



Visual CaféTM

for JavaTM

What's New

Professional Development Edition

Database Development Edition

Windows 95

Windows NT



Visual CafeTM for JavaTM

Windows Edition

What's New

Symantec Visual Cafe™ for Java™ What's New

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Copyright Notice

Copyright © 1998 Symantec Corporation.

All Rights Reserved.

Released: 3/98 for Visual Cafe for Java 2.5

This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent in writing from Symantec Corporation, 10201 Torre Avenue, Cupertino, CA 95014.

ALL EXAMPLES WITH NAMES, COMPANY NAMES, OR COMPANIES THAT APPEAR IN THIS MANUAL ARE IMAGINARY AND DO NOT REFER TO, OR PORTRAY, IN NAME OR SUBSTANCE, ANY ACTUAL NAMES, COMPANIES, ENTITIES, OR INSTITUTIONS. ANY RESEMBLANCE TO ANY REAL PERSON, COMPANY, ENTITY, OR INSTITUTION IS PURELY COINCIDENTAL.

Every effort has been made to ensure the accuracy of this manual. However, Symantec makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability and fitness for a particular purpose. Symantec shall not be liable for any errors or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual or the examples herein. The information in this document is subject to change without notice.

Trademarks

Symantec Visual Cafe, Symantec, and the Symantec logo are U.S. registered trademarks of Symantec Corporation.

Other product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are the sole property of their respective manufacturers.

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

SYMANTEC LICENSE AND WARRANTY

The software which accompanies this license (the "Software") is the property of Symantec or its licensors and is protected by copyright law. While Symantec continues to own the Software, you will have certain rights to use the Software after your acceptance of this license. Except as may be modified by a license addendum which accompanies this license, your rights and obligations with respect to the use of this Software are as follows:

- You may:
 - (i) use one copy of the Software on a single computer;
 - (ii) make one copy of the Software for archival purposes, or copy the software onto the hard disk of your computer and retain the original for archival purposes;
 - (iii) use the Software on a network, provided that you have a licensed copy of the Software for each computer that can access the Software over that network;
 - (iv) after written notice to Symantec, transfer the Software on a permanent basis to another person or entity, provided that you retain no copies of the Software and the transferee agrees to the terms of this agreement; and
 - (v) if a single person uses the computer on which the Software is installed at least 80% of the time, then after returning the completed product registration card which accompanies the Software, that person may also use the Software on a single home computer.
- You may not:
 - (i) copy the documentation which accompanies the Software;
 - (ii) sublicense, rent or lease any portion of the Software;
 - (iii) reverse engineer, decompile, disassemble, modify, translate, make any attempt to discover the source code of the Software, or create derivative works from the Software; or
 - (iv) use a previous version or copy of the Software after you have received a disk replacement set or an upgraded version as a replacement of the prior version, unless you donate a previous version of an upgraded version to a charity of your choice, and such charity agrees in writing that it will be the sole end user of the product, and that it will abide by the terms of this agreement. Unless you so donate a previous version of an upgraded version, upon upgrading the Software, all copies of the prior version must be destroyed.

- Sixty Day Money Back Guarantee:

If you are the original licensee of this copy of the Software and are dissatisfied with it for any reason, you may return the complete product, together with your receipt, to Symantec or an authorized dealer, postage prepaid, for a full refund at any time during the sixty day period following the delivery to you of the Software.

- Limited Warranty:

Symantec warrants that the media on which the Software is distributed will be free from defects for a period of sixty (60) days from the date of delivery of the Software to you. Your sole remedy in the event of a breach of this warranty will be that Symantec will, at its option, replace any defective media returned to Symantec within the warranty period or refund the money you paid for the Software. Symantec does not warrant that the Software will meet your requirements or that operation of the Software will be uninterrupted or that the Software will be error-free.

THE ABOVE WARRANTY IS EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE.

- Disclaimer of Damages:

REGARDLESS OF WHETHER ANY REMEDY SET FORTH HEREIN FAILS OF ITS ESSENTIAL PURPOSE, IN NO EVENT WILL SYMANTEC BE LIABLE TO YOU FOR ANY SPECIAL, CONSEQUENTIAL, INDIRECT OR SIMILAR DAMAGES, INCLUDING ANY LOST PROFITS OR LOST DATA ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF SYMANTEC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

IN NO CASE SHALL SYMANTEC'S LIABILITY EXCEED THE PURCHASE PRICE FOR THE SOFTWARE. The disclaimers and limitations set forth above will apply regardless of whether you accept the Software.

- U.S. Government Restricted Rights:

RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software-Restricted Rights clause at 48 CFR 52.227-19, as applicable, Symantec Corporation, 10201 Torre Avenue, Cupertino, CA 95014.

- General:

This Agreement will be governed by the laws of the State of California. This Agreement may only be modified by a license addendum which accompanies this license or by a written document which has been signed by both you and Symantec. Should you have any questions concerning this Agreement, or if you desire to contact Symantec for any reason, please write:

Symantec Customer Service, 175 W. Broadway,
Eugene, OR 97401

Language Addendum

If the Software is a Symantec language product, then you have a royalty-free right to include object code derived from the libraries in programs that you develop using the Software and you also have the right to use, distribute, and license such programs to third parties without payment of any further license fees, so long as a copyright notice sufficient to protect your copyright in the program is included in the graphic display of your program and on the labels affixed to the media on which your program is distributed. You also have a royalty-free right to include unmodified Symantec Class files required by your programs. The Symantec Class files are in the following installed zip file in the Visual Cafe directory – VisualCafe\redist\symbeans.jar. The Java Virtual Machine (VM) or Just In Time (JIT) compiler are not freely distributable.

What's New in Visual Cafe for Java

Visual Cafe 2.5 adds power and flexibility to your style of developing Java applets and applications. This document covers the new features in this release for Professional Development Edition and Database Development Edition. The information included here is in addition to what you can find in the *Visual Cafe User's Guide* and similar to the information in the Online Help.

The features in Visual Cafe Professional Development Edition (PDE) and Database Development Edition (dbDE) are distributed as shown in the following table.

Feature	Page number
Customizable user interface	page 2
Bean Wizard	page 6
JAR Packager	page 8
JavaBeans Dynamic Development	page 19
Just-in-Time Compiler enhancement	page 21
JFC Swing component support	page 21
JDK 1.1.5 support	page 22
Visual Cafe Open APIs	page 22
Browser debugging	page 23
Enabling and disabling RAD	page 25
Support for JDBC (dbDE only)	page 27

Feature	Page number
Wizard component	page 40
JAR Viewer	page 41
Conversion from no layout to GridBagLayout	page 42
JavaBean customizer support	page 43
Sorting files in the Files view	page 44
Using Visual Cafe 2.1 menus	page 45
J++ keyboard shortcuts	page 46
Java virtual font classes support	page 48

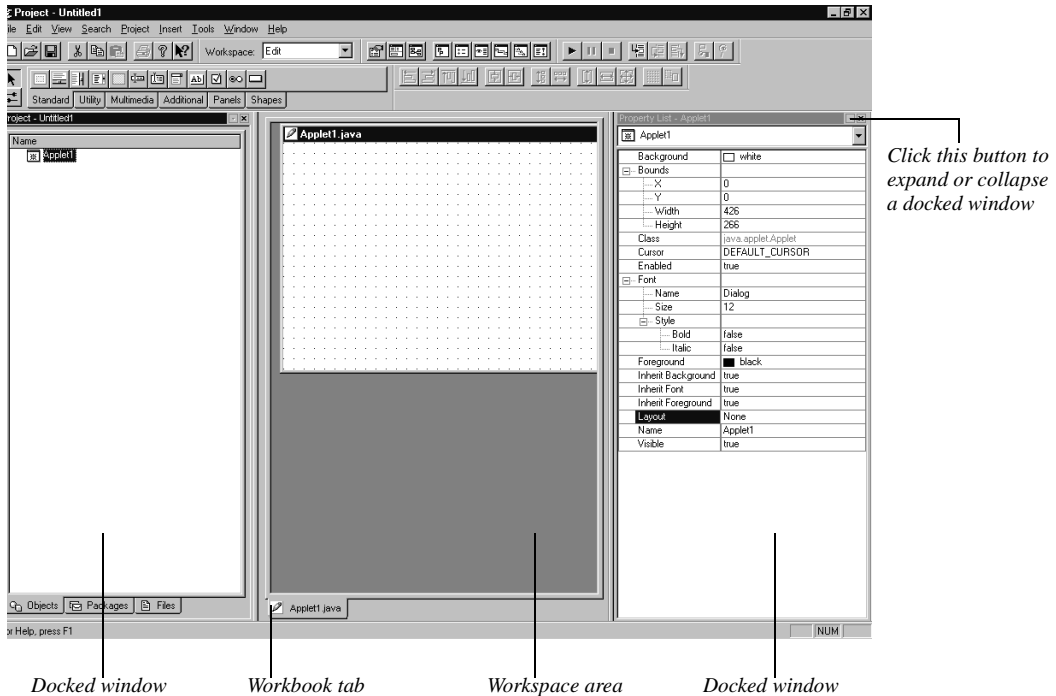
To go directly to information on a specific feature, go to the page number listed for that feature.

Customizable user interface

Visual Cafe Professional and Database Development editions allow you to customize your workspace even further by retaining your window preferences between Edit and Debug development modes. You can control your workspace by docking or undocking windows and by using the Workspace and Workbook tabs. Docking a window makes the window stay against the edge of the Workspace area. If you like to have different windows open and in varying locations depending on what development mode you are working in, you can keep these preferences, even after closing a project or Visual Cafe. For each project, you can retain your settings for the Project, Source, Class Browser, and Hierarchy Editor windows.

These options are available to you when you are in Multiple Document Interface (MDI) mode. If you do not turn MDI on, your workspace will look the same as it did in versions of Visual Cafe prior to version 2.5.

Here is an example of what MDI environment looks like.



To enable MDI

- 1 Choose Tools > Environment Options > General Tab.
- 2 Select MDI Development Environment to enable MDI, or deselect it to turn MDI off.

While using MDI, you work with dockable and non-dockable windows. Non-dockable windows appear in the workspace area; dockable windows can be docked along the edges of your Visual Cafe workspace or can float in the workspace area. The windows that are dockable are:

- Breakpoints
- Calls
- Component Library
- Find in Files results
- Messages

- Project Window
- Property List
- Threads
- Variables
- Watch

Note: Docked windows have the same highlight color in the title bar as a non-dockable window that is selected. To ensure that a docked window is the active window, click it.

When using MDI mode, some window positions are saved with the workspace and the project. For more information on setting your workspace options, see the *User's Guide* or Online Help.



Tip: If you have maximized a non-dockable window, such as the Source window, you can minimize it again by clicking the innermost Restore button.

To use MDI, you can:

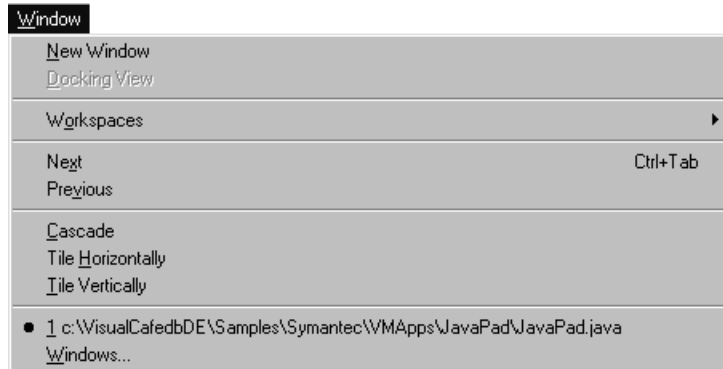
- Turn Docking View on or off for a dockable window
- Dock a window floating in the workspace area
- Change the size of a docked window
- Prevent a dockable window from docking while dragging it
- Expand or collapse a docked window
- Organize the placement of windows in the workspace area
- Toggle the display of Workbook tabs
- Activate a window in the workspace area
- Toggle the display of the status bar

To turn Docking View on or off for a dockable window

- Right-click and select Docking View from the context menu.

or

- Activate the window, then choose Window > Docking View, as shown below:



When you dock a window that you've docked before, the window will dock to its previous size and location.

To dock a window floating in the workspace area

- While Docking View is enabled, drag the window to the edge of the workspace.

or

- Double-click the title bar of the window.

A docked window takes up an entire side of the workspace, unless another window is docked along the same side.

To change the size of a docked window

- Move the cursor over the edge of the window until the cursor changes to a sizing cursor, then click and drag to the desired size.

To prevent a dockable window from docking while dragging it

- Press Control while dragging.

To expand or collapse a docked window



- If you have more than one window docked along an edge, you can expand or collapse one of the windows by clicking the triangle button in the corner of the window.

Expanding and collapsing windows along an edge will resize the windows so that they all use equal amounts of space.

To organize the placement of windows in the workspace area

- Choose Window > Cascade, Tile Horizontally, or Tile Vertically. You can also drag the windows within the workspace.

The workspace area contains non-dockable windows, such as the Form Designer or Source window, and dockable windows that are not docked.

To toggle the display of Workbook tabs

At the bottom of the workspace area, you can display Workbook tabs — one for each open window (either non-dockable windows or dockable windows with Docking View disabled). An example is shown below:



Clicking a tab activates that window in the workspace area.

- Choose View (or Window) > Workbook.

To activate a window in the workspace area

- Choose the window from the Window menu. You can also click the window, click a Workbook tab, or choose Window > Next or Previous.

To toggle the display of the status bar

- Choose View (or Window) > Status Bar.

Bean Wizard

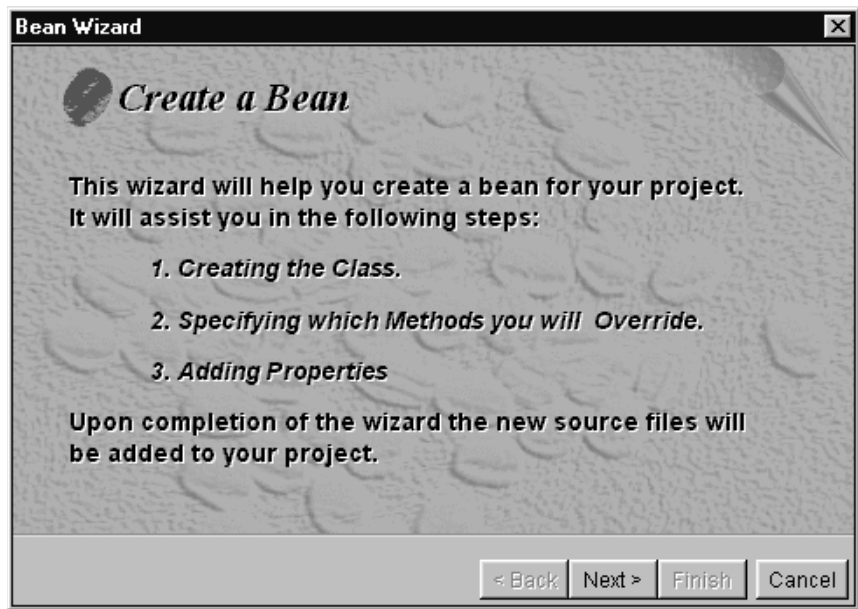
Use the Bean Wizard to help you easily create new JavaBeans. The wizard steps you through a series of choices and then generates the class and

BeanInfo source files for you, based on your selections. The wizard creates two files; `beanclassBeanInfo.java` and `beanclass.java` where *beanclass* is the name of your bean

To learn more about JavaBeans, refer to Sun Microsystems' website at <http://java.sun.com> for the JavaBeans API specification.

To start the Bean Wizard

- 1 Choose File > New Project.
- 2 Select Bean Wizard and click OK. The Bean Wizard appears, as shown here:



The Bean Wizard consists of these steps:

- 1 Enter a bean name and package (choosing a package is optional).
- 2 Choose a bean type.
- 3 Choose a bean weight or base class (if creating a custom bean).
- 4 Select override methods (optional).
- 5 Create properties (optional).
- 6 Add icons (optional).
- 7 Review your choices.

Each step in the Bean Wizard includes an explanation of the associated task. However, there are some things you should know when creating a JavaBean:

- If you've chosen to add icons, they also get added to your project, as shown in the Files tab. This allows the icons to be added to the JAR when building your project.
- On the Create Properties screen there are five abbreviations, defined as follows:

Abbreviation	Definition
Bd	Bound
Cd	Constrained
Stg	Storage
Rd	Read Method
Wt	Write Method

JAR Packager

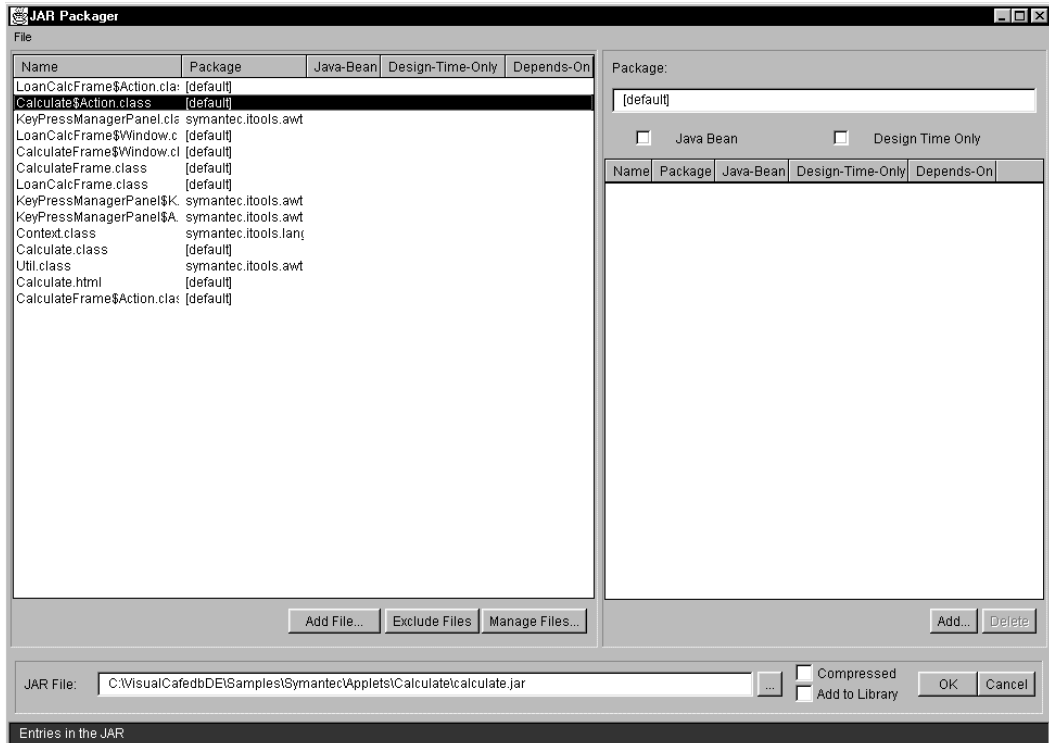
Use the JAR Packager to quickly create Java Archive (JAR) files. The JAR Packager sets up much of the JAR for you. Initially, the display includes all of the classes in your project, any classes they depend on, and other files you have added to the project. You can also create JAR files with the AutoJAR tool, as described in “JavaBeans Dynamic Development” on page 19.

You cannot remove one component independently of the others in the JAR file, nor can you add to an existing JAR file. Instead you need to create a new JAR file.

Tip: If you want non-Java files to be automatically included in your JAR, you should add them to the project. Ideally, these files should be in the project directory, project output directory, or their subdirectories; otherwise, Visual Cafe might not be able to determine the correct package name.

If you're already shipping one set of resource files in another JAR, you can exclude those duplicate files from the current JAR. You can also specify whether a file within a JAR is a JavaBean or used only at design-time.

Here's an example of what the JAR Packager looks like:



When working with the JAR Packager, you can resize its window by clicking and dragging on any of the edges of the dialog box.

Creating a JAR

You need to create a new JAR file if there are changes you want to make to a current JAR file. When you add a JAR file to the Component Library, all JavaBeans components in the JAR file are added (as specified in the manifest file).

To create a JAR

- 1 If needed, set up the `Bundler.properties` file so that it contains the JAR options you want. For details, see “Wizard component” on page 40.
- 2 Make sure your environment options are set to the type of release you want to create, either Debug or Final.

By default, when the release type is Debug, all dependent files that are in archive files (JAR or ZIP) are not included in the JAR. By default, these files are included when the release type is Final.
- 3 Activate the project you are working with, and choose Project > JAR.
- 4 If you use Swing components in the project, you may be prompted to specify the look and feel that you want to use. Select the desired look and feel items, then click OK.

The JAR Packager displays. The left panel shows the files in your JAR, and the right panel is where you specify information about the files in the JAR.

- 5 Add or remove files as desired.
 - To add files to the JAR, select a file and click Add File.
For more details, see the following section “Adding files to a JAR.”
 - To remove files from the JAR, select one or more files and click Exclude Files.
 - To add or remove files, packages, and archives that were automatically added to the JAR when you started the JAR tool, click Manage Files.
For details about managing files, packages, and archives in a JAR, see the section “Managing files in a JAR.”
- 6 To specify information about a file, select a file in the left panel and specify the option in the right panel (optional):
 - To specify a package, type the package name in the Package field.
Visual Cafe tries to specify the correct package for you. However, if a file is not in the project directory, project output directory, or their subdirectories, Visual Cafe might not be able to determine the correct package name.
 - To identify a class file as a JavaBean, select JavaBean.

If a class file has a *name*BeanInfo.java, *name*Customizer.java, or *name*Editor.java file corresponding to a *name*.class file, Visual Cafe automatically marks that class file as a JavaBean.

- Select Design Time Only if a class file is needed only at design time, such as a BeanInfo file, which is used when you add a JAR to the Component Library.

If a class file has a *name*BeanInfo.java, *name*Customizer.java, or *name*Editor.java file corresponding to a *name*.class file, Visual Cafe automatically marks that class file as Design Time Only.

- To specify dependencies within the JAR, click Add and select a file that the class depends on, such as a graphics file or another support file. (This flag is the same as the depends flag in the manifest file).

Note: The file must be listed in the left panel before you can specify dependencies with it; clicking Add does not let you add files.

- 7 The JAR Packager provides a default path and filename, based on your project name and location. This default filename and path is shown in the JAR File field. If you prefer, type the name and full path that you want the JAR file to have. Click the browse button, which is indicated with an ellipsis (...), to go to a specific path, and then enter the filename.
- 8 To create a compressed JAR file, select Compressed.

Tip: You probably do not want to add a compressed JAR file to the Component Library, because it takes more time for the development environment to access the files in the JAR. Also, some browsers may not support compressed JAR files.

- 9 To automatically add the JavaBeans in the JAR to the Component Library, click Add to Library.
- 10 Click OK to create the JAR.

The JAR is created. If Add to Library was selected, any JavaBeans in the JAR are automatically added to the Component Library.

Adding files to a JAR

Add File allows you to add additional files to the JAR that may not be in your current project. These could be GIF or HTML files, as well as additional class files that may be dynamically loaded.

To add files to a JAR

- 1 Click Add File.
The Select File dialog box is displayed.
- 2 Select a file to add and click OK.
The files are added to the JAR that you are creating.
- 3 If needed, change the package name in the Package field.
- 4 Repeat steps 1 and 2 to add additional files.

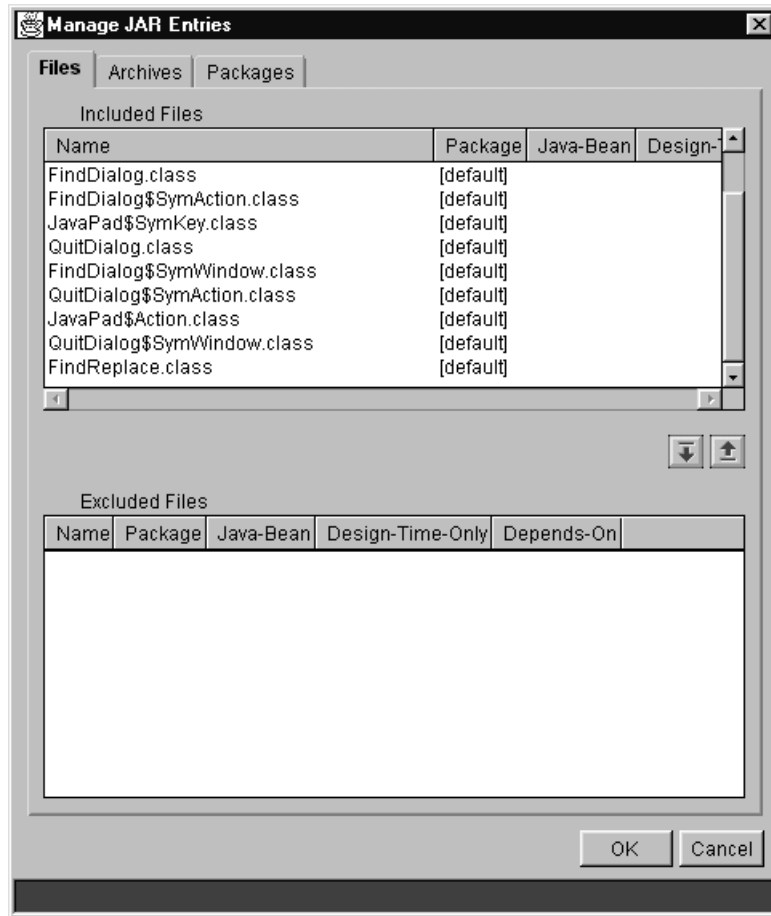
Note: The Select File dialog box does not allow selection of multiple files.

Managing files in a JAR

Use Manage Files to add a file from a package or archive that you have previously excluded from the current JAR. You can also remove all files at once from a given package or archive.

To manage your files, packages, and archives

- 1 Click Manage Files.
The Manage Files dialog box displays. An example follows:



- 2 Select the Files, Packages, or Archives tab to manage what items are excluded or added.

To select multiple files, press Shift while clicking on the files.



- 3 Add any items you have excluded back into the JAR by using the Add button.



- 4 Remove any items you no longer want in the JAR by using the Remove button.

- 5 Click OK to save the changes or Cancel to remove the changes.

Specifying JAR options in the Bundler.properties file

You can customize how the JAR Packager tool and the AutoJAR command set up JAR files by configuring the `Bundler.properties` file. The `Bundler.properties` file is located in the Visual Cafe `\java\lib` directory, and a default version is provided in your installation.

Customize your JAR file creation process by:

- Preserving your JAR Packager settings
- Changing what archive files are included in JARs for Debug and Final release types
- Creating a JAR even if there are compiler warnings
- Including or excluding Swing classes
- Prompting for Swing look and feels
- Specifying the default Swing look and feels
- Specifying where to look for Swing JAR files
- Displaying the JAR Packager version number
- Gathering troubleshooting data for Symantec Technical Support

Each of these tasks are explained below. Most of these options apply to both the JAR Packager and AutoJAR command, except where noted.

Preserving your JAR Packager settings

If you make changes in the JAR Packager, changes such as the JAR path and filename, compression, and whether the JAR is added to the Component Library are always preserved when you next open a JAR tool. If you want to preserve changes made in the rest of the JAR tool, such as whether a component is a JavaBean, used at design time only, or what files it depends on, you can set the `PersistData` option to true:

```
PersistData=true
```

The default setting is false. When `PersistData` is set to true, the JAR tool saves the JAR settings separately for each project.

Changing what archive files are included in JARs for Debug and Final release types

When the release type is Debug, all dependent files that are in archive files (JAR or ZIP) are by default not included in the JAR. In contrast, these files are included by default when the release type is Final. This activity is controlled by the `IgnoreArchivesInDebug` and `IgnoreArchivesInFinal` settings (default is false).

To include these dependent archive files in Debug builds, set `IgnoreArchivesInDebug` to false:

```
IgnoreArchivesInDebug=false
```

The default setting is true.

To not include these dependent archive files in Final builds, set `IgnoreArchivesInFinal` to true:

```
IgnoreArchivesInFinal=true
```

The default setting is false.

Creating a JAR even if there are compiler warnings

A JAR tool will not, by default, create a JAR if there are compiler warnings. To create a JAR despite compiler warnings, set `TestSJOutput` to false:

```
TestSJOutput=false
```

The default setting is true.

Including or excluding Swing classes

If you use Swing components in your project, Swing classes are included in your JAR by default. However, including these classes can make your JAR quite large and your users may already have the Swing classes on their computers. In this case, you could exclude Swing classes by setting `Swing.include` to false:

```
Swing.include=false
```

The default setting is true.

Prompting for Swing look and feels

The JAR Packager prompts by default for Swing look and feels to include in your JAR:

```
Swing.promptForLAFs=true
```

Before the JAR Packager prompts you, however, the Swing classes need to be included in the JAR. That means the `IgnoreArchivesInDebug` or `IgnoreArchivesInFinal` property needs to be set to true for the corresponding release type you are creating when generating the JAR, and `Swing.include` must be true.

For Debug release types, if either the `IgnoreArchivesInDebug` or `Swing.include` property is set to false, the Swing classes are not included, and the JAR Packager does not prompt you for Swing look and feels. This is also the case for Final release types, if either the `IgnoreArchivesInFinal` or `Swing.include` property is set to false.

This setting does not apply when using the AutoJAR command.

Specifying the default Swing look and feels

With the `Swing.defaultLAFs` property, you can set default Swing look and feels to include when you create a JAR with the JAR Packager or the AutoJAR command.

```
Swing.defaultLAFs=metal, motif
```

If `Swing.promptForLAFs=true`, the default Swing look and feels appear in the dialog box when you create a JAR with the JAR Packager.

The values must be listed in the `Swing.LAFs` section. See the Online Help or the `Bundler.properties` file for the default settings.

Specifying where to look for Swing JAR files

Visual Cafe looks in the Visual Cafe Jfc directory for the Swing JAR files. You can use the `Swing.home` property to change this location if you have your JFC files somewhere else. For example, if you have your Swing JAR files in your `c:\jars` directory, set `Swing.home` to `c:\jars`:

```
Swing.home=c:\jars
```

Displaying the JAR Packager version number

The JAR Packager version number appears in the JAR Packager title bar by default. To not display the version number, set the `DisplayVersion` property to false:

```
DisplayVersion=false
```

The default setting is true.

Gathering troubleshooting data for Symantec Technical Support

During normal operation, the Debug and KeepFiles properties should be set to false. This is the default. If you need to gather troubleshooting information, you can set them to true:

```
Debug=true
KeepFiles=true
```

When Debug is true, debug logging starts. When the KeepFiles property is true, Visual Cafe keeps the intermediate files (`bundler.dar` and `bundler.rsp`) that the JAR Packager generates so that Technical Support can examine them.

By default, the debug log is kept in `c:\Debug.log`. You can change this location and name with the DebugLog property. For example:

```
DebugLog=C:\MyDebug.log
```

Sample Bundler.properties file

The contents of the default `Bundler.properties` file is duplicated here for your convenience.

```
# - example Bundler.properties file (3.1.0).
# - All of the entries are commented out & refer to their
# default values. To change a setting you need to uncomment
# the line and set the value.

# - Turn on the debug logging.
# Debug=false

# - The debug file to write to.
# DebugLog=C:/Debug.log

# - Fail when a build gives warnings
# TestSJOutput=true

# - Keep the intermediate files that the JAR Packager
# generates
#   - bundler.dar
#   - bundler.rsp
```

```
# KeepFiles=false

# - Save the included/excluded class lists.
# PersistData=false

# - Display the JAR Packager version number
# DisplayVersion=true

# - Include files that come for JAR/ZIP files for Debug
  builds.
# IgnoreArchivesInDebug=true

# - Include files that come for JAR/ZIP files Final builds.
# IgnoreArchivesInRelease=false

# - Include swing classes - effects all Swing options.
# - If the IgnoreArchivesIn??? flag is true then Swing
#   is excluded regardless.
# Swing.include=true

# - Where to look for the swing JAR files.
# - You only need to set this if your JFC files are not
#   located in the <VisualCafeInstall>\jfc directory.
# Swing.home=c:/jars

# - Show dialog of LAFs to select from.
# - Auto JAR does not use this.
# Swing.promptForLAFs=true

# - The default swing LAFs to include.
# - Included in JAR if no prompt; checked on dialog if
# - prompt. Auto JAR will use these.
# - The entries must be listed in the SwingLAFs section.
# Swing.defaultLAFs=metal,motif

# - The required look and feels.
# Swing.mandatory.LAFs=plaf,basic
# Swing.mandatory.LAF.plaf=swingall.jar
# Swing.mandatory.LAF.plaf.base=com/sun/java/swing/plaf
# Swing.mandatory.LAF.basic=swingall.jar
```



```
# Swing.mandatory.LAF.basic.base=com/sun/java/swing/plaf/basic

# - The optional look and feels.
# Swing.LAFs=metal,motif,windows
# Swing.LAF.metal=swingall.jar
# Swing.LAF.metal.base=com/sun/java/swing/plaf/metal
# Swing.LAF.motif=swingall.jar
# Swing.LAF.motif.base=com/sun/java/swing/plaf/motif
# Swing.LAF.windows=swingall.jar
# Swing.LAF.windows.base=com/sun/java/swing/plaf/windows
```

JavaBeans Dynamic Development

Visual Cafe enhances the development of JavaBeans components by providing a single command that immediately shows updates made to JavaBeans components. After modifying the form, source code, or events in your JavaBeans components, use AutoJAR to update the JAR file. You quickly see the results of those modifications in real time as any open projects using those components are automatically updated.

Choosing AutoJAR automatically:

- Saves your source files
- Compiles your source files into class files
- Creates a JAR file with a manifest file
- Adds the JAR to the Component Library

The process of dynamically developing JavaBeans involves preparing a JavaBeans component for dynamic development, modifying code, and updating the Component Library. Each of these steps are explained below.

To prepare a JavaBeans component for dynamic development

- 1 Create a bean using the Bean Wizard.

For more details, see “Bean Wizard” on page 6.

If you choose to use already created source code by adding a source file to the project, you need to include a corresponding *BeanNameBeanInfo.java* file, where *BeanName* is the name of your bean.

2 Choose Project > AutoJAR.

If you want to customize how Visual Cafe creates JAR files, you can do so by editing the `Bundler.properties` file. For more details, please see “Specifying JAR options in the `Bundler.properties` file” on page 14.

Your JavaBean component now has a JAR file, has been added to the Component Library, and all open projects using this component have been updated.

The manifest file generated by AutoJAR includes the following attributes for the class files when the files that meet the following pattern are present, where *BeanName* is the name of your JavaBeans component:

Pattern	Attribute:Value
<i>BeanName</i> .Java	Java-Bean:True
<i>BeanName</i> BeanInfo.Java	Design-Time-Only:True

If *BeanName*Customizer.Java and *BeanName*Editor.Java are present in the project they will also be assigned the attribute of Design-Time-Only:True.

Note: If the name of your JavaBean does not conform to the above file pattern, use the JAR Packager to make modifications. Make sure you select Add to Library so your JavaBean component is added to the Component Library.

3 Open a different project and add your new JavaBeans component(s).

Now, if you later modify the code of your JavaBean component, you can dynamically update it in the Component Library and in all projects where it is used.

Modifying the code

In order to dynamically update your JavaBeans components, before making any modifications to the form, source code, or events of the project containing the JavaBean component you should have run AutoJAR on the JavaBean component.

If the name of your JavaBean doesn't conform to the *BeanName*BeanInfo.java pattern (where *BeanName* is the name of

your JavaBeans component), use the JAR packager to modify files. The JAR Packager creates a JAR file with a manifest file and, if Add to Library is selected, adds the JAR to the Component Library. See “JAR Packager” on page 8 for more details.

To update the Component Library

Choose Project > AutoJAR to update the JavaBeans component(s) in the Component Library and all instances of the component in any open projects.

If your project is not open at the time you run AutoJAR, all instances of your components are updated the next time you open that project.

Note: If you replace a JAR file added to the Component Library by drag and drop from Windows Explorer or by choosing Insert > Component into Library, all instances of components in the JAR that are in any open projects are automatically updated.

Just-in-Time Compiler enhancement

Projects now run even faster than before with the 3.0 version of the Just-in-Time Compiler (JIT). When you deploy your applets and applications, your programs will run up to three times faster than before— faster than any other available JIT.

JFC Swing component support

Visual Cafe includes basic support for Swing components. This includes:

- Support for JFC Swing components version 1.0.1
- Proper code generation for changes made in the Visual Cafe visual environment
- JAR Packager support for pluggable look and feel
- Interaction Wizard support for Swing events
- Custom Property Editors for some Swing properties

For more information about Swing components, please see Sun Microsystems's website at <http://java.sun.com>.

JDK 1.1.5 support

Visual Cafe supports JDK 1.1.5 in our compiler, parser, editor, debugger, class browser and other tools. See the Sun Microsystems website (<http://java.sun.com>) for more information regarding JDK 1.1.5.

Visual Cafe Open APIs

Symantec's Visual Cafe Open APIs offer a broad range of functionality. In general, the Open APIs are designed to give third-party vendors flexible ways to integrate their tools with Visual Cafe without imposing any unnecessary restrictions. For technologies requiring a native interface, third parties can use any Java, C, C++, or Pascal to use the APIs. For more details, please see the Open API documentation, available at <http://cafe.symantec.com>.

Here are some features of the Open APIs below:

Project Templates API

The Project Templates API offers vendors the ability to register and add new project templates to Visual Cafe, and supports both dynamic (interactive wizard) and static project-template creation.

Source Code Manipulation and Integration API

The Source Code Manipulation and Integration API provides a way for third-party software developers to ensure that interactive source code changes can be made with Visual Cafe, allowing for live integration with source code editing and modeling tools. Vendors can use this API to add their own classes and components to Visual Cafe and ensure that when a user is working with advanced tools, changes made in the source code are reflected across all files.

Symantec BeanInfo Extension APIs

The Symantec BeanInfo Extension APIs provide visual development environments with additional information for the placement and attributes of JavaBeans within a development environment, and to provide code generation between objects within our environment.

Menu Hooks API

The Menu Hooks API provides a way to add menu items to the Visual Cafe menu bar.

Project Windows Extension API

The Project Windows Extension API provides a way to add columns to a Project Window, notify owners or users of specific events, and customize the Project window with their own contextual help messages.

CASE API

The CASE API provides a way for third parties to receive notification of project-level events, query, and set component property values, thus seamlessly integrating developers' use of CASE tools with Visual Cafe.

Version Control API

The Version Control API allows for the implementation of version control systems user interfaces into Visual Cafe, allowing end users to install and work with version-control systems on a per-project basis.

Data Binding API

The Data Binding API provides component developers with the ability to allow their users of components to develop the exchange between Java UIs and databases without editing code, and offers a convenient way to bind visual components to databases.

Browser debugging

Applets and applications may run differently in different Web browser Virtual Machines (VMs), so you can save time and effort by debugging in a browser to preview these differences. With Visual Cafe Professional or Database Development editions, you can debug your applets and applications while running them in either Internet Explorer or Netscape Navigator.

Another advantage of running your programs in the debugger is that you can make use of classes that are only supported by that particular web

browser's VM. For browser debugging to work, you must have either version 4.04 of Netscape Navigator (with the JDK 1.1 PR3 or newer patch), or Internet Explorer 4.0 or higher.

To turn on debugging in a web browser

- 1 Activate the Project window of the project you want to work with.
- 2 Choose Project > Options.
- 3 In the Project Options dialog box, click the Debugger tab.
- 4 Choose General in the Category field.
- 5 Select either Netscape Navigator or Microsoft Internet Explorer/Jview.
The default is to debug using Sun Java VM (the Visual Cafe Applet Viewer).
- 6 Choose Project > Run in Debugger or Project > Step Into to start your debugging session in the selected web browser.

Caveats

- Netscape can only debug applets. If you change the project type to Java application, then you will notice that Web browser debugging option will be set back to Sun Java VM and the Netscape Navigator option is unavailable. Internet Explorer supports both applets and applications.
- If Visual Cafe cannot detect the supported version of either web browser then the web browser option is unavailable.
- If you save a project that has the debug in Netscape Navigator or Microsoft Internet Explorer/Jview project option selected, and then for some reason you remove Netscape Navigator or Microsoft Internet Explorer and try to debug your project, you will get a message stating that the browser is not installed and that the Sun Java VM will be used instead. This dialog box has a checkbox that allows you to permanently change the option for the project to Sun Java VM to prevent the message from appearing in the future.
- The following options are not allowed while debugging in Netscape and Internet Explorer: incremental debugging, expression evaluation, and remote debugging.

To allow debugging in Netscape Navigator

- 1 Install Communicator 4.04.

- 2 Install the Netscape JDK 1.1 Support Patch for 4.03 or greater.
- 3 Change to your current Communicator user directory, and edit your `prefs.js` file to allow local applet access to the hard disk. In `...\netscape\user\username\prefs.js` add the following text:

```
user_pref("signed.applets.low_security_for_local_classes", true);
```

This allows Netscape Communicator to access the Visual Cafe applet on your hard disk.

Enabling and disabling RAD

Using Visual Cafe Professional or Database Development editions, you can turn off the visual environment for a file. Rapid Application Development, or RAD, refers to the process of creating your programs in a visual development environment. You can specify in your project options that new files you add to the project have RAD enabled or disabled. You can also turn RAD on and off for an individual file.

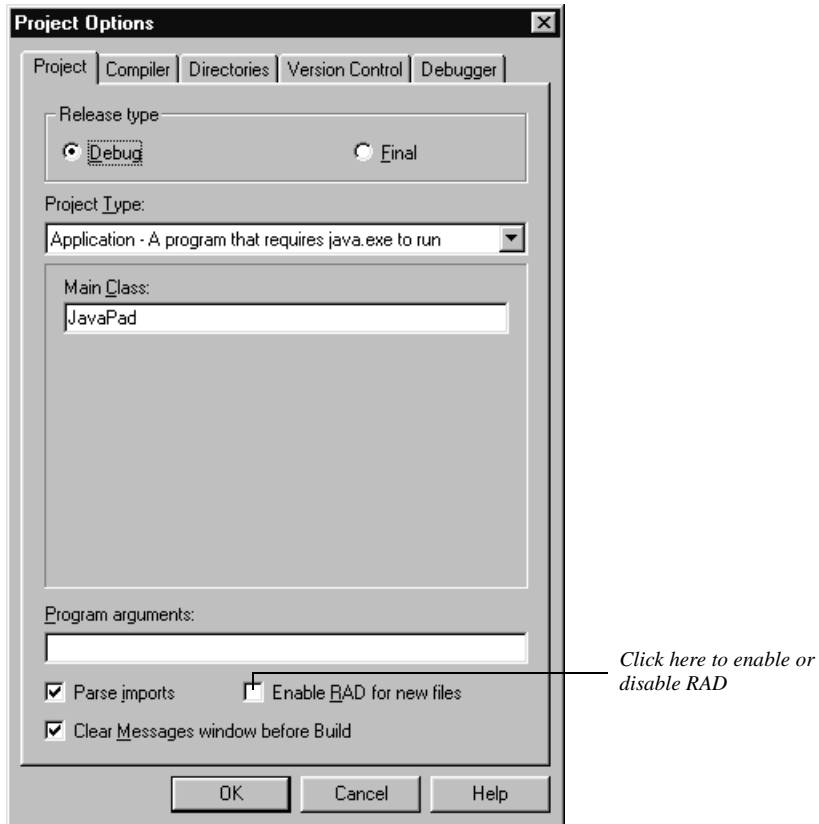
By default, RAD is turned on. This means that components appear in the Objects view of the Project window and you can design your forms in the Form Designer. You might want to turn RAD off if, for example, you do not want Visual Cafe to generate code for you. When RAD is turned off, components disappear from the Objects view and you cannot use the Form Designer with the file.

Caution: If RAD is off and you turn it on, Visual Cafe might change your code in order to parse it into its visual environment.

To enable or disable RAD for new files

- 1 Activate the Project window of the project you want to work with.
- 2 Choose Project > Options.

The Project Options dialog box displays, for example:



- 3 Click the Project tab.
- 4 Select Enable RAD For New Files to specify that new files have RAD enabled. Or deselect it to disable RAD for new files.
- 5 Click OK.

The change takes effect immediately for new files.

To enable or disable RAD for an existing file in a project

- 1 In the Project window, click the Files tab.
- 2 Right-click a file, then choose Stop RAD or Start RAD.

To enable or disable RAD when saving a source file

- 1 While within the source window, choose File > Save As.

The Save As dialog box displays.

- 2 Select or deselect Enable RAD, as desired.

Support for JDBC

This version of the Visual Cafe for Java Database Development Edition extends support for data access through JDBC (the Java database connectivity API). You can take advantage of JDBC through the following features:

- JDBC Data Access Components, plus Mediators
- Editing tools for URL, column, filter, and sort order JDBC elements
- The Master/Detail capabilities of JDBC

Preliminaries

dbANYWHERE is a middleware tool that manages connections between a Java applet or application and one or more databases. It enables you to write database-independent Java applets and applications.

The dbANYWHERE server is required

At run time, you can execute database-aware Java applets and applications with various middleware tools, including dbANYWHERE and the JDBC-ODBC bridge, this release of Visual Cafe requires that a dbANYWHERE server be running at development time when you are developing database-aware applets and applications, no matter what middleware you use at run time.

Minor changes to functionality

To support JDBC features, some minor changes have been made to the functionality of Visual Cafe.

- A dbANYWHERE server upgrade is provided
The dbANYWHERE server and the Java classes used by dbANYWHERE were upgraded after release 2.0. This release of Visual Cafe requires that you be running the upgraded your dbANYWHERE server.

- Legacy projects are supported

The open data-binding architecture implemented for JDBC is also used for applications and applets that use the dbANYWHERE API. For this reason, some of the dbAWARE component properties have changed.

However, you can edit and compile applications and applets that contain dbAWARE components and were created with Visual Cafe releases of 2.0 and earlier. When you open an “old” project, Visual Cafe automatically converts the code to the new dbANYWHERE API implementation.

Note: When a project has been converted to the new dbANYWHERE implementation, it cannot be manipulated by older versions of Visual Cafe.

Specifying the type of code generation

You have the option of generating code for the dbANYWHERE API or for the JDBC API.

To specify the type of database connection code to generate

- 1 On the Tools menu, Select Environment Options to display the Environment Options dialog box.
- 2 Select the Database tab.
- 3 In the Code Generation group, specify the type of code to generate by selecting the corresponding radio button.

JDBC components

The component palette has a new tab entitled JDBC. This tab contains components that support the JDBC API. They are visible only at design time. When you use JDBC components to set up a database connection for a form, you use the visual components on the dbAWARE tab to add database-aware fields to the forms. (The dbAWARE visual components can be used with both the JDBC API and the dbANYWHERE API.)

- **QueryNavigator**

The QueryNavigator JavaBean component manages the set of records retrieved from the database. Dragging a Data Table item from dbNAVIGATOR to a form in a project creates a top-level

RecordDefinition object and a QueryNavigator object. The QueryNavigator object is contained by the form.

- **RecordDefinition**

You use the RecordDefinition JavaBean component to define and access a row set of data in a database table. The RecordDefinition is not contained by a specific form or applet; rather it is contained by the project. Therefore, it can be used by multiple forms. Once you have defined a RecordDefinition JavaBean, you might drag it to the Component Library.

- **JdbcConnection**

The JdbcConnection JavaBean component represents a connection to a specific data source. It has properties for establishing connection parameters in accordance with JDBC.

- **ConnectionManager**

The ConnectionManager JavaBean component manages and logically contains the JdbcConnection JavaBean components for a project. Dragging a DataSource item from the dbNAVIGATOR window to a project adds a ConnectionManager object and a JdbcConnection object to the project. There is one ConnectionManager object per project. You can drag a ConnectionManager object to the Component Library.

- **Mediator**

The Mediator JavaBean is non-visual component that lets you make a standard component (that is, one that is not dbAWARE) behave like a dbAWARE component. It serves as a bridge between the component and the QueryNavigator (JDBC) or RelationView (dbANYWHERE) components. It is found on the JCBC tab of the Component Palette, but is applicable to both dbANYWHERE and JDBC APIs.

For more information on the new JDBC components, see the Online Help.

For information on using a mediator to connect a non-dbAWARE component to a database column, see the Online Help.

Editing tools for JDBC elements

Visual Cafe now provides graphical tools for editing the following properties of JDBC components:

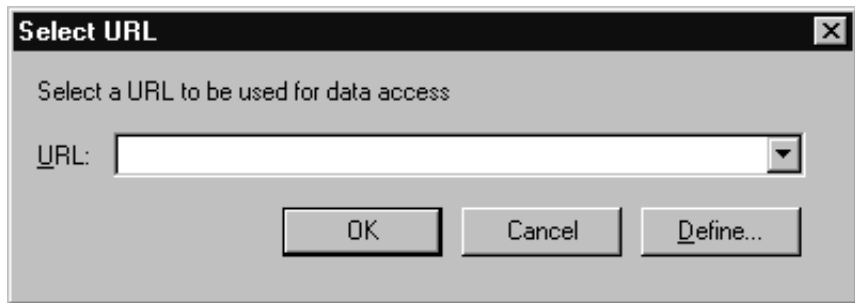
- The URL for the JdbcConnection object

- The URL string
- The SQL properties

Visual Cafe has also added new functionality to the Create dbAWARE Project Wizard and the Add Table Wizard to enable you to create a form that displays your table as a grid or as individual field components.

Establishing the URL for the JdbcConnection object

To display the Select URL dialog box, in the Property List for the JdbcConnection component, in the value cell of the URL property, click the ellipsis (...) button.



To edit a URL

- 1 Select a previously-recorded URL from the drop-down list. If the list is empty, or the URL you want is not shown in the list, type the URL in the list.

If you need help in creating the URL, click the Define button.

Editing the URL string

When you are using the Select URL dialog box, and need help editing or creating the URL, clicking the Define button causes the URL editor dialog to appear. The URL Editor helps you to construct the URL string.

To construct the URL

- 1 The Protocol field displays the protocol that is used in the URL, and can be dbANYWHERE or jdbc. This is dependent on the setting of the
- 2 Select one of the two subprotocols offered by the drop-down list (dbaw or odbc). You can also specify a 3rd-party subprotocol by typing it in.
- 3 If the subprotocol selected is odbc, the SubName field is enabled, and you can type the remaining substring of the URL in this field.
- 4 If the subprotocol selected is dbaw, the SubName field is disabled and the SubName group is enabled. This group enables you to construct the URL without having to type it in.
 - a Select a dbANYWHERE server from the dbANYWHERE Server Name drop-down list.

- b In the Host Name field, specify the host name or IP Address of the dbANYWHERE server you want to use.
- c In the Port Number field, specify the port number of the dbANYWHERE server.
- d Press the >> button to populate the Data Sources list from the specified dbANYWHERE server.
- e From the DataSources list, select a data source for the currently selected dbANYWHERE server.

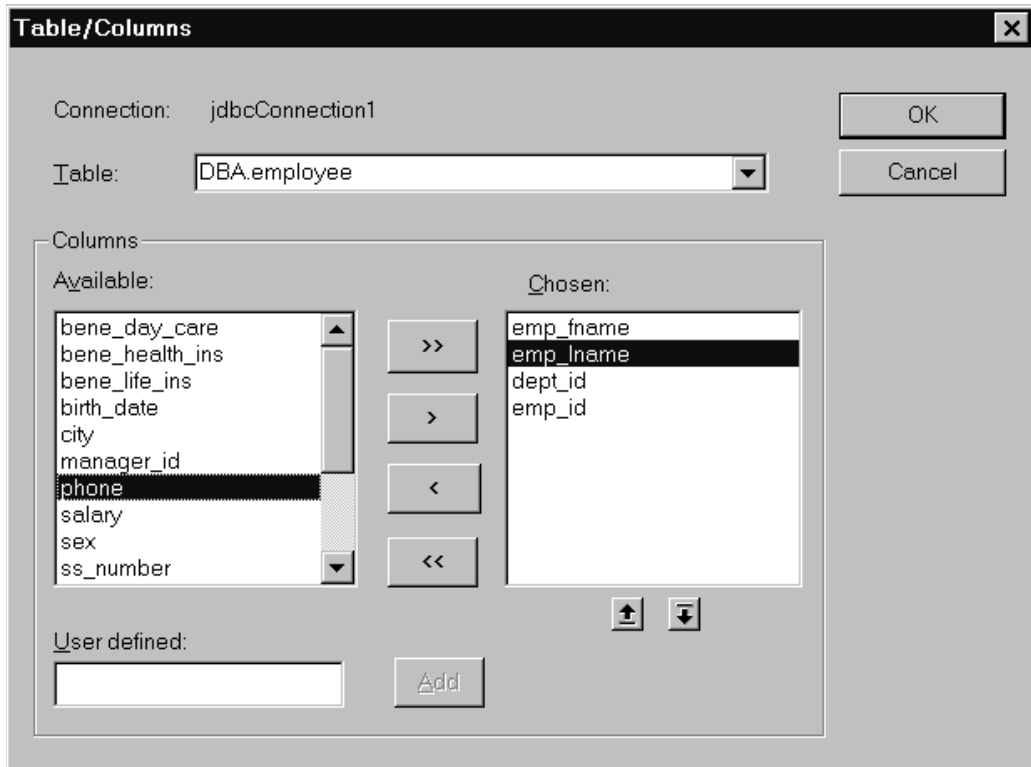
Editing SQL Properties

Visual Cafe has added new functionality that enables you to graphically build the SQL statements that define the set of data used in your Java applet or application.

Specifying the table and columns to use

The Table/Columns Property Editor helps you to specify the tables and columns you want to use. To display the Table/Columns dialog box, from the Property List of the RecordDefinition component, in the value cell of the

Table property, click the ellipsis (...) button. The Table/Columns dialog box appears:



The Connection field shows the name of the JdbcConnection object.

To specify the tables and columns you want to use

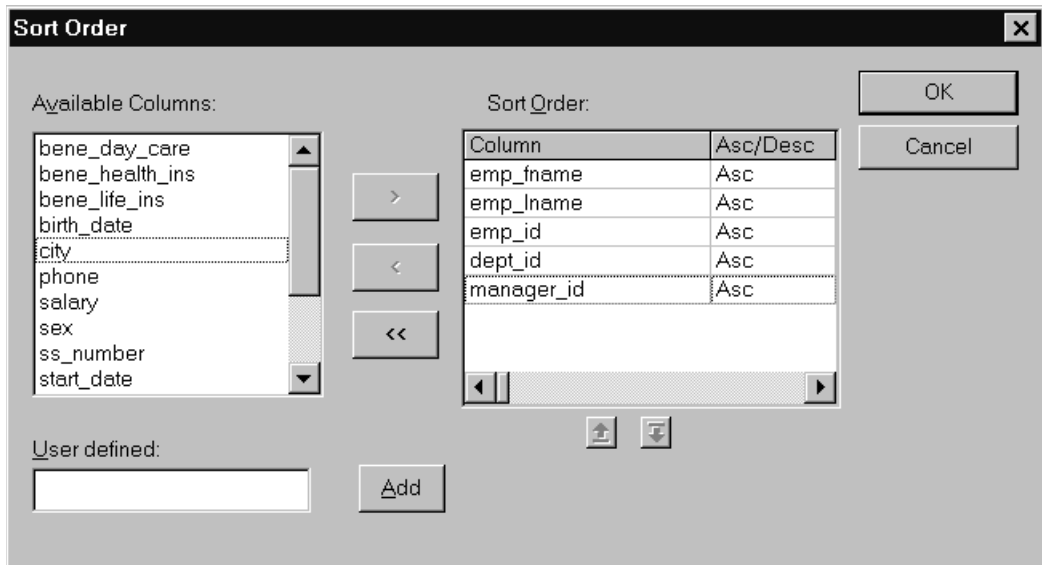
- 1 Select a table from the Table drop-down list. By default, all columns are chosen, as indicated by <ALL> in the Chosen list.
- 2 From the Available list in the Columns group, you select the names of the columns you wish to include in the query. Each time you choose a name, it is moved to the Chosen list.
 - To choose all columns, click the >> button.
 - To choose one column, select its name, then click >.
 - To choose some but not all columns click one name, then shift-click additional names.

- 3 You can also deselect columns (move them from the Chosen list back to the Available list).
 - To deselect one column, select its name in the Chosen list and click the < button.
 - To deselect all the chosen columns, click the << button.
 - To deselect some but not all chosen columns, click one column name, shift-click the rest, then click the < button.
- 4 To change the sequence of a column name in the Chosen list, select a column name and click the up arrow (shown) or the down arrow as needed.
- 5 To add a non-database column, enter a name in the User-defined field and click Add. For example, you might enter an expression like `dept_id > 4000`.



Specifying sort order

You can use the Sort Order Property Editor to specify the sort order for database records.



To specify the sort order for database records

- 1 To display the Sort Order dialog box, from the Property List for the QueryNavigator component, in the value cell of the Sort Order property, click the ellipsis (...) button.
- 2 Select any of the column names in the Available Columns display that you want to sort on.
- 3 Click the > button to move the selected names into the Sort Order display.

If you need to remove any of the fields, use the < and << buttons to move one or all names from the Sort Order display back to the Available Columns display.

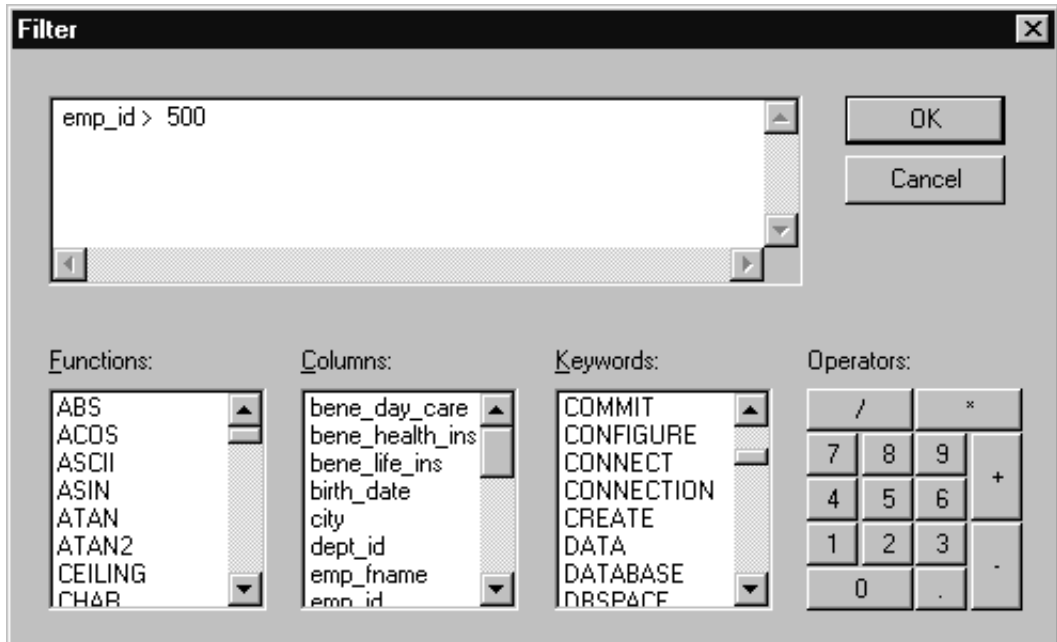
- 4 The Asc/Dsc value for each column determines whether the column will be sorted in ascending nor descending order. All items in the Sort Order display default to ascending sort order. To change the value, click the appropriate cell and choose the desired value from the drop-down list.



- 5 To change the sorting priority of a column name in the Sort Order list, select the column name and click the up arrow (shown) or the down arrow as needed to reposition it.
- 6 To enter any arbitrary column name or expression as part of the sorting (not just those in the Available Columns list), type it in the User Defined field, then click Add. For example, you might enter an expression like `salary + bonus`.

Specifying a filter

You can specify a filter for database records, to constrain the records returned during a query by using the Filter Property Editor.



You can edit the string in the usual fashion.

To construct the string

- 1 To display the Filter Property dialog box, from the Property List for the QueryNavigator component, in the value cell of the Filter property, click the ellipsis (...) button.
- 2 The main text area in the Filter Property Editor displays the filter string. You can type the entire string or use the predefined functions, keywords, and operators to construct the string.
- 3 Use the Functions, Columns, and Keywords lists and the Operators group to select the elements (or type the string manually).
 - The Functions list displays the function names which can be applied to the columns.
 - The Columns list shows the columns of the table specified in the RecordDefinition object.

- The Keywords list displays SQL keywords for you to add to the filter string.
 - The buttons of the Operators group enable you to add operators to the filter string.
- 4 To exit the Filter Property editor without defining the filter string, click Cancel.
 - 5 When you have constructed the desired string, click OK.

Master-detail capabilities

Master-detail support, in its simplest form, is the ability to display and manipulate database records that are related to each other (typically through foreign keys). Associated with a given master record, there is a limited set of related detail records. When the user scrolls a master record, the application must display a set of detail records based on the new master record's values.

Master-detail provides the ability to handle these data restrictions and data filters automatically.

When you need to have a third level to the hierarchy, you can create a grand-detail record that is related to a detail record.

You can write code by hand to use the JDBC API for master-detail, but the Add Table Wizard enables you to erect a framework for master-detail that you can then customize.

In the JDBC API, master-detail joins are defined at the detail record, and use the same properties and join dialogs used with the dbANYWHERE API. The basic property set necessary to establish a master-detail relationship is:

- Master navigator identifier
- Master column identifier
- Detail column identifier
- Binary operator (optional)

Master-only data buffering

An application retrieves a set of master records by making a query to the database through the API. The set of master records returned is called the *master result set*. The master result set is buffered to allow forward and backward scrolling of data. To reduce memory impact on the client, and to ensure fresh data, the detail result sets are not buffered. They are resolved

(discarded or saved), and a new detail result set is queried corresponding to the new master record.

Choice of save methods

The JDBC API offers three possible save methods for you to use:

<code>Save ()</code>	affects the current row and all its detail rows
<code>SaveAll ()</code>	affects all the rows in the current view and all related detail rows
<code>SaveAllLevels ()</code>	begins at the topmost level and affects all rows of all related views

The following table shows the extent of a save operation in various situations:

	<code>Save ()</code>	<code>SaveAll ()</code>	<code>SaveAllLevels ()</code>
Parent rows saved?	None	None	All
Parent refreshed?	No	No	Yes
Current view rows saved?	Current row	All	All
Current view refreshed?	No	Yes	Yes
Detail rows saved?	All	All	All
Detail refreshed?	Yes	Yes	Yes

Handling changed detail records

If the user makes changes in the detail result set, and then scrolls the master record, the system displays a dialog box asking the user whether to save the changes, discard the changes, or cancel the scroll operation. If the user chooses to save the changes, then all changed (dirty) records will be saved for all result sets in the relationship, including the master. That is, the dialog issues a call to save records with the `SaveAllLevels ()` method. This is to ensure data integrity.

Data manipulation

When the user performs a save operation, all Data Manipulation Language (DML) operations are performed in a single transaction if possible. The DML operations are delete, insert, and update.

Saving changes

The system performs data changes by traversing a tree of related result sets. In order to save any changes made during a master/detail operation, the system does the deletes in the detail records, then in the related master record.

The order of traversal for inserts and updates is the from the top down; for each master record, the inserts or deletes are performed first, then those of the related detail records.

Inserting new records in the master

When the user positions the master view on a new, unsaved row, the related detail views will not accept data. The user must first save new master row before the details will accept data. The details will not allow the user to insert or scroll records; attempting to do any of these actions will result in an exception: "Parent record has not been committed. Data changes are not permitted at this time.

Transactions

Each time the user causes a DML operation to be issued to a database, the system first sends a begin transaction request to the database. This is to ensure that all calls to that particular database for the save operation occur as a unit. Once all the DML operations have been successfully issued to the particular databases involved, then all the pending transactions are committed. If any of the operations failed, then all transactions are rolled back.

All DML operations issued to the same database are performed within a single transaction if possible.

When a master-detail relationship spans more than one database, there is a scenario in which DML operations could be committed on one database and not the other(s). This situation can occur because one database supports transactions, but another does not. It can also occur if one

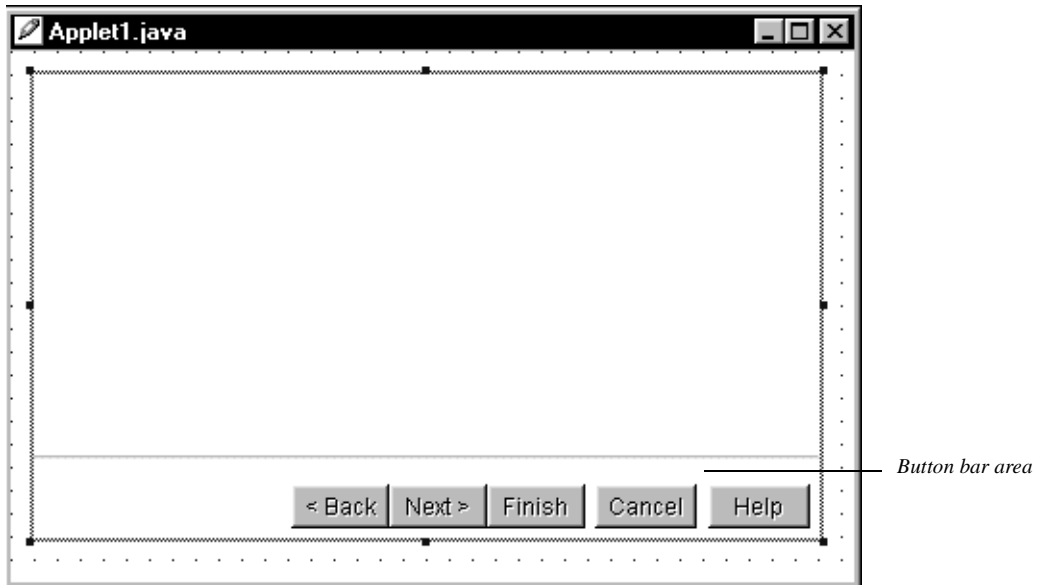
database fails after the DML operations have occurred, but before the commit can be performed.

Wizard component

Use the Wizard component to help you create a “wizard dialog” in your applet or application. A wizard prompts a user for information using a series of pages, helping the user to perform an otherwise complex operation in simple steps. You can drop any type of component on each page of the wizard.

To create a wizard

- 1 Create a new Frame or Dialog (typically a Dialog).
- 2 Drop a Wizard component onto the Dialog. The wizard component displays, as shown below:



- 3 Drop components onto the Wizard. Use a Panel to contain and organize more than one component on a page.
- 4 To create additional pages, drop components onto the button bar area. A new page is created with the newly-added component, and becomes the active page. The state of the Back and Next

buttons are updated to reflect the relative position of the active page.

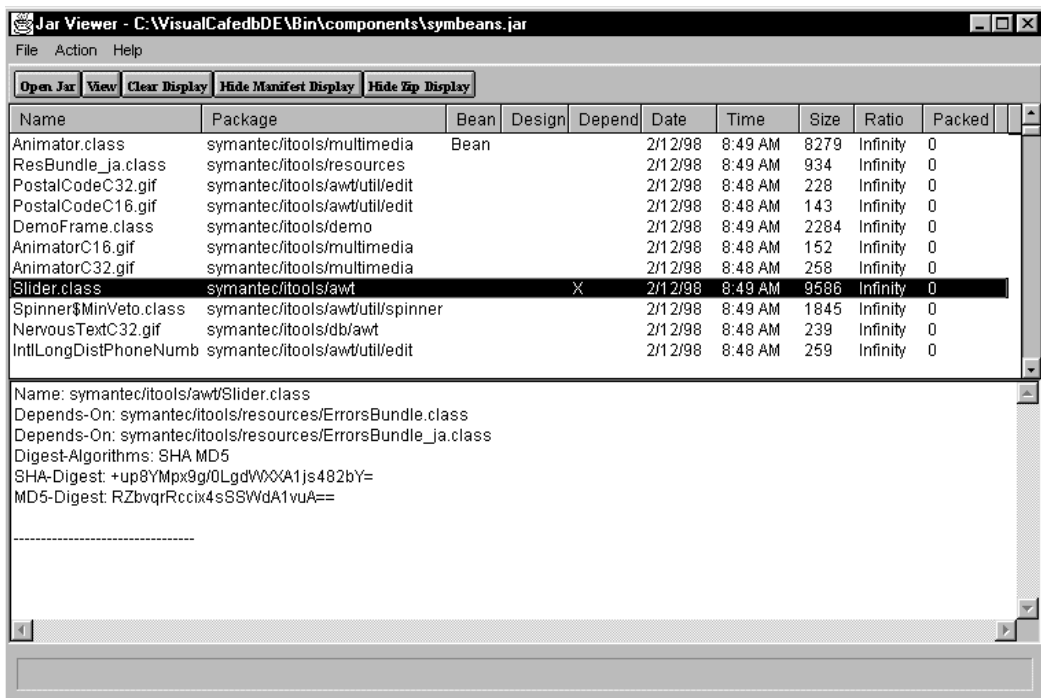
- 5 Pages can be moved within and removed from the Wizard by deleting or reordering them in the Project window.

At run-time, the Wizard's behavior can be extensively customized. See the Online Help and API documentation for more details. To access these, choose Help > Help Contents, click the Index tab, and type wizard in the index field. The Wizard component screen displays, with links to properties, methods, and API information.

JAR Viewer

The JAR Viewer enables you to easily view the contents of a JAR file, just as you would a ZIP file. You can sort by date, type, and filename. Use the JAR viewer to open an existing JAR or ZIP file, or to select a file in a JAR or ZIP file. You can also use the JAR Viewer to view the JAR's manifest file.

Here is an example of what the JAR Viewer looks like:



To view the contents of a JAR

- 1 From the Tools menu, choose JAR Viewer.
The JAR Viewer displays.
- 2 Click Open JAR, and browse to the desired JAR file.
The JAR Viewer opens the JAR file.

Within the JAR Viewer you can:

- Sort the JAR contents by clicking the title of the column. Click the column title again to reverse the sort order.
- View the manifest of a class file in the JAR by double-clicking the selected file or choosing View.
- View the manifest of multiple files by selecting the files and then clicking View.
- Hide the manifest details by clicking Hide Manifest Display. Click Show Manifest Display to reveal the manifest details.
- Hide the zip/jar details by clicking Hide Zip Display. Click Show Zip Display to reveal the jar/zip details.

Conversion from no layout to GridBagLayout

Java provides layout managers that help you arrange components inside a container (a form or a panel) so that the appearance adjusts automatically across platforms and Web browsers, and at varied screen sizes and resolutions. When you do not use a layout manager, components are placed by exact pixel positions in a layout. A layout manager arranges components proportionately.

For more information on using layout managers, please see the *User's Guide* or the Online Help.

When you change from not using a layout manager in the Form Designer to using the GridBagLayout layout manager, your components will automatically resize proportionate to the form size. This saves you from having to manually reset each component's size and position. GridBagLayout supports this proportional resizing at both design- and run-time.

GridBagLayout establishes a framework for a form, separating areas of the layout into cells of a grid. This is similar to GridLayout, except that with GridBagLayout you can have a component occupy more than one cell.

To convert from no layout to GridBagLayout

- 1 Arrange and size your components as desired.
- 2 Choose GridBagLayout for the Layout property of a form or panel.
The Form Designer immediately converts the layout by setting the Grid Bag Constraints of each component to resemble the placement and size of components from the null layout.
- 3 If you want to rearrange or add components after converting to GridBagLayout, we recommend switching back to no layout, making the desired changes, and then switching back to GridBagLayout.

When you change to GridBagLayout, Visual Cafe establishes the Grid Bag Constraints properties that specify each component's place in the layout. Think of these properties as "suggestions"—to change these properties, click the component to display the Property Inspector, and make changes from there.

While the cursor is in any Grid Bag Constraints property field, press F1 to get a description of the property. Insets specify how much space to leave between the borders of a component and its display area.

- 4 Test your layout by running it at different form sizes and resolutions. For example, you can resize the form when you run it from Visual Cafe.

JavaBean customizer support

Visual Cafe allows you to call an available customizer to configure the characteristics of a JavaBean. Use a customizer when you want more guidance for configuring a JavaBean's behavior than the Property List provides.

If Visual Cafe detects a `bd.SetValue("hidden-state", Boolean.True)` statement in the BeanInfo file of a JavaBean, you will not be able to access the customizer. If a JavaBean has a hidden state set to true, it may be designed to work with serialization, though you can still edit the properties exposed in the Property List.

If a JavaBean is a hidden state JavaBean and has not set this information in the BeanInfo file, any changes made in the customizer which do not correlate to an item on the Property List will be lost. Thus, if the JavaBean's hidden state is set to false, or if the hidden state is not set at all, the changes in the customizer will take effect.

Note: Customizers will vary since each JavaBean has a unique customizer created by the JavaBean developer. Not all JavaBeans may have a customizer.

To access a JavaBean's customizer

- 1 Right click a component in the Form Designer.
- 2 Choose Customizer from the context menu.

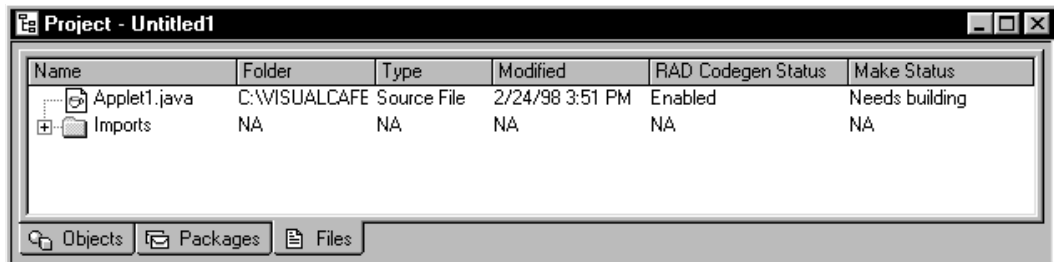
If the Customizer menu item is not available, the component has not specified an available customizer.

- 3 Make any necessary changes in the customizer.

After finishing the customizer, the appropriate code is generated.

Sorting files in the Files view

In Visual Cafe Professional and Database Development editions, you can sort files in the Files view of the Project window by clicking a heading button, as shown here:



Click Name, Folder, Type, Modified, RAD Codegen Status, or Make Status to sort by that attribute. Click again to reverse the sort order.

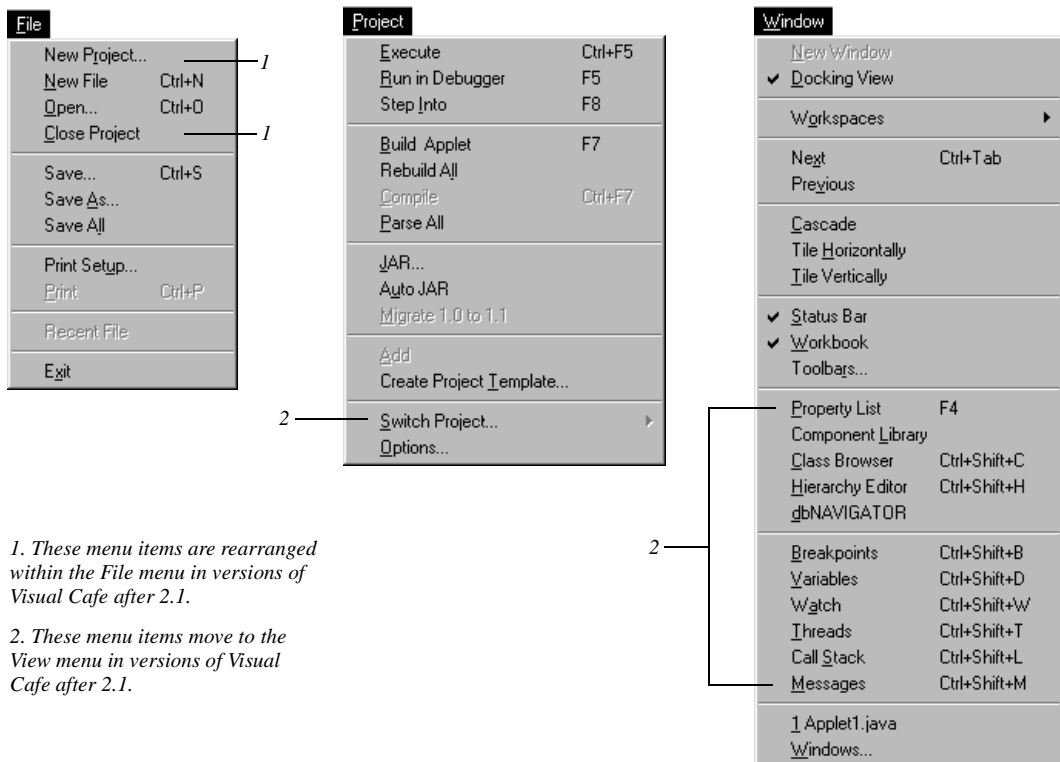
If you do not see all the headings, resize the Project Window.

Using Visual Cafe 2.1 menus

Some menu items were rearranged in Visual Cafe after the 2.1 release. Visual Cafe version 2.1 and previous versions did not have a View menu. Some items from the Window menu were moved to the View menu (menu items that opened a specific type of window, such as the Component Library menu item). Some items on the File menu were rearranged or added, and also an item on the Project menu moved to the new View menu.

You can specify that you want to use the old menu system, if you prefer that menu structure. You might want to do this if, for example, you have macro scripts that rely on the older version of the Window menu.

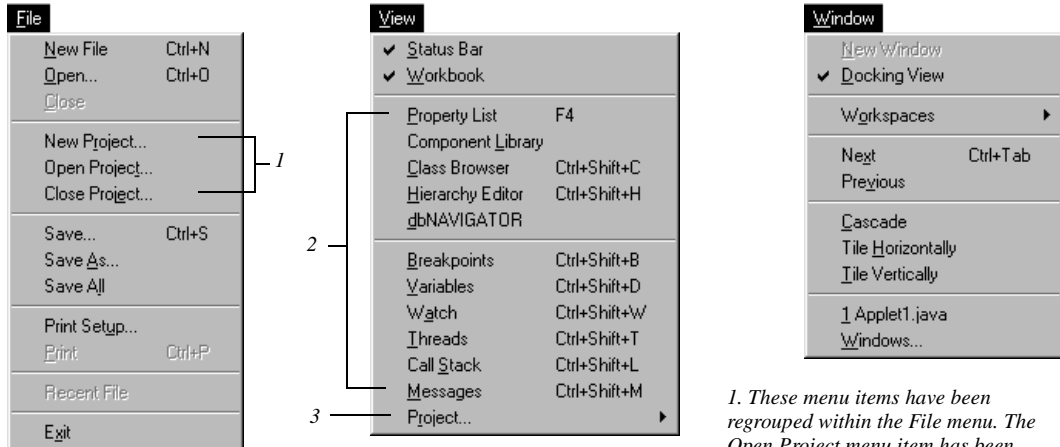
Here is an example of the version 2.1 menus (only the File, Project, and Window menus are affected):



1. These menu items are rearranged within the File menu in versions of Visual Cafe after 2.1.

2. These menu items move to the View menu in versions of Visual Cafe after 2.1.

Here is an example of what the menus look like in versions after 2.1:



1. These menu items have been regrouped within the File menu. The Open Project menu item has been added.

2. These menu items have moved from the Window menu.

3. This menu item has moved from the Project menu.

To use Visual Cafe 2.1 menus

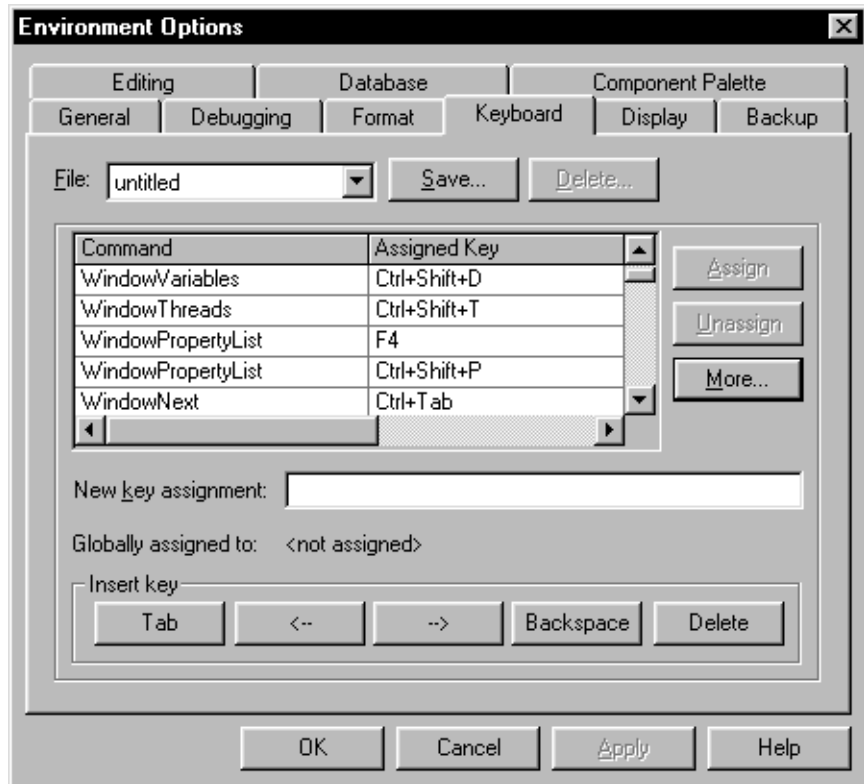
- 1 Choose Tools > Environment Options > General tab.
- 2 If you want to use an earlier version of Visual Cafe menus, select Use 2.1 Menus. Otherwise, deselect it.

J++ keyboard shortcuts

You can now choose to use Visual J++ keyboard shortcuts in addition to those of other development environments. You can define a custom keystroke sequence for many Visual Cafe editing operations and macros.

For more information on defining keyboard shortcuts in Visual Cafe, please see the *User's Guide* or Online Help.

Here is an example of the Keyboard pane in the Environment Options dialog box:



To choose Visual J++ as your set of keyboard commands

- 1 Choose Tools > Environment Options, and then click the Keyboard tab.
- 2 Choose the VJ key file from the File drop-down menu.
The Visual J++ key file is loaded.

To add a new keyboard command to the Visual J++ set of keyboard commands

- 1 From the Tools menu, choose Environment Options, and then click the Keyboard tab.
- 2 Choose the VJ key file from the File drop-down menu.
- 3 From the command list, select a command.

If an item has more than one key assignment, the item is listed separately for each key assignment.

Tip: Click the column header to sort the list by command or key assignment, and click again to reverse the sort order.

- 4 To specify a key assignment, click in the New Key Assignment field and press the key sequence. Click the buttons in the Insert areas to insert a Tab, left arrow, right arrow, Backspace, or Delete keystroke.

If the value that you enter is already mapped to a command or macro, that command name appears in the Globally Assigned To area.

Tip: A command can have multiple key assignments. If the command already has a key assignment and you specify another assignment, the command now has two separate assignments.

- 5 Assign the key sequence to the command by clicking Assign. The assignments are automatically saved in an untitled file.
- 6 To save the settings to a file, click Save.
In the dialog box, specify the name of a new or existing key file. Saving to a file allows you to reload or distribute the key assignments. Key files are stored in the \Bin\Keys directory.

To copy the Visual J++ command assignment list as text

- 1 From the Tools menu, choose Environment Options, and then click the Keyboard tab.
- 2 Choose the VJ key file from the drop-down menu to load the Visual J++ set of keyboard commands.
- 3 Right-click in the Command list and select Copy All.
You can paste the list into another window, such as the Source window.

Java virtual font classes support

Visual Cafe supports the Java virtual font classes. Use these font classes to ensure cross-platform compatibility. Use Serif, Sans-serif, Monospaced, Dialog, and DialogInput in your programs, rather than a particular typeface such as Times or Helvetica. Virtual font classes are no longer mapped to non-Latin1 classes of fonts.

To select a virtual font class

- 1 Select a component that uses text.
- 2 In the Property List, select the Font Name property.
A list of font classes displays.
- 3 Choose from the list of classes.
Your component's text changes to the appropriate font.

For information on default font properties, see the `font.properties` file in your Visual Cafe \Java\Lib directory.



World Headquarters
Symantec Corporation
10201 Torre Avenue, Cupertino, CA 95014
408-253-9600

World Wide Web site: <http://www.symantec.com>
Java Central site: <http://cafe.symantec.com>
Java newsgroup: comp.lang.java
CompuServe: Sym Dev.Tools