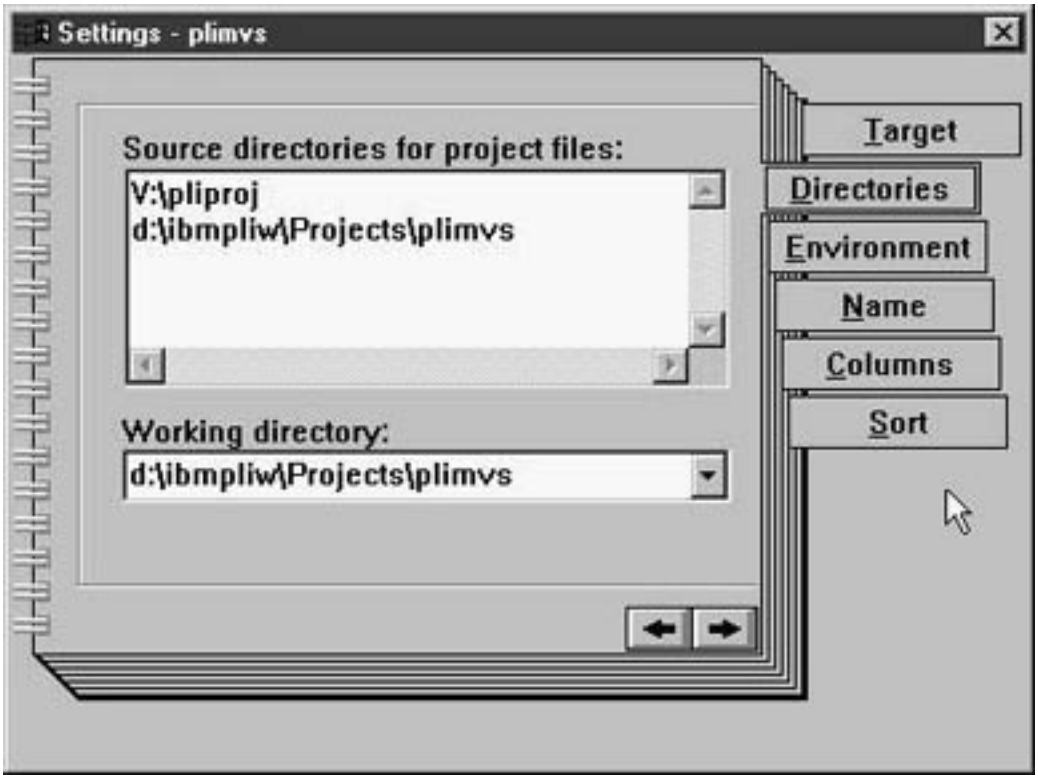
1. Click **Start** -> **Programs** -> **IBM VisualAge PLI** -> **Accessing PLI Projects**.
2. In the PL/I Projects window, click the **Open Existing Project** tab.

3. Click the project you want to work with and then click **Open**. The project you selected opens in an IBM VisualAge PL/I Project window.



4. In the Project window, click **File -> Project Settings -> Directories**.

   In **Source directories for project files**, the current host directory path that you designated for the project source files is displayed. In **Working directory**, the directory path that you designated for the working folder to hold your generated files and your project definition files is displayed. This information is generated from the information that you specified when you created the MVS project.



In this example, `V:\` is the same drive as the drive chosen when you created the MVS connection. `pliproj` is the high-level qualifier.

5. Add, modify, or delete directory information as necessary. You can enter more than one source directory, but be sure to enter them on separate lines. The working directory must be a directory on your workstation that is not attached to the host. Click **OK** when you have finished.

   If this is a new directory, you will be asked if you want to create the specified directory. Click **Yes** to accept it. In the Project window, the directories you just created are now displayed.

RELATED TASKS

"Configuring and connecting your workstation for remote ECD" on page 6
"Creating an MVS project" on page 13

Accessing host data sets

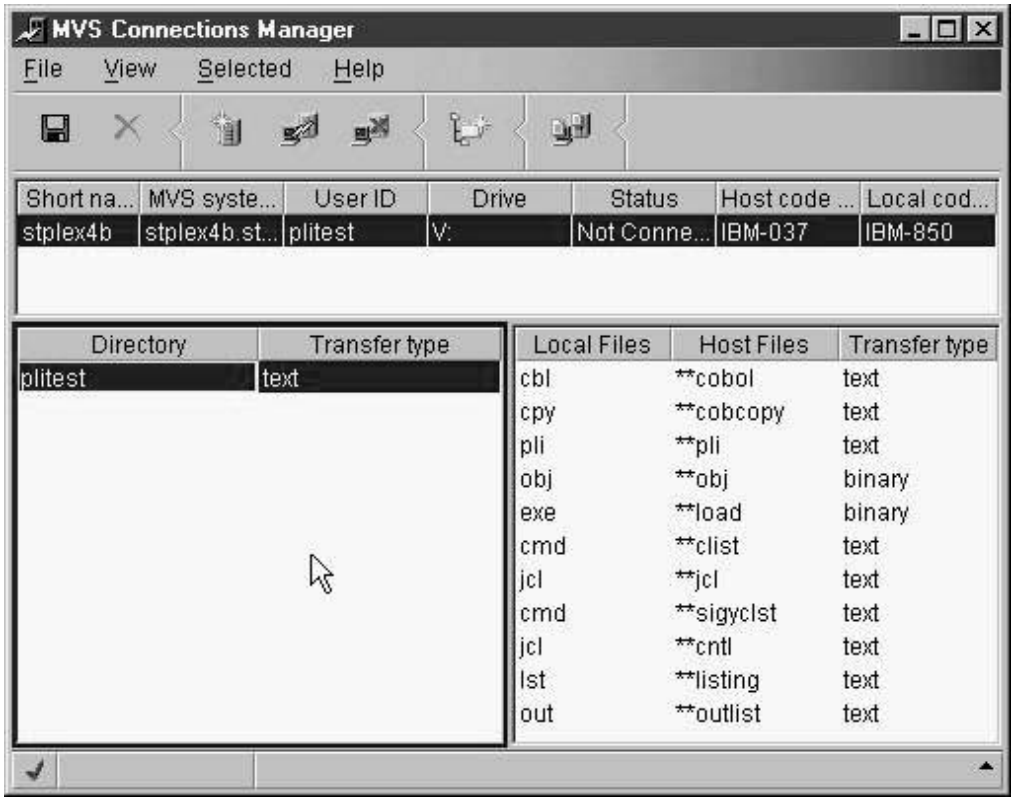
Application parts (such as PL/I source code, PL/I include files, and host JCL) are kept in partitioned data sets (PDSs) or partitioned data sets extended (PDSE) on the host. You access and work with these files through an MVS project. If you keep your source, include files, or JCL in a library system on the host, you must copy them from the host library to a PDS to make them available for MVS project actions.

Before you can edit, compile, or debug remotely, make sure that your workstation is configured for communicating with the host. You also need to define PDS names for the project and connect to the host. The names that you see on the workstation for your host data set members depend on how you defined your network drives while setting up for remote edit-compile-debug.

After you complete the required configuration, you can connect to PDS or PDSE data sets and include members in your project. You can then access your host files as if they were workstation files.

You define PDS names by using the MVS connections manager. You can start this manager from the VisualAge PL/I folder (click **Start -> Programs -> IBM VisualAge PLI -> Tools - Remote Access -> MVS Connections**). Or you can access this manager when you create your first MVS project (if no PDS names are already defined). When you create an MVS project, you select the PDS names that are appropriate for the project.

You can connect to the host in the MVS Connections Manager window by selecting the drives that you want to connect and clicking the **Connect All** icon, or you can click **Actions -> Connect MVS drives** in an MVS project.



If your workstation is already connected, the host drives are disconnected and then reconnected. Depending on your setup, you might be prompted for your TSO user ID and password. This prompting can occur for each drive that you access. By reconnecting you can correct file system problems such as a connect timeout.

You can access host data sets that are cataloged in user catalogs and connected to the master catalog by using an alias entry for the high-level qualifier.

In the following cases, data sets might not display in your MVS project:

- A PDS or PDS member is currently allocated by a TSO session or a batch job with a disposition of OLD, MOD, or NEW. Members of a PDS that have been allocated in this way will not display.
- A data set has been migrated.

If you have trouble accessing a host file through the file system or you receive an "Access is denied" error message, check the host system log to see whether the access resulted in RACF violations. Before your host projects can access host data sets that are cataloged directly in the VSAM master catalog, you must have access in one of these ways:

- RACF generic entry for the high-level qualifier that allows you read access
- At least read access as the default RACF universal access code

Typically, system data sets (such as SYS1) are cataloged directly into the master catalog. Attempts to access them can cause RACF violations to be logged.

RELATED CONCEPTS
"Remote editing" on page 2

RELATED TASKS
"Configuring and connecting your workstation for remote ECD" on page 6
"Mapping MVS data sets"
"Changing the source directories for your MVS project" on page 16
Editing host files ("Editing host data sets" on page 21)

Mapping MVS data sets

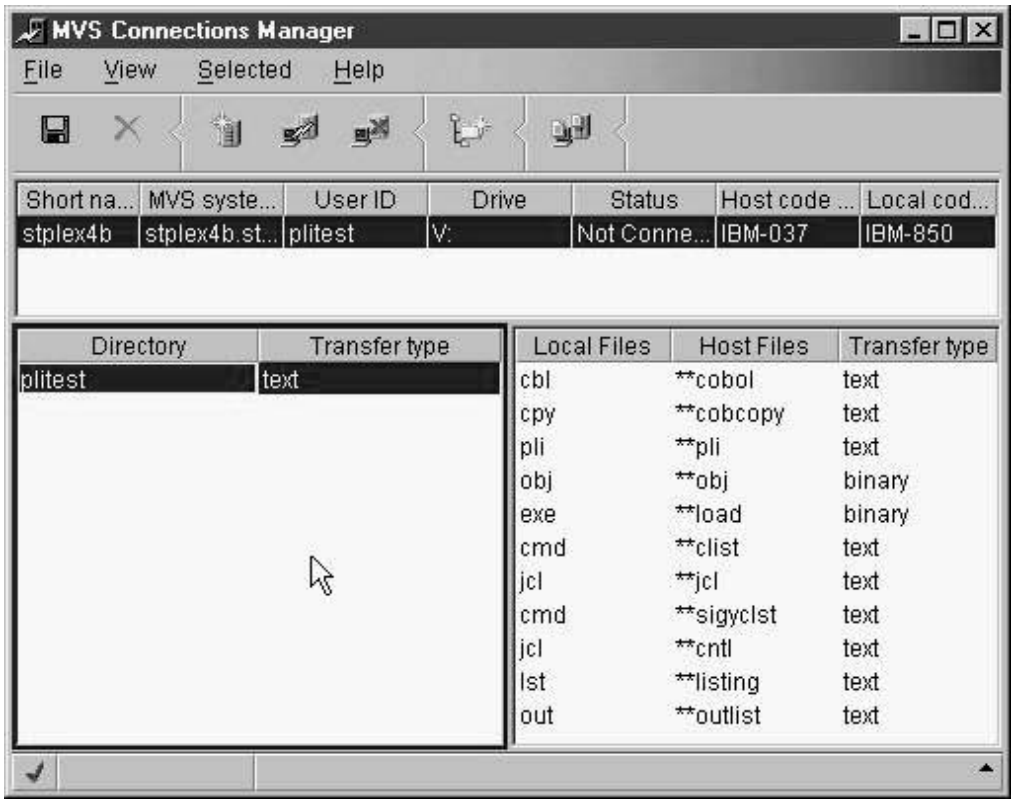
When you define an MVS system by using the MVS connections manager of VisualAge PL/I, you can also define how the names of your MVS data sets map to file extensions on the workstation. The MVS connections manager provides the following defaults, which you can change:

File extension on workstation	PDS name on MVS	Description
pli	**pli	PL/I source (input to the PL/I compiler)
cmd	**clist or **sigyclst	CLIST or REXX procedure
cpy	**include	PL/I include file (used by the PL/I INCLUDE statement)
exe	**load	Load module (output of the linkage editor)
jcl	**jcl or **cntl	Job control language (JCL) on the host used for submitting batch jobs
lst	**listing	Compilation or link-edit listing
obj	**obj	Object code (input to the linkage editor)
out	**outlist	Host job SYSOUT output

For example, by default, if a PDS name ends in PLI, its members show up on the workstation with the file extension .pli. Therefore, you see AAA.BBB.PLI(PGM1) as pgm1.pli on the workstation.

Similarly, when you create or copy a new member in a PDS by using workstation tools, the file extension of the file that will become the new member must conform to the mapping that is already set. For example, you can save a new file as pgm2.pli into AAA.BBB.PLI because .pli is a valid extension for PDSs that end in PLI. However, if you try to save the new file as pgm2.xxx, you will get an error. If you are using the VisualAge editor, you will see a message that the file name is not valid.

To change the mappings in the MVS Connections Manager window, right-click an item in the lower right pane and then click **Properties**. Or you can right-click a drive in the top pane and then click **Add Mapping**.



Appropriate file extensions are useful for these reasons:

- You can use tools and execs that expect workstation file names to have extensions.
- You can easily distinguish the type of file by its extension, and so you do not have to read the fully qualified data set name.
- Within a project, the file extensions ensure that appropriate actions are available for different types of files.
- You are protected against inadvertent actions on the wrong data set when manipulating host files.

RELATED TASKS

"Configuring and connecting your workstation for remote ECD" on page 6
"Changing the source directories for your MVS project" on page 16

Editing host data sets

PDS members that appear in your project still reside on the host. However, you can interact with them as if they resided on your workstation.

To edit a member:

1. Click the file to select it.
2. Click **Actions -> MVS Edit** from the menu bar.

The **MVS Edit** action provides safeguards to ensure your file is handled properly. For example, your file will be locked so that other users (ISPF or VisualAge PL/I) cannot edit it at the same time.

If you click **Actions -> Delete**, you delete the member on the host. Being connected to the host makes your host data appear as another drive (or drives) on your workstation.

When you edit host files on the workstation, consider the supported data sets and record formats:

- Data set organizations of sequential (PS) or partitioned (PO, either a PDS or PDSE) are supported.
- Only record formats (RECFM) of fixed (F) or variable (V) are supported. The records can be blocked or unblocked (FB or VB) and can have printer carriage control characters (FBA).

If you encounter a problem saving your file to the host, use **File -> Save As** to save the file on your workstation. After the problem has been corrected, you can upload the saved file to the host.

Files with sequence numbers

OS/390 or MVS files that have fixed-length records with a logical record length of 80 can use columns 73 through 80 for sequence numbers. When you start **MVS edit**, it checks whether your file contains numeric data in columns 73 through 80 and whether the numeric data is in ascending order throughout the file. If so, the editor automatically maintains the sequence numbers for you. The sequence numbers do not display in your edit session unless you use a key behavior (such as ISPF) that provides a prefix area.

You can use columns 73 though 80 for data other than sequence numbers. For example, you can use them to identify maintenance changes. You can request that the editor handle columns 73 though 80 separately from the rest of the file by selecting **Options -> Cols 73-80 separate**. With this option, when you use functions like inserting or deleting characters or splitting lines in columns 1 through 72, the data in columns 73 through 80 is not affected. Otherwise, the data in 73 through 80 shift to the left.

When **Cols 73-80 separate** is active, you can also choose the **Propagate cols 73-80** option. With this option in effect, each new line you add has the same characters in columns 73 through 80 as the previous line. This option is convenient when you are adding new code and want all the lines to have the same identifier in columns 73 through 80.

RELATED CONCEPTS
"Remote editing" on page 2

RELATED TASKS
"Emulating other editors" on page 23

RELATED REFERENCES
MVS Edit safeguards"MVS edit safeguards" on page 23

MVS edit safeguards

Host files, unlike workstation files, have a logical record length (LRECL) that must be maintained. The LRECL is enforced. For variable record files, the usable record length is 4 less than the LRECL. **MVS edit** recognizes the LRECL for a file and prevents you from typing lines that are too long.

Host files that contain certain characters can cause problems when edited on the workstation:

- Hex code less than X'40'
- Unprintable hex characters with a hex code greater than X'40'

MVS edit detects nondisplayable hex characters in your file and presents a displayable version of the file in a read-only edit session.

RELATED TASKS
Editing host files ("Editing host data sets" on page 21)

Emulating other editors

You can tailor the PL/I editor to emulate some of the functions provided by other IBM editors, such as EPM, SEU, XEDIT, or ISPF (PDF).

To emulate a different editor:

1. From the PL/I editor menu bar, select **Options -> Key behavior**.
2. Choose the editor you want to emulate.
   To reset the PL/I editor to its original state, select **Options -> Key behavior -> LPEX**.

When emulating the SEU, XEDIT, or ISPF editors, the editor provides a prefix area on the left where you can type prefix commands. To run a prefix command that you have typed, press Alt+Enter.

Emulating other editors affects what the editor does when you press certain keys. For example, when emulating the ISPF editor, the editor does not insert a new line if you press Enter while the cursor is in the prefix area.

RELATED CONCEPTS
"Remote editing" on page 2

RELATED TASKS
Editing host files ("Editing host data sets" on page 21)

Compiling host applications

Before you compile and link your applications or run batch jobs on the host, you can compile your source programs on the workstation to check their syntax. This compilation on the workstation can be more efficient than compiling on the host and can reduce the workload on the host.

After ensuring that your source code is free of errors, submit batch jobs from your host project on the workstation to execute on the host and then monitor their

status. You can monitor any of various job queues and take actions such as requesting to be notified when a job has completed and browsing held job output.

RELATED CONCEPTS
"Remote compiling" on page 3

RELATED TASKS
"Checking syntax on the workstation"
"Submitting jobs on the host"
"Monitoring the status of submitted jobs" on page 25

Checking syntax on the workstation

You can use VisualAge PL/I to find and correct syntax errors in your source code that resides on the host.

1. From your host project on the workstation, click **Options -> Compile**.
2. In the Compile notebook, set the options that your program needs, and click **OK** to save them. In particular, you might want to set the following options:

Page	Compiler option	Comments
Processing	Compilation Control -> Syntax check	No object code will be produced.

3. Select the file you want to compile. From the Project menu bar, choose **Edit -> Compile**. The project monitor displays the results of the compile.
4. If the compile has errors, double-click an error message to bring up an edit session for your source code. The edit session opens positioned at the line in error. Continue correcting the compile errors.
   If your source code contains hex characters with a hex code less than X'40' or unprintable hex characters above X'40', you might get compile errors that do not occur when you compile on the host. The editor opens in read-only mode and contains information messages informing you about the hex characters.

RELATED CONCEPTS
"Remote compiling" on page 3

RELATED TASKS
"Submitting jobs on the host"
"Editing host data sets" on page 21

Submitting jobs on the host

From your workstation, you can submit jobs to run on the host.

1. From your host project on the workstation, select the JCL file that you want to submit. You can submit only one job at a time.
2. Click **Actions -> Submit job on MVS**. After a few moments, the job monitor shows messages that indicate whether your job was successfully submitted. You can select and submit another job before you get the messages for a previous job.

You can also submit batch jobs for running on the host while editing a JCL file by choosing **Submit job -> Submit job on MVS**. This action is available only for host

files that are recognized as JCL. Make sure that, when you set up remote edit-compile-debug, you mapped the JCL PDSs to the JCL workstation type.

You can also submit batch jobs from the job monitor.

From your MVS project, you can monitor the status of jobs that you have submitted to the host.

RELATED CONCEPTS
"Remote compiling" on page 3

RELATED TASKS
"Monitoring the status of submitted jobs"
"Checking syntax on the workstation" on page 24

Monitoring the status of submitted jobs

You can monitor host batch jobs that you submitted from your host project, as well as other jobs that you normally have access to through ISPF.

By checking the job status, you can do tasks such as ensuring that execution is not blocked, checking return codes, and viewing the output.

After you submit a job for compiling, do the following steps:

1. Click **Actions -> MVS Job Status**. The left pane of the Job Monitor window shows a tree view of the host systems.
2. Select the system where the job queue or job is that you want to check. When you select a system, it is connected if the connection does not already exist. In the left pane you can see the job filters for that system.
3. Click the plus sign for the job filter that you want to check. You will see the list of jobs.
4. Select the job that you want to monitor.

   To monitor more than one job at a time, select the jobs (by pressing the Ctrl key as you click the items) and select the appropriate action, such as cancel.

You can ask to be notified when a job has finished. A completion window for that job will show the status of the job.

To browse the output of a job in the held queue, double-click the job. The contents of the job output shows in the editor.

You can also release, cancel, or purge a job:

Release	Causes a job in the JES input queue to start executing.
Cancel	Cancels a running job but does not purge its output from the JES queue.
Purge	Removes all of a job's outputs from the JES queues and cancels the job if it was running.

To exit the job monitor, close the window.

RELATED CONCEPTS
"Remote compiling" on page 3

RELATED TASKS
"Submitting jobs on the host" on page 24

Debugging host applications

You can use the remote debugger to debug applications running under OS/390, CICS, DB2, or IMS (with or without BTS). The sessions are cooperative. That is, you work on your workstation with the Distributed Debugger while Debug Tool runs with your application on the host. The workstation and the host communicate with each other through TCP/IP.

You need to take the following steps to use the remote debugger:

1. Prepare the environment for debugging. ("Preparing the environment for debugging")
2. Prepare your application for debugging. ("Preparing your application for debugging" on page 27)
3. Indicate the location of the Distributed Debugger daemon. ("Indicating the location of the Distributed Debugger daemon" on page 27)
4. Start the remote debugging session. ("Starting the remote debugging session" on page 28)
5. Start your application on the host. ("Starting your application on the host" on page 29)

When Debug Tool has established a connection with the Distributed Debugger daemon, the user interface will be started on the workstation, and you can start debugging. You can set breakpoints, monitor and change variables, and follow the flow of execution of your application.

Preparing the environment for debugging

Make sure that your workstation TCP/IP communications are configured correctly. If you are using CICS, the system programmer should have configured the host so that TCP/IP will work with both Debug Tool and CICS. Instructions for configuring TCP/IP in general and CICS in particular are provided in the program directory that comes with the product.

Debug Tool has a partitioned data set called SEQAMOD, which contains the Debug Tool load modules. This data set and the Language Environment run-time library must be available to the application when it runs in order for you to debug it. You can either add these data sets to LINKLIB or use a STEPLIB to include them (see the specifics for each environment below).

For a batch application

Use either a STEPLIB statement or LINKLIB to make the SEQAMOD data set of Debug tool and the Language Environment run-time library available to the application.

CICS environments

Include the Debug Tool SEQAMOD data set and the Language Environment run-time library in the startup JCL for the CICS region.

DB2 environments

To debug a DB2 program that accesses tables update the JCL that starts DB2 so that it includes the SEQAMOD data set of Debug Tool and the Language Environment run-time library in the STEPLIB statement.

IMS environments

You can run IMS database (IMS DB) and IMS Transaction Manager (IMS TM) applications with or without Batch Terminal Simulator (BTS):

- With BTS, run in TSO in the foreground (IMS TM and DB).
- With BTS, run BTS as a batch job (IMS TM and DB).
- Without BTS, run as an IMS batch job (IMS DB only).
- Without BTS, run under IMS TM.

For BTS, you need to include the Debug Tool SEQAMOD data set and the Language Environment run-time library in the STEPLIB or TASKLIB data set concatenations for BTS. For IMS batch, you need to put Debug Tool SEQAMOD and the Language Environment run-time library in the STEPLIB concatenation (for DLIBATCH, for example). For IMS TM, include the Debug Tool SEQAMOD data set and the Language Environment run-time library in the IMS MPP STEPLIB.

Preparing your application for debugging

The application programs that you want to debug must be compiled on the host with the TEST compiler option, and the programs source must be in host data sets.

When you debug your program, the actual program statements are shown in the Distributed Debugger as they are executing. To show these statements, the debugger uses the programs source . The programs source must be in a permanent data set.

There may be additional preparation steps, depending on the environment where your application will run. Check the information for the particular environment.

Indicating the location of the Distributed Debugger daemon

Indicate the workstation where the Distributed Debugger daemon will run by using the TEST run-time option. This option has the form:

```
TEST(,,,VADTCPIP&machinename:*)
```

where *machinename* is the IP address of your workstation.

If you debug an application that uses DBCS, you also need to specify the VADSCP930 or VADSCP939 parameter in the TEST run-time option to indicate the host code page that you want Debug Tool to use. Here is an example:

```
TEST(,VACSCP930,,VADTCPIP&machinename:*)
```

The mechanism for setting the TEST run-time option varies depending on the environment. Link-editing your application with the LE CEEUOPT run-time options module is a technique used in several of the environments.

See *Debug Tool User's Guide and Reference*, SC09-2137, for:

- Full syntax of the TEST run-time option for debugging
- Instructions for creating the CEEUOPT run-time options module and using the CEEXOPT macro

For a batch application

The JCL to run the job must include the TEST run-time option.

```
//GO EXEC PGM=MYPROG ,PARM=('TEST(,,,VADTCPIP&machinename:*)/')
```

A slash (/) at the end of the TEST option shows that it is a run-time option and not a program parameter.

CICS environments

To indicate the workstation where the Distributed Debugger daemon is running, use one of the following techniques:

- Use Debug Tool transaction DTCN to identify the workstation location just before you start the application.
- Link-edit LE CEEUOPT run-time options module. The CEEUOPT module must include the TEST run-time option with the workstation IP address.

DB2 environments

The way that you indicate the location of the Distributed Debugger daemon depends on the particular type of DB2 program:

- When you submit the DB2 program by using the TSO RUN command, set the PARMS parameter as follows:

  ```
  RUN PROG (MYPROG)
      PLAN (MYPLAN)
      LIB ('MY.TEST.LOAD')
      PARMS ('TEST(,,,VADTCPIP&machinename:*)/')
  ```

  If *machinename* is the numeric IP address, the subparameter must be in quotation marks. A slash (/) at the end of the TEST option shows that it is a run-time option and not a program parameter.
- When you run a batch DB2 program, you must link-edit the program with the LE CEEUOPT run-time options module to set the TEST run-time option.

IMS environments

To indicate the workstation where the Distributed Debugger daemon is running, use one of the following techniques:

- When you link the program to be debugged, include a run-time options module. You must code and assemble the run-time options, including the TEST option, in this module. For example:

  ```
  CEEUOPT CSECT
      CEEUOPT AMODE ANY
      CEEUOPT RMODE ANY
      CEEXOPT TEST=(,,,"VADTCPIP&&9.11.22.33:*")
      END
  ```

  If you use the run-time options module method and you change the workstation where the daemon is running, assemble a new run-time options module and relink your program with the new module. You do not need to recompile the program.
- Link-edit the CEEUOPT run-time options module with the NOTEST run-time option to specify the workstation IP address. Include the TEST run-time option and call CEETEST.

  If you use CEEUOPT to specify the workstation IP address and that address changes, you must assemble CEEUOPT with the application link-edited again to pick up the new CEEUOPT

  For IMS TM, as an alternative to link-editing each application with CEEUOPT, you can use the LE CEEROPT run-time options module to set the TEST option for an entire region.

Starting the remote debugging session

Start the remote debugging session by using one of the following methods:

- From a project, click **Actions -> Start remote debugger daemon**.

- From the Start menu, click **Start Programs -> IBM VisualAge PLI -> Tools - Remote Access -> Remote debugger**.

A DOS window and then the Distributed Debugger Daemon window will open. You can minimize this window, but do not close it.

After you start the daemon, you can conduct as many remote debug sessions as you need without having to restart the daemon.

Starting your application on the host

Start your application on the host by using the TEST run-time option, which loads Debug Tool:

```
TEST(,,,VADTCPIP&machinename:*)
```

The variable *machinename* is the IP address for your workstation. For example, if mypc.mycompany.com is the address for your workstation, use:

```
TEST(,,,VADTCPIP&MYPC.MYCOMPANY.COM:*)
```

If your code has DBCS data, you can use the DBCS option as follows, where *xxx* can be 930 or 939, depending on which host code page you want Debug Tool to use:

```
TEST(,VADSCPxxx,,VADTCPIP&machinename:*)
```

The details of how to start your application vary depending on the host environment in which it runs. For a batch application, you can submit your job from the workstation while in the project, an edit session, or the job monitor.

RELATED CONCEPTS

"Remote debugging" on page 3

Issuing host commands from the workstation

If the optional remote commands server has been installed and configured on MVS, you can issue TSO commands from the workstation:

1. Open a host commands window from your project.
2. Submit TSO commands (such as TIME, LISTDS, and LISTCAT) to the host, run a CLIST, or run a program from a load library. You can select and edit recently executed commands.
3. Check the output.

You can paste text into the command input field or copy text from the output into an editor.

You can have an active connection to each MVS system at the same time.

Appendix. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

© Copyright IBM Corp. 1995, 2000

Readers' Comments — We'd Like to Hear from You

VisualAge PL/I
Remote Edit-Compile-Debug User's Guide
Version 2.1.7

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____

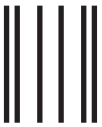
Phone No. _____

Readers' Comments — We'd Like to Hear from You



Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department HHX/H3
P.O. Box 49023
San Jose, CA
United States of America
95161-9023



Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line





Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.