README for Microsoft® Visual J++™ Version 1.0, Publisher's Edition

This document includes updated information for the documentation provided with Microsoft® Visual J++™ Version 1.0, Publisher's Edition. The information in this document is more up-to-date than the information found in the Help system. Many of the issues outlined in this document will be corrected in upcoming releases.

Please note that technical support for this product is not provided by Microsoft Corporation. For tips on troubleshooting your installation, see the Technical Support help topic included in the Microsoft Visual J++ 1.0 program group.

Contents

- Remote Connection Dialog Not Enabled
- The Debugger Only Supports Symbols Shorter than 255 Characters
- Compiler Error J0068 Thrown in Assignment

* Late Breaking News

Debugging Applets on the Internet

You can use the Visual J++ debugger to debug applets over the Internet. For example, assume you find a cool applet on your favorite website and want to debug it. If the applet author provides a link to the source, you can do source level debugging; otherwise, you can debug at the bytecode level. Another powerful feature is the ability to debug an applet inside a .CAB file. If you can run the applet in Internet Explorer 3.0, you can debug it!

Use these steps to debug a website applet:

- Start Visual J++, then select New... from the File menu.
- Select Project Workspace, and press OK.
- Select Java Workspace, specify a name for the new workspace, then click Create.
- Select Settings... from the Build menu. When the Project Settings dialog appears, select the Debug tab.
- In the Category drop-down, select General. Select the Browser (applets only) radio button. Enter the name of the .CLASS file you wish to debug into the Class for debugging/executing box.
  Remember that .CLASS names are case-sensitive. You can find the .CLASS file name by browsing to the website page with Internet Explorer and then selecting Source from the View menu. Look for APPLET tag.
- In the Category drop-down, select Browser. Select the Use parameters from HTML page radio button and enter the complete http://... address to the .HTML page containing the applet you wish to debug.
- Press OK on the Project Settings dialog.
- Select Debug/Step Into from the Build menu, then answer No to the dialog asking you to build.

At this point the debugger will break in the init() method and ask you for the source code location. Cancel this dialog and you are now debugging the applet over the Internet at the bytecode level. If you have the source, you can point the debugger to it and step through the source code, line by line.

Note that these same steps apply if the applet is packaged in a .CAB file.

Publisher's Edition Does Not Include Database Support

The Publisher's Edition of Visual J++ does not include DAO or RDO database support. As such, several sample applets found within Books Online that demonstrate DAO and RDO will not build or function properly.

* Setup

Setup Troubleshooting Information

Information regarding how to troubleshoot installation problems encountered during setup of Visual J++ 1.0 is available in the PSS.HLP file located in the \MSDEV\HELP directory on the CD-ROM. Open this file and click the Contents tab. Next, double-click Your First Stop For Visual J++ 1.0 Support, and then double-click Setup Troubleshooting Guide. This help topic will lead you to the appropriate information.

Windows 95

On some video adapters, when running setup under Windows 95, the setup progress bar displays improperly and mouse support is lost. To resolve this problem you will need to run a utility supplied by your

display adapter manufacturer that adjusts the refresh rates. For more information please see the Windows 95 README file.

File Conflicts During Installation

While installing new or updated versions of software on Windows 95 or Windows NT, you may encounter situations where the setup program identifies two files using the same name, and asks which file you wish to have installed. When asked, always select the newest version of the file.

DLL Files Hidden on Windows 95

By default, the Windows 95 Explorer sets DLLs as a hidden file type. If you want to see DLLs listed in File Open dialogs you can do one of two things:

- Remove DLLs from the Windows 95 Explorer hidden files list.
- After running Visual J++ Setup, log off and then log on again. Setup will reset the DLL file type so that it is no longer hidden.

Installing Updated Visual C++, Visual C++ for Macintosh, or Fortran Powerstation Packages

If you install any of the above packages **after** installing Visual J++ version 1.0, you may be unable to run the packages correctly in Microsoft Developer Studio until you make an adjustment. Developer Studio will notify you of this situation when you first try to run the newly installed product. A message box will tell you that Developer Studio has detected an incompatible version of the product you are trying to run.
The incompatibility does not preclude you from using Visual J++. Simply click OK in the message box and you can then use Visual J++ 1.0. The incompatibility does, however, preclude you from running any of the other listed products until you make the needed adjustment.
The fix needed for this incompatibility is stored on the Visual J++ 1.0 CD. To fix the problem, re-run the Visual J++ 1.0 Setup program, choose Custom installation and uncheck all the options. This step will automatically update the necessary files on your disk. After doing so, you will be able to run your add-on language program in Microsoft Developer Studio.

Visual C++ Standard Edition Not Compatible with Visual J++

Visual C++ Standard Edition's version of Microsoft Developer Studio does not support multiple packages. As such, you must install Visual J++ 1.0 and Visual C++ Standard Edition in separate directories.

Installing/Uninstalling on a Dual-Boot Machine

To install Visual J++ 1.0 on a dual-boot machine, run Setup once for each operating system. For example, boot the machine into Windows NT (version 4.0 or later) and run Setup. This will copy all the required files into the installation root directory (usually C:\MSDEV) and into the Windows NT \SYSTEM32 subdirectory. It will also set up a program group for Visual J++ and make the required registry entries. Next, reboot the machine into Windows 95 and run Setup again. You can choose the same installation directory. This will save disk space and installation time since Setup copies only those files that need replacement or updating. As a result, only system files will be copied to the system directory. Setup will also create a program group for Visual J++ and make the required registry entries for Windows 95. When you have finished the installations, you can use Visual J++ with either operating system. Because the operating systems do not share the same registry, Visual J++ will have to be configured separately for each one.
Note If you wish to uninstall Visual J++ from a dual-boot system, you will need to run the uninstall application from the operating system on which Visual J++ was originally installed. Therefore, if you originally installed Visual J++ using Windows 95, you will need to run the uninstall application from Windows 95. The Visual J++ registry entries will not be removed on the other operating system and will need to be removed manually.

Application Error During Online Registration

If you choose Online Registration after setting up Visual J++, you may get an application error in RegWiz.EXE. You can close the application error dialog box and restart your computer. After restarting your computer, you should be able to register Visual J++ electronically by clicking the "Register Visual J++" icon. If you are unable to run RegWiz, then you can fill out the registration card included in the Visual J++ package.

Setup Cannot Write to the Registry

Near the end of the Visual J++ installation, you may encounter a dialog box warning that "Setup Cannot Write To The Registry." The only option available is to click OK, at which time the setup continues. You can safely ignore this warning.

This warning is generated when installing Visual J++ from an account which does not have administrative privileges. This is a problem with the setup program only– it is not required that you have administrative privileges to install or use Visual J++.

The information Visual J++ stores in the registry is not critical. It is primarily information about the last few project and source files that were opened, window sizes/positions, and so on. If a registry entry is not present when Developer Studio is started, a new one is created with the original default settings.

Access Permission for Installation Directory

The Setup program requires that you have access permission for the root directory of the drive where you are installing Visual J++. Setup will fail if you do not have access rights. Under these circumstances, your system administrator must give you temporary access rights to the root directory for you to complete setup.

Real-Mode CD-ROM Drivers Under Windows 95 May Cause Error During Setup

If you are running Windows 95 and are using real-mode CD-ROM drivers, you will not be able to access long filenames from the Visual J++ CD. This may cause errors during Setup.

* Documentation Errata

User's Guide: Updating Dependent Files in a Project

Visual J++ 1.0 does not currently support use of dependency folders. As such, the following sentence, located at the beginning of the topic **Updating Files in a Project**, is incorrect:

"After editing one or more source files, you can explicitly update the project to add dependent files to the appropriate dependency folders."

The sentence should read like this:

"After editing one or more source files, you can explicitly update the project to synchronize dependent files with their dependencies."

User's Guide: Using the Variables Window

In the topic **Using the Variables Window**, you can ignore the procedure for viewing the return value of a method. Visual J++ does not show return values in the variable window.

* Development Environment

BRIEF Editor Emulation

The BRIEF system of window controls has not been completely emulated. The Create Window command brings up splitter bars within the active document window or activates movement of existing splitters. When the splitters are activated, ARROW keys allow placement of the splitters. Activated splitters are anchored by pressing ENTER. Change Window commands move between split panes. The Delete Window commands delete either horizontally or vertically adjacent panes.

Generating Class Files in Package Directories

By default the build system creates package directories when a package statement is encountered within a .java file. However, the corresponding .class output file will not be generated in this newly created package directory. To ensure the output .class file is generated within the appropriate package directory, perform the following steps:

1. Select Settings... from the Build menu.
2. Type a period in the Output directory box of the General Tab.

Performing the steps shown above tells the build system to place the corresponding output .class files into the newly created package directories.

New Project Dependency Updates May Affect Projects Created With Visual J++ Beta Releases

Projects originally created with beta or trial versions of Visual J++ may receive the following error message when used with the final release version of Visual J++ 1.0:

"One or more dependencies are out of date. Do you wish to rebuild them?"

This error message may appear every time you execute or debug.

If you encounter this error and it persists, use the following steps to correct this behavior:

1. Manually delete all files with the extension .DEP from your project and output directories.
2. Select Update All Dependencies from the Build menu. (Perform this step for both the Release and Debug versions of your project.)
3. Select Rebuild All from the Build menu. (Perform this step for both the Release and Debug versions of your project.)

Using Your Own Help Files for F1 Help in a Source File

You can customize Microsoft Developer Studio to get F1 help on keywords documented in a private or third-party help file. Once this feature is set up, Developer Studio adds a command to the Help menu allowing you to toggle the feature on or off. When the command is active, all source-file F1 help requests are passed to a specified external file or files rather than to Visual J++ Books Online. WinHelp 4.0 displays the topic.

Manually Refreshing Source-Code Control Status

If you have installed a source-code control system that conforms to the Microsoft Source Code Control Interface, under certain circumstances you may need to manually refresh the status of your files under source-code control. Microsoft Developer Studio automatically refreshes the source-code control status for files in a project workspace until the number of files reaches a certain threshold value. This value is the Project Threshold, set in the Registry under Source Control in the Microsoft Developer entry for the current user. After that value is reached, Developer Studio disables the automatic refresh for performance reasons. If you need to refresh the source-code control status of your files after the automatic refresh has been disabled, from the Tools menu, choose Source Control, and from the cascading menu, choose Refresh Status.

Information Title Not Found Error

If you install Microsoft Developer Studio into a directory other than the default, and receive a dialog titled "Information Title Not Found", verify that the path provided is correct and that it ends with a backslash. Click

the Open button to continue after completing any necessary changes to the path. This problem is known to occur when certain double-byte characters with the trailing byte of 0x5c appear as the last character of the help directory name. This problem only occurs on operating systems that support double-byte characters such as Japanese Windows 95.

Source-Code Control Integration with Visual J++ 1.0

Visual J++ 1.0 does not ship with a source-code control system.
Microsoft Developer Studio provides facilities for integrating a source-code control system into the development environment. If you install a source-code control system that conforms to the Microsoft Common Source Code Control Interface, you can directly access source-code control functionality from the Developer Studio menus.
Visual Source Safe 4.0 conforms to the Microsoft Common Source Code Control Interface. Earlier versions of Source Safe do not. If you currently use a source-code control system other than Visual Source Safe, you may wish to contact your source-code control vendor to inquire about an update that conforms to the Microsoft Common Source Code Control Interface.

Visual SourceSafe: Error Accessing Source Code Control System

Using Microsoft Visual SourceSafe, you may encounter problems attempting to add more than 300 files to a project via Developer Studio's "Add To Source Control" menu item. Similar problems may occur when attempting to add files to a project already containing more than 300 files (including dependencies).
Symptoms include error messages such as the following:

```
Error accessing Source Code Control system. Check Installation.
```

Suggested work-arounds:

1. Add the files to the SourceSafe project directly from the SourceSafe Explorer.
2. Use RegEdit to edit the HKEY_CURRENT_USER\SOFTWARE\Microsoft\Developer Source Control key, and change the "ProjectThreshold" value to a value larger than the number of files in your project (including dependencies).

   Note Increasing this value will impact overall Source Code Control performance. Only increase this value as much as necessary.
3. Contact Product Support Services for Microsoft Visual SourceSafe. See the information that came with your Microsoft SourceSafe product for details.

Combo Boxes Cannot Be Populated in Resource Wizard

After adding a combo box to a dialog in the resource editor, you are allowed to populate the control with strings. While the resource editor allows this, the strings will not be displayed in the listbox when your program is run.

The GIF and JPEG Files Supported by the Image Editor Cannot Be Loaded as Resources

The image editor has the capability to read and write GIF and JPEG graphic files. The Resource View only supports BMP files. GIF and JPEG files cannot be loaded in Resource View. Additionally, you cannot use the Insert Resource command to create GIF and JPEG files. To add a GIF or JPEG file to a project, you must use the Insert File command.

To create a new GIF or JPEG file and add it to a project

1. From the File menu, choose New.
   The New dialog box appears.
2. Select Bitmap File, and then choose OK.
   The image editor appears.
3. Use the image editor to create the image.
4. From the File menu, choose Save As.
   The Save As dialog box appears.
5. In the Save File As Type list box, select Bitmap File (*.bmp;*.dib;*.gif;*.jpg).

6 In the File Name box, append a GIF or JPG extension to the filename you choose.
7 Click the Save button.
8 From the Insert menu, choose Files Into Project.
The Insert Files Into Project dialog box appears.
9 In the List Files Of Type list box, choose Image Files (*.bmp;*.dib;*.gif;*.jpg).
10 Select the GIF or JPEG file you just saved.
11 Choose OK.

The GIF or JPEG file will appear in the File View. You can double-click the file icon and the image editor will appear with the image loaded.

Bilevel GIF File Format Not Supported

Bilevel (1-bit) GIF files may not display correctly when loaded into the image editor. Some other GIF file formats remain untested. If you encounter problems displaying or updating GIF files within the image editor, use another picture editor that supports a wider variety of GIF file formats to perform your updates.

* Using Visual J++ with Windows 95

Batch File Builds Cannot Be Halted from Developer Studio

Under Windows 95, Visual J++ is unable to stop a build that uses a batch file (.BAT) on the command line for an external project type. The Stop Build command on the Build menu will do nothing if a batch file is being run as part of the build. The build must continue to completion.

* Compiler and Tools

Remote Connection Dialog Not Enabled

Remote Connection, a utility allowing you to connect to other machines via the network or by serial modem cable, is not supported in Visual J++ 1.0.

The Debugger Only Supports Symbols Shorter than 255 Characters

The debugger only supports symbol names that are shorter than 255 characters. Any symbol longer than the 255-character limit will be truncated, and it will not be evaluated and updated in the debugger.

Compiler Error J0068 Thrown in Valid Assignment

The following sample throws J0068 - "cannot implicitly convert int to short" when compiled with JVC:

```
class Simple {
    void method1() {

        short s1 = 1, s2 = 2, s3 = 3;
        s1 /= s2 * s3; // problem occurs here
    }
}
```

To correct this problem, replace the expression shown above with the following line:

```
        s1 /= (short) (s2 * s3);
```

Note that the behavior shown is not a bug. Rather, it is in full compliance with the Java Language Specification.