CA-Visual Objects
Version 1.0
Release Notes

November 13, 1994

This file contains important information about CA-Visual Objects Release 1.0. This information is provided as an update to your printed CA-Visual Objects documentation.

Using Write to View This Document

If you enlarge the Write window to its maximum size, this document will be easier to read. To do so, click the Maximize button in the upper right-hand corner of the window. Or, click on the system menu in the upper left-hand corner of the Write window (or press Alt+Spacebar), and then choose the Maximize command.

To move through the document, press Page Up or Page Down or click the arrow at the top or bottom of the scroll bar along the right side of the Write window.

To print the document, choose the Print command from the File menu.

For help using Write, press F1.

To read other on-line documents, choose the Open command from the File menu.

Contents

This document contains information on the following topics:

Read Me First!

Repository Backup

Before installing a new version or build of CA-Visual Objects or re-installing the repository of an existing version, you must export all applications and libraries (yours, not the system's) that you wish to save. Installing the repository creates a new repository containing only the CA-Visual Objects libraries. After the installation is complete, you can import your applications and libraries back into the newly created repository. **If you do not follow this procedure, your applications and libraries will be lost, with no chance of recovery.**

Please note that when you import AEF files from a prior version or build, you may receive a message about a conflict with a file called STD.UDC. You should choose **Keep Existing.**

Internationalization

CA-Visual Objects Release 1.0 does not fully support the Microsoft Windows internationalization standards (as specified from the Windows Control Panel, using the International icon). Additional support for these standards will be available in the very near future.

ANSI vs. OEM Databases

As a Windows product accessing databases built in a DOS environment, CA-Visual Objects supports both the ANSI and OEM character sets to the fullest extent possible. The CA-Visual Objects runtime provides for conversion back and forth between the two character sets, allowing you to maintain your OEM-based databases. However, it is strongly recommended that you migrate your databases to ANSI format, unless you have a compelling reason not to do so--the most compelling being the need to share data with a DOS-based application.

Migrating your files to ANSI format is simple and easy and doing so will make your applications more efficient (because no character translation is necessary when reading and writing the files). Simply insure that the SetAnsi() flag is on (the default) when you create new files, and they will be created in ANSI format. For existing files, just copy to a new file with SetAnsi() on. Note also that ANSI is the default for saving files in the DB Server Editor, so that if you perform a File Export, the file will be saved as ANSI (unless you check the OEM box).

Repository Reindex

The Repository Reindex utility, CAVORIDX.EXE, is not included in your printed documentation. This utility is designed for correcting repository corruption that is due solely to index damage. CAVORIDX.EXE is installed in your main CA-Visual Objects directory and, if you experience repository corruption, you may want to run it before attempting to repair the damage using the Repository Manager (documented in Appendix D of the *IDE User Guide*). **Do not forget to backup your repository before running this utility.**

Documentation Issues

In the "Operating Environment" chapter of the *Programmer's Guide, Volume II*, the table in the Other Files to Distribute section incorrectly lists CTL3D.DLL as a file to distribute if you are using the GUI Classes library. The correct file name is CTL3DV2.DLL.

Since the documentation was printed, the DBClearIndex() function has been updated. Please refer to your online help reference for the most up-to-date information on this function.

Note that, in general, if you find a discrepancy between the printed documentation and the online help, you should rely on the online help to contain the most recent information.

_

Installing over the Pre-Release Version

You should remove any previous version or build of CA-Visual Objects entirely from your system before installing this version. **Do not forget to export your applications and libraries before doing this.** You should also delete the CA-Visual Objects program group from the Windows

Program Manager and remove the directory where CA-Visual Objects was installed. It is especially important that you erase all the old repository files in \CAVO\DATA. The CAVO.INI file should be removed from your Windows directory. It is strongly recommended that you exit Windows after removing the Pre-release version and prior to installing the 1.0 version, otherwise problems may occur when trying to compile programs under 1.0 including a spurious message, *VORC.BAT cannot be found*.

Note, however, that several files installed in your Windows SYSTEM directory may present conflicts during installation due to a minor error in the Pre-Release installation and, under most circumstances, you will want to overwrite what appear to be newer versions of files in this directory with the versions from the installation disks.

In general, if you receive the message "<*FileName*> already exists in target directory and is newer than the same file on the installation disks. Overwrite anyway?" and the file in question resides in the Windows SYSTEM directory, you will want to overwrite the version on your disk. In other words, choose **Yes**.

However, the following files represent exceptions and must, therefore, be investigated further before making a decision:

CTL3DV2.DLL
ODBC.DLL
ODBCINST.DLL
ODBCINST.HLP
TOOLHELP.DLL

For these files, you should switch (Alt+Tab) to the Program Manager, start the File Manager, inspect the file's date and time stamp, and switch back to the installation routine.

If the date and time stamp is 08/12/94, 11:00 am, choose to overwrite the file. A file with this date and time stamp was installed as part of the pre-release of CA-Visual Objects and is, therefore, an older version than the file being installed, despite the newer date and time stamp. In other words, choose **Yes**.

Otherwise, if the file has any date and time stamp other than this, you can assume you have a newer version of the file and should keep the existing file. In other words, choose **No**.

_

Installation Notes

This section contains important information that you may need to properly install CA-Visual Objects.

Installing From a Network Drive

If you have problems when attempting to run the installation program from a network drive, copy the contents of disk 1 to a floppy disk using XCOPY with the /S option and install from the floppy drive. Then, when prompted for the location of the other disks, enter the appropriate network drive and directory.

It is important that the installation procedure run from beginning to end with no problems. Therefore, if your system locks up or suddenly exits from Windows to the DOS prompt during installation, follow the advice given above. Otherwise, you cannot be sure that CA-Visual Objects was installed successfully.

CAVO.INI Settings

Do not set the ExecutablePath and ApplicationsPath entries in the CAVO.INI file to point to the same directory. Doing so will cause problems when importing or running applications. Note that CAVO.INI is installed in your Windows directory. You must also ensure that the ApplicationsPath points to a current version of the CA-Visual Objects Repository, otherwise CA-Visual Objects will fail with an ADAM Initialization message.

Verify the CA-Visual Objects Directory in Your DOS PATH

Before running CA-Visual Objects, verify that C:\CAVO (or the drive and directory you specified during the installation procedure) is included in your DOS PATH because some network operating systems reroute the path during network login.

Remember that anytime you make changes to AUTOEXEC.BAT or CONFIG.SYS, you should reboot your computer for the changes to take effect.

Loading SHARE.EXE in CONFIG.SYS

If you are using Windows version 3.10, it is necessary to load the DOS SHARE.EXE program to install file sharing and locking capabilities on your system. (If you are using Windows version 3.11 or greater, SHARE.EXE is installed automatically.) To do this, include a line similar to the following in your CONFIG.SYS file:

```
INSTALL=C:\DOS\SHARE.EXE /F:4096 /L:500
```

The values you specify for the /F and /L options will depend on your system (see your DOS manual for details).

If VSHARE.EXE has been provided by Microsoft as part of Word for Windows it should be used instead of SHARE.EXE.

Remember that anytime you make changes to AUTOEXEC.BAT or CONFIG.SYS, you should reboot your computer for the changes to take effect.

ODBC Conflicts

You may receive a message during installation about a conflict with ODBC DLLs or Help files. You should choose **Yes** if the file already exists with a more recent date or, if you are installing over the Pre-Release version, refer to Installing over the Pre-Release Version earlier in this document.

CTL3DV2 and ToolHelp Conflicts

You may receive a message during installation about a conflict with CTL3DV2 or ToolHelp DLLs. You should choose **Yes** if the file already exists with a more recent date or, if you are installing over the Pre-Release version, refer to Installing over the Pre-Release Version earlier in this document.

Logitech Mouse Problems

Various problems have been reported with Logitech mice using the serial port. See Other Release 1.0 issues for details.

List of Files Installed

You can verify that all files were installed correctly by referring to the file list in INSTALL.LST.

_

Migration Issues from Pre-Release Version to Release 1.0

If you are upgrading from the CA-Visual Objects pre-release to Release 1.0, you should be aware of the following issues regarding migration:

__Handle() Method Renamed to Handle()

The __Handle() method has been renamed to Handle(). We strongly recommend that you do a global Search and Replace All in your source code to reflect this name change.

New AppWindow:EndWindow() Method Added

The new AppWindow:EndWindow() method should be used instead of Destroy() to get rid of data windows, child app windows, and shell windows.

Standard Program Improvements

The Standard Program has been substantially improved. If you are using the Standard Program source code as the basis for your own application, it is strongly recommended that you create a new application (which, effectively, regenerates the Standard Program portion of the application), export the modules involving your own source code from the old application, and import them into the new application. This way, you will have a new version of the application that uses the new, improved Standard Program shell.

Generated Source Code

The generated source code from all the visual editors (Window Editor, DB Server Editor, and so on) has also been improved. To take advantage of this and bring your generated source code up to date, open the associated entity in the editor, make some fake change (e.g., change something, then change it right back), and save the entity. Note that any customization of the source code that you have made will be lost, so you may want to save the original so that you can recover the changes.

_

Improvements Since Pre-Release Version

CA-Visual Objects has been substantially improved since the pre-release version. This sections lists several, but by no means all, of these improvements.

Compile-Time Method Checking

The compiler now checks for the presence of methods (including accesses and assigns) and generates the warning message, "<*MethodName*> is not a method of class <*ClassName*>: 51441," or "Variable or access <*Name*> in class <*ClassName*> does not exist or is not accessible: 51442," if a particular method or variable name referenced in the source code cannot be found. Although these situations are trapped only as warnings, they generally indicate runtime failure and should be addressed.

Strongly Typed Methods and Defines

The compiler now checks for methods and defines declared using strong typing and generates the warning message "Ignoring type declaration for <*Name*>: 51234." The intent of this warning message is to inform you that data type declarations in method and define declarations are strictly for documentation purposes and have no other effect on the application, either at compile time or at runtime.

DBFCDX and DBFMDX Drivers

These RDDs are now included with CA-Visual Objects, and the DBServer Editor has been enhanced to fully support the multi-order index file paradigm.

User-Defined Toolbar Buttons

The Toolbar:AppendItem() method now fully supports user-defined toolbar buttons.

CAVOWED.INI Improvements

CAVOWED.INI has several improvements, allowing you tighter control over customization of the Window Editor. These include the ability to add user-defined bitmaps to the tool palette. CAVOWED.INI is now fully documented in Appendix C of the *IDE User Guide*.

Calculated or phantom columns

The Window Editor now supports calculated columns through the Block and Block Owner properties.

_GetCmdLine()

The _GetCmdLine() function has been added to the System Library. This function is not documented. It returns a PSZ pointer to the command line parameters used to invoke the application.

_

Other Release 1.0 Issues

This section describes undocumented CA-Visual Objects behaviors that may be unexpected.

FieldSpec Required property not honored for "disconnected" controls

Problem: In a data window, the FieldSpec Required property is not honored for controls that are not bound to a database field.

Workaround: In the method that you use to move information to the actual database field, provide code to satisfy that the control has been filled in by the user.

Close and QueryClose events are not sent to sub-data windows

Problem: The Close and QueryClose events are not sent to sub-data windows and can, therefore, not be driven by corresponding methods defined for the sub-data window.

Workaround: Put the code to execute in case of a Close or QueryClose event from a sub-data window in the Close() or QueryClose() methods of its owner data window. If you prefer to have Close() and QueryClose() methods defined for the sub-data window itself, simply call those methods from the owner data window's corresponding method.

User-defined functions in order key expressions not recognized by DB Server Editor

Problem: Attempting to import/export an index file in which an order key expression is based on a user-defined function will fail.

Workaround: Instead of importing the index file as part of your data server in the DB Server Editor, manually enter the index and order properties into the DB Server Editor. To manually create such an order, use the DBServer:CreateOrder() or DBServer:CreateIndex() methods.

DBServer:RLockVerify() and DBServer:ConcurrencyControl conflicts using DBFBLOB RDD

Problem: When using the DBServer class for databases with BLOB files to represent memo fields (i.e., using the DBFCDX RDD or otherwise using the DBFBLOB as an inherited driver), the CC_OPTIMISTIC setting for the ConcurrencyControl property (and the underlying RLockVerify() method) may not be reliable. In particular, if the **only** changes to the record involve memo fields, RLockVerify() will indicate that it is safe to lock a record when it is not.

Workaround: Use a ConcurrencyControl setting other than CC_OPTIMISTIC when BLOB files are involved.

Descend() not operational in order expressions for OEM files

Problem: The Descend() function, when used in the order key expression of OEM index files, will not reliably produce keys in descending order.

Workaround: If possible, convert to ANSI format. Otherwise, you may be able to recreate the order using the DESCENDING option of the INDEX command.

Shift+F1 not functional for data window sub-forms in browse view

Problem: Shift+F1 is not available for data window sub-forms in browse view to get context sensitive help.

Objects created from DrawObject class and descendants must be explicitly destroyed

Problem: Objects created using the DrawObject class or any of its descendants are not subject to garbage collection.

Workaround: When the application is finished with the object, explicitly destroy it using the Destroy() method.

Resource compilation (RC.EXE) fails due to insufficient disk space

Problem: CA-Visual Objects uses the Microsoft Resource Compiler to compile all resource entities in an application. The resource compilation process may produce a lot of temporary files that take up a large amount of disk space, which will lead to problems if your disk space or the number of available directory entries is limited, as may be the case with a RAM disk.

Workaround: The location of temporary file storage is controlled by the TEMP environment variable. If you are exceeding the available number of directory entries, you can either set the TEMP environment variable to a subdirectory, which lets you create as many files as you want, or increase the number of entries allowed when you set up the RAM disk. If you are running out of actual disk space, you will have to redirect the TEMP environment variable to another location, such as a hard disk with more available space. Note also that it is not advisable to set TEMP to a network drive, since use of this location is intensive and may slow down your system.

Resource compilation (RC.EXE) fails due to conflicts with video and SCSI drivers

Problem: The Microsoft Resource Compiler runs in a DOS box under Windows, and this can conflict with the high memory area on systems using high memory management. Problems have been reported when using CA-Visual Objects on systems with certain video and SCSI (for example, Adaptec) cards that also make use of this area of memory. The problems that occur when there is a conflict can range from mild (brief flickerings or mode changes in the monitor display) to severe (GPFs) and may even cause a system reboot.

Workaround: If you are experiencing problems that you think might be related to high memory conflicts, remove all high memory management from your system configuration, reboot, and see if the problems go away. If so, there are two steps you can take to eliminate the memory conflicts. First, make sure you have the most recent versions of your video or SCSI driver. Then, find out the area of ROM being used by your video or SCSI card (this will be somewhere in the area of C000 to D000) and exclude this area from use by your high memory manager. Consult the documentation produced by the manufacturer of your high memory management system for more specific information on how to do this.

Serial mouse problems

Problem: We have had reports of problems using CA-Visual Objects with a serial mouse (specifically, Logitech).

Workaround: If you experience mouse-related problems (for example, the mouse capability goes away or behaves unpredictably or the application freezes), include the entry OutPutTo=NUL in the [Debug] section of your SYSTEM.INI.

_

CompuServe Facilities

There is a production CompuServe forum (GO VOFORUM) targeted specifically for CA-Visual Objects. Computer Associates will use this forum to communicate information regarding updates, fixes, and so on.

Operating Systems

CA-Visual Objects Release 1.0 currently supports MS-Windows 3.1, 3.10, and 3.11, Windows for Work Groups, Windows for OS/2, and Windows 3.1 running native under OS/2.

At this point, there is only qualified support for Windows NT. Attempts to run CA-Visual Objects under these operating systems may fail with various errors. There is currently no support for Chicago. Problems under NT should be reported on the CompuServe forum.

Redistribution of Software

CA-Visual Objects Release 1.0 comes bundled with several software packages. Computer Associates may or may not allow you, the 3rd-party developer, to redistribute these packages as part of your application. This section describes your redistribution rights for each of the various software packages included with CA-Visual Objects. For a complete list of files associated with each of these products, please refer to the INSTALL.LST file installed in your main CA-Visual Objects directory.

CA-Visual Objects Compiler Runtime DLLs

You have full redistribution rights for these DLLs.

CA-RET

You have redistribution rights for CARETRUN.EXE and CARET.HLP, but **not** for CARET.EXE.

SQL Database Drivers and RDDs

You have full redistribution rights for these drivers.

Standalone WATCOM SQL for Windows

You have **no** redistribution rights for this software; however, please refer to the "Delivering the Power for Your Client/Server Solutions" for information on purchasing the WATCOM SQL Runtime System for Windows.

Microsoft ODBC Administration

You have full redistribution rights for this software.

Microsoft Help Compiler, HotSpot Editor, and Resource Compiler

You have **no** redistribution rights for this software.

The South Seas Adventure Tutorial

The South Seas Adventure is a sample application that is installed as part of the Samples option during the CA-Visual Objects installation process. This section contains brief instructions on setting up and running the South Seas Adventure application. This application is complex, and it contains many useful CA-Visual Objects programming tips and techniques. A written tutorial, designed to work with the South Seas Adventure application, will be sent to you when you send in your completed registration card.

Setting up the Accounting Department Data Source

Using the Microsoft ODBC Administration program located in your Control Panel, set up the ODBC driver for the "Accounting Department" by adding the following configuration:

CA-VO dBASE (*.dbf) driver

Data Source Name: Accounting
Description: Accounting Department
Database Directory: <CAVO dir>\SAMPLES\SSATUTOR
Create Type: Clipper
Lock Compatibility: Clipper
Locking: Record

Importing and Building the Application

If you installed CA-Visual Objects in a location other than C:\CAVO, follow these steps to import and build the South Seas Adventures:

1. From the Application Browser, choose File Import.

2. Select SSA.AEF from your CA-Visual Objects SAMPLES\SSATUTOR\FILES directory.

3. Click OK to import the application, **without** checking the Build Application check box.

4. After the import is complete, change the Path for EXE and DLL files application property (choose the Application Properties menu command or click the Application Properties toolbar button) to point to your SAMPLES\SSATUTOR directory. By default, it will point to C:\CAVO\SAMPLES\SSATUTOR.

5. Click OK.

6. Build the application by clicking the Build toolbar button.

If you installed CA-Visual Objects in the default directory, C:\CAVO, follow these steps to import and build the South Seas Adventures:

1. From the Application Browser, choose File Import.

2. Select SSA.AEF from the C:\CAVO\SAMPLES\SSATUTOR\FILES directory.

3. Check the Build Application check box to automatically build the application after it is imported.

4. Click OK.

Running the Application

From within IDE, execute the application by choosing the Execute toolbar button (or Application Execute menu command).

You can also generate an .EXE file and run it. To generate the .EXE from CA-Visual Objects, choose the Make EXE toolbar button (or Application Make EXE menu command). This will create the program group (if necessary) and the program icon in your Windows Program Manager. You can then run the application from Windows.

Once the application is running, login using the employee entry:

Login	Password
user	trainee

You can then create an entry in the EMPLOYEE file for yourself.

SSA.INI

The initial screen and login can be bypassed by modifying the entries in SSA.INI (in your CA-Visual Objects SAMPLES\SSATUTOR directory) as follows:

```
[Runtime]
Debug=1
OpeningDialog=No
```

Known Problems

This section describes significant issues that we are already aware of in Release 1.0. Included is a brief description of each problem and, if possible, a way to circumvent it.

Please note that if you get a GPF of any level while operating in the IDE (including dynamic execution of an application), you should exit CA-Visual Objects and exit Windows. If you get a GPF while running one of your CA-Visual Objects applications, this practice is also recommended, but not necessarily mandatory.

Toolbar button events not passed to owner window

Problem: Menu command events generated by clicking a toolbar button are not propagated to the owner window, although corresponding events generated from equivalent menu selections are.

Workaround: Code a method owned by the child window with the correct name, which simply calls the same method in the owner window. For example, if the File Open toolbar button for a data window should propagate the event to the owner window's FileOpen() method, write a FileOpen() method for the data window that invokes Owner:FileOpen(). Alternatively, you can handle menu command events for the child window using a custom-coded MenuCommandEvent() method.

QueryClose() called repetitively when objects of TextBox class and its descendants used

Problem: If you use the TextBox class (or any of its descendants, such as WarningBox or ErrorBox) to display a message from a QueryClose() method, the method will be called repetitively. This will only happen if the QueryClose event is associated with an MDI child window and more than one child window is active.

Workaround: Place SUPER:QueryClose and SELF:SetFocus() calls at the beginning of the method before the instantiation of the TextBox.

Dialog window resources are not released by EndDialog()

Problem: Resources created by a dialog window are not released when you call the EndDialog() method.

Workaround: When the application is finished with the dialog window, explicitly destroy it using the Destroy() method.

Instantiating a window which resides in a user library results in a vestigial window

Problem: When a window residing in a user library is instantiated it is displayed only as a small box with no visible controls.

Workaround: In the root application, ensure that the GUI Classes are placed after the user library in the Libraries Included box of Application/Properties.

ADAM Initialization failure starting CA-Visual Objects with a utility active

Problem: Attempting to start CA-Visual Objects when the Install Maker, Repository Manager, or CAVORIDX.EXE is running, will result in an ADAM Initialization failure message, followed by a GPF and possible repository corruption.

Workaround: Wait for the utility to finish, then exit the utility before starting CA-Visual Objects.

GPF when closing CA-RET window before report is finished

Problem: If you close the CA-RET window before the report is finished running, you may get a GPF related to the fact that the DDE connection was destroyed.

Workaround: Do not close the CA-RET window until the report is finished.

ComboBox does not scroll when owner window is scrolled

Problem: ComboBoxes may not be properly repositioned when you scroll the owner window.

Workaround: Close the ComboBox before scrolling the owner window.

DBOI_INDEXNAME parameter of DBOrderInfo() returns the full path for DBFMDX

Problem: The *DBOI_INDEXNAME* parameter of the DBOrderInfo() function returns the full path of the current index as opposed to just the index name when using the DBFMDX driver.

Workaround: Write code to strip off the path.