IBM Component Broker Products
Release 1.3
Late Breaking News

This page last updated Thursday, July 30, 1998 at 7:20PM.

Contents

Migration Information for Release 1 Users

Migrating Object Builder Models

Model information developed using prior releases of the Object Builder tool will be automatically migrated to this release when opened. This information is maintained in .uni files.
WARNING: If these .uni files are stored in a subdirectory in the Component Broker directory tree, they will be deleted when Component Broker is uninstalled.

Release-to-Release Compatibility

The server portion of Component Broker Release 1.3 models must be migrated, regenerated, and recompiled. Client code generated from 1.2 models may be simply recompiled. Binaries produced on Release 1.2 or earlier will not run on Release 1.3 without recompilation or regeneration.

The following are exceptions require source code changes:

- The names of interfaces used for factory finding for reference collections have changed. Some backward compatibility is provided, but only for DB2 reference collections. See "Using Factory Finding with Reference Collections" below for more information.

- Query iterators after a transaction commit no longer remain valid after a transaction commit. See "Using query iterators" below for more information.

Release-to-Release Interoperability

Component Broker Release 1.3 servers and clients do not interoperate with Component Broker Release 1.2 or earlier clients and servers. For this release, it is necessary to migrate the clients and the servers to which they connect at the same time.

Release Contents

Changes from Prior Release

The following has been added or completed testing since Release 1.2:

- Improved Stability
- Improved Performance
- Application Adaptors (Windows NT only)
  - CICS (3270, ECI)
  - IMS
  - Samples for Oracle, and MQ
- Composed Business Object tools (Windows NT only)
- Java client on Windows 95
- Support for DCE 2.0

Early-Tested Functions

In order to promote rapid feedback on new functions, the Component Broker team includes certain functions in a release that have not been completely tested. We encourage you to work with these functions and report any problems observed, in the same manner as with the remainder of the release. In this way, we can deliver function more quickly and improve its quality by exposing it to your environment earlier. Please be assured that problems in these functions are treated with the same gravity as problems elsewhere in the release.

In this release, the following functions are classified as "early-tested" and available on Windows NT only:

- Application Adaptors
  - CICS, IMS Application Adaptor (LU 6.2, Sync Level 2)
  - Oracle Application Adaptor (Version 8 database)
- Java client on OS/2
- Samples
  - Enterprise JavaBeans
  - Business rules
- SSL enablement
- Tivoli integration (large-grained)
- Transient Notifications
- Workload Management

Known Component Broker Restrictions

This release of the Component Broker products has the following restrictions. These restrictions arise from known problems that are being resolved for subsequent refreshes.

General

Installation and Configuration

Note: most installation and configuration information is discussed in the *IBM Component Broker Quick Beginnings*

Guide.

Preparing for Windows Registry problems	Occasionally, Component Broker installation fails and subsequent restarting of Windows NT fails. This problem appears to involve the registry and the path length of various environment variables; however, IBM and Microsoft have not been able to identify the particular conditions causing the problem.	Windows NT

Investigation of this problem is continuing. Contact your IBM sponsor for any updates.

To be prepared in the event this problem occurs, create an Emergency Repair Disk (ERD) prior to installing Component Broker using the RDISK.EXE utility provided with the operating system.

If the system fails to reboot after installing Component Broker, do the following steps to recover the previous configuration:

1. After selecting the operating system to start (e.g., Windows NT Workstations Version 4.00) a screen will briefly appear with the message:
   "Press spacebar NOW to invoke Hardware Profile/Last Known Good Menu".
2. Press the spacebar at this time and choose the most recent good configuration. This should allow the machine to restart.

If the method described above fails to restart the computer, boot the Emergency Repair Disk and follow the prompts to restore the system.

Investigation of this problem is continuing. Contact your IBM sponsor for any updates.

Installing DCE 2.0	The *IBM Component Broker Quick Beginnings Guide* discusses the Commercial Data Masking Facility (CDMF) version of DCE. Component Broker actually supports both the CDMF version and Data Encryption Standard (DES) version of DCE. Since the availability of a particular version in a particular country is governed by export laws, the appropriate version of DCE is provided on the Supplemental Programs CD supplied to each customer.	Windows NT

The instructions for installing DCE are the same for both versions of DCE.

	On some machines, starting another process while the DCESETUP process is running will cause the DCESETUP process to terminate prematurely.	Windows NT

When installing DCE 2.0, stop any background processes (especially Microsoft IIS) and avoid starting any other process until DCESETUP completes.

Configuring certificate lifetime	On Windows NT, the maximum DCE certificate lifetime value for an authentication policy is 24 days. Do not set this value greater than 24 days.	Windows NT
Using DCE 2.0	dce_login generates an extraneous "Password must be changed!" message. Merely changing the password does not eliminate the message.	Windows NT

To eliminate the extraneous message, do the following steps:

1. From a Windows NT command prompt, invoke the DCE command processor using the command
   `dcecp.exe`
2. At the dcecp prompt, enter the DCE command
   `account modify principal_name -pwdvalid yes`
   followed by the command
   `quit`
3. Change the DCE password.

Installing UDB	If the **CLASSPATH** or **PATH** environment variables have very long values, installation of UDB Version 5.0 may not succeed and may leave these variables with incomplete settings.	Windows NT

Either of the following symptoms indicate that this problem has occurred:

- Either of these variables is left with only the data added to it by the UDB install.
- The installation may fail, showing a window titled "DlgcacWinName:_INS0432._MP - Application Error" and containing the statement "The instruction at 0x00706c65" referenced memory at 0x00706c65. The memory could not be read."

In either case, the environment variable settings must be corrected.

To recover from this problem, save the environment settings prior to starting the UDB installation. There are two methods to save the settings:

- Save the settings into a text file, then manually update the settings after the UDB installation completes.
- Create new environment variables to hold the original settings, then manually update the environment variables with the original settings after the UDB installation completes.

In either case, reducing the **CLASSPATH** and the **PATH** variables' settings to lengths less than 300 characters allows the UDB installation to complete successfully. After the installation completes, these variables may be modified to include all of the original settings. When reducing these variables' settings, do not remove any operating system paths such as C:\WINNT\system32 or C:\WINNT.

Overcoming VA Java startup errors	Due to a known problem, the workspace file for VisualAge Java may become corrupted. During startup, the following message appears if the file has been corrupted:	Windows NT

VisualAge for Java is unable to start due to an unrecoverable error in the startup sequence. One or more required files may be missing or the workspace file may be corrupt.

Fortunately, this corruption does not affect any of your files.

There are two options for recovering from this problem.

The first option is to restore the workspace file from a backup copy, if one is available. To create a backup, copy the IDE.icx file from the IBMVJava/Ide/program directory to a safe location immediately after installing VisualAge Java. If the startup error appears, do the following steps to recover:

1. Replace the current copy of the IDE.icx file with the backup copy.
2. Start VisualAge Java.
3. Add projects to the repository.

If the backup file is unavailable, the second option is to reinstall VisualAge Java. Do the following steps for this option:

1. Delete the IDE.icx file from the IBMVJava/Ide/program directory.
2. Reinstall VisualAge Java.
3. Start VisualAge Java.
4. Add projects to the repository.

Selecting an installation path	If the installation path contains a directory name with a space in its name (e.g. "Program Files"), the CB Connector service may fail to start after system restart at the end of installation.	Windows NT

Do not select an installation path that contains a space in a directory name.

Understanding install or uninstall failures	The Component Broker install and uninstall process uses the Windows NT 16-bit Windows Subsystem. On occasion, install or uninstall of Component Broker fails when the Windows NT Virtual DOS Machine has encountered an illegal instruction.	Windows NT

Either of the following symptoms appear when this failure occurs:

- A '16 Bit Windows Subsystem' window is generated with the text of "Hidden Console of WOW VDM. The NTVDM CPU has encountered an illegal instruction. Choose 'Close' to terminate the application".
- The install or uninstall process starts and ends without the Component Broker welcome screen popping up.

Consult the Microsoft Knowledge Base articles *Q165214 Troubleshooting MS-DOS-Based Programs in Windows NT4.0* and *Q119365 Runtime Error 6009 - Default Environment Size* for information on resolving this problem.

Insuring sufficient file space on AIX	Before installing Component Broker on AIX, back up ODM and validate that sufficient file space is available.	AIX

Do the following steps as the root user:

1. Uninstall any previous versions of Component Broker and Component Broker Toolkit.
2. Install all the prerequisite software.
3. Backup the ODM
     cd $HOME
     mkdir ODM
     cp -r /usr/lib/objrepos/* ODM
4. Ensure that there is sufficient free space in the /usr filesystem:
     df /usr
   The value returned is in 512-byte blocks. The Typical Install option requires at least 900,000 512-byte blocks of free space. The Custom Install option for the System Management User Interface or for the documentation require at least 200,000 512-byte blocks of free space each.
5. If not enough free space, expand /usr
     smitty jfs
     choose: Change / Show Characteristics of a Journaled File System
     move the cursor to /usr and press enter.
     change SIZE of file system to the new size. This should be the
        current size plus enough 512-byte blocks to provide the free
        space needed (as described in step 4 above).
     exit smitty (press F10)
6. Verify sufficient free space:
     df -k /usr
7. If not enough free space, repeat step 5.
8. Install Component Broker per the instructions in the *IBM Component Broker Quick Beginnings Guide*.

Installing AIX clients or remote servers	On AIX, adding additional systems into the CB network will cause the following error during system management configuration:	AIX

Cardinality error - there is already a forward relationship.

If this error appears, update the /etc/inittab file to ensure that Component Broker is automatically started when the system is restarted by doing the following steps:

1. Under the root userid, backup the /etc/inittab file.
     cp /etc/inittab /etc/inittab.back
2. Edit the original /etc/inittab file to add this line at the end of the file:

CBConnector:2:wait:/usr/lpp/CBConnector/etc/rc.CBConnector >
/dev/console 2>&1 # Start CBConnector

3. Restart the machine to start Component Broker.

Installing Component Broker on a system where Component Broker is already installed	During installation of Component Broker on Windows NT, a popup message titled "Read Only File Detected" with the message "A read only file, <CBInstall_Path>\bin\somscl1i.dll, was found while attempting to copy files to the destination location. To overwrite the file, click the Yes button, otherwise click the No button." may appear.	Windows NT
	Click No to ignore this message. The popup is displayed only if Component Broker is already on the system and the file system where Component Broker is installed is not set up to allow replacement of existing files.	
Installing a System Manager workstation	In this release, the System Management installation package is incomplete when installed alone.	All Platforms
	To configure a workstation as a System Manager workstation, install the Typical Install option, then proceed with the System Management only installation instructions.	
	Furthermore, on AIX, custom installation does not support adding the System Management Application to a Server installation. Install the workstation using the Typical Install option to install both the Server and the System Management Application.	AIX
Obtaining a JDK	For assistance in obtaining an appropriate Java Developer Toolkit, contact your IBM sponsor.	All Platforms
Installing a Java client	If a JDK is installed after the Java client is installed, the **PATH** environment variable may be incorrect.	All Platforms
	Check the **PATH** environment variable to ensure that the JDK entry (e.g. \jdk1.1.6\bin) appears before the Java client entry (\cbroker\bin).	
Selecting the directory to install a Java client on OS/2	On OS/2, the browse function of InstallShield Java Edition does not work correctly when selecting the directory into which the Java client is to be installed.	OS/2
	If the default directory is not acceptable, type the desired directory into the entry box rather than using the browse function to select the directory.	
Installing a Java client on OS/2	The **CLASSPATH** environment variable is not set correctly when the Java client is installed on OS/2.	OS/2
	Before restarting the system, locate where the **CLASSPATH** environment variable is being set, e.g. config.sys, and change its setting to include C:\CBROKER\JAVAORB\LIB\SOMOJOR.ZIP.	
Configuring the location of the transaction log	For machines with multiple physical drives, it is possible to significantly improve the transaction throughput of Component Broker by placing the transaction log on a different physical drive from the drive containing the operating system's paging file.	All Platforms
	On Windows NT, to check and adjust the configuration of the transaction log, do the following steps:	

1. Determine which drives map to which physical disks in your system:
   1.1 Click on **Start** button.
   1.2 Go to **Programs->Admin Tools->Disk Admin**.
   1.3 Make a note of which drives map to which physical disks.
2. Determine transaction log location:
   2.1 Using CB System Management User Interface, click on Server Image.
   2.2 Click on the Application Server.

2.3 Right click and go to the "**Edit**" box.
2.4 In the upper right hand corner of **Edit**
screen use the scroll button to see the
Transaction Service tab.
2.5 Click on the **Transaction Service** tab.
2.6 Make a note of which drive letter the log resides on.
(e.g. c: d:) .
2.7 Repeat the 2.1-2.6 for each Application Servers.

3. Determine paging file location:
   3.1 Click on **Start** button.
   3.2 Go to **Setttings->Control Panel**.
   3.3 Click on the **System** icon.
   3.4 Click on the **Performance** tab.
   3.5 Click on the "**Change button...**" in the Virtual Memory
   box.
   3.6 Make note of the drive letters used for paging files.
4. Determine if transaction log and paging files reside on same disk:
   4.1 Compare the drive letter(s) obtained in step 2.6
   with those obtained in step 3.5. If there are no
   matches your transaction log and paging files reside
   on separate disks and nothing further is required.
   4.2 If you find a match in step 4.1 repeat steps 2.1-2.5
   and change the transaction log location to a
   disk that does not contend with the paging files.

On AIX, to check and adjust the configuration of the transaction log, do the
following steps:

1. Determine transaction log location:
   1.1 Follow steps 2.1-2.5 above.
   1.2 Make note of the transaction log path name.
   1.3 From command prompt issue "lsfs" to
   obtain file system list.
   1.4 Under the Mount Pt. column look for a path that
   matches the first portion of the trans log path
   noted in step 1.2. The is usually "/var" .
   1.5 Make note of the File System Name corresponding to
   this mount point. (e.g. /dev/hd9var).
   1.6 From cmd prompt issue "lspv -l hdisk0". Under LVNAME
   column search for logical volume name noted in step
   1.5. If found you have located the disk which
   contains the transaction log location.

   If not found repeat the "lspv" cmd with a different
   disk number (e.g. hdisk1). Continue until you
   find the disk which contains the logical volume
   name noted in 1.6. To list all physical volumes
   use "getlvodm -C" cmd.
2. Determine your paging file(s) location:
   2.1 From cmd prompt issue lsps -a.
   2.2 Make note of the physical volumes used for the
   paging space. (e.g. hdisk0 hdisk1).
3. Determine if your transaction log and paging file reside on the same
   disk:
   3.1 Compare the disk names found in step 1.7 and 2.2.
   If no match then the transaction log is already
   on a separate disk from paging files and nothing
   further need be done.
   3.2 If a match is found repeat steps 2.1-2.5 listed
   above in the NT section and change the transaction
   log location to a disk that does not contend with
   the paging files.

IMPORTANT NOTE: If "Activation" of a Server occurs you will need to repeat the steps outlined in 2.1-2.7 after the activation is complete. Activation is typically the last configuration step done after new applications are configured on a Server Image.

Preparing to build applications on AIX	When installing a development environment on AIX for building client applications, the CBConnector.filesets.rt_samples fileset must be installed prior to the installation of the CBToolkit.TOOLKIT.clientSamples.SDK fileset. When using the "Typical Install" option, this fileset is installed. When using the "Typical Clients" option, however, explicitly specify that the CBConnector.filesets.rt_samples fileset be installed.	AIX
Setting up the EJB sample	The documentation describing how to setup the LIB environment variable for the EJB sample is incorrect (it is correctly described in the EJB tools setup guide.) The correct setting for the LIB environment variable is: SET LIB=%SOMCBASE%\samples\EarlyTestFunction\EJB\lib;%LIB%	Windows NT
Loading applications	Before loading applications, ensure that an initial activation of the configuration has occurred. This causes the port for the Name Server to be set correctly before the application is loaded. Failure to activate prior to loading applications will cause client timeout errors while trying to contact the Name Server.	All Platforms
Invoking "Force Delete" for Host models	The *IBM Component Broker System Administration Guide* describes the use of a "Force Delete" action on Host Models. Contrary to the documentation, this action is not available from the user interface. To perform a "Force Delete" action, do the following steps:	All Platforms

1. On the machine running the System Manager application, change to the Component Broker data subdirectory, e.g.
   c:
   cd \CBroker\data
2. Issue the following command:
   bgcmd @<System-manager-host-name>/
   HostM.<host-to-be-deleted> blockdo _forceDelete
   For example, if the System Manager application is running on host "smappl.aaa.com" and a force delete is needed on host "client1.aaa.com", enter:
   bgcmd @smappl.aaa.com/HostM.client1.aaa.com
   blockdo _forceDelete
3. Upon completion of the "Force Delete" action, stop and re-start the CBConnector service.

Invoking "Force Delete" for Name Servers	Although not documented in the *IBM Component Broker System Administration Guide*, a "Force Delete" action is available for Name Servers that behaves similarly to a "Force Delete" for Application Servers. However, such a "Force Delete" does not remove entries from the DCE CDS name space that are related to the Name Server being deleted. Use the DCE Director to examine the DCE CDS name space contents and remove any entries related to the Name Server being deleted. The following entries may be deleted:	All Platforms

- The host tree located at "CBC-local-roots/<some long identifier>". This can be deleted if long identifier in CBC-local-roots represents the host in question. Some of the possible ways to determine this are:

  Look in CBC-root/hosts. If there is a binding for the host whose name

server is being forcibly deleted, open that binding. It should provide the identifier for the local root for that host.

Look in CBC-root/workgroups/<name of workgroup>/hosts for each workgroup. If there is a binding for the host whose name server is being forcibly deleted, open that binding. It should provide the identifier for the local root for that host.

Look in CBC-local-roots/<some long identifier>/host/resources/servers for every long identifier under CBC-local-roots. Check each one to see if a name server is registered there. The name of the host is used in constructing the name of the name server, so this can be used as a mechanism to relate the long identifier to the host it represents.

If this analysis confidently identifies which long identifier represents the host for which the force deletion is being done, then delete it.
- If the name server being deleted manages a workgroup or workgroups, the workgroups may also be deleted. They can be found at "CBC-workgroup-roots/<some long identifier>". These can be deleted if the proper long identifier in CBC-workgroup-roots can be identified for the workgroup to be deleted. The best way to do that is to look in CBC-root/workgroups. Open the bindings for the workgroups to be deleted and find the long identifier for the workgroup in question.

Uninstalling client applications	Applications defined in DDL as having "Client Applications" are installed with shortcuts registered in the Start menu. When uninstalling Component Broker, these shortcuts are not removed. To remove these shortcuts, uninstall the client application before uninstalling Component Broker.	Windows NT
Reverting to the Original Install State	If an application adaptor is installed (CICS and IMS or Oracle), the procedure documented in the Quick Beginnings Guide for reverting to the original install state should be modified after the line "The system is now ready for activating or loading applications". Add the following line: If a CICS and IMS or Oracle adapter is installed, use the "Load DDL" action rather than the "Load Application" action to reload your DDL. In particular, load sompaa.ddl for CICS and IMS; somoaa.ddl for Oracle.	Windows NT
Uninstalling the VisualAge C++ compiler	The VisualAge C++ compiler removes the directory into which it was installed when the compiler is uninstalled. Any user subdirectories added to this directory are also removed.	Windows NT

Component Broker Toolkit

Tools

Note: most tool information is discussed in the online documentation and the *IBM Component Broker Quick Beginnings Guide*, including many restrictions. Additional restrictions are described below.

Using Rose 98	In this release, the Rose bridge support handles Rose 98. Some names used for Rose 4.0 have changed in Rose 98, specifically: - the "rosecpp.pty file in the C++ subdirectory of the Rose install directory" in Rose 4.0 is the"rose.pty file in the Rose install directory" in Rose 98. - the "C++ page" in Rose 4.0 is the "IDL page" in Rose 98.	Windows NT

Mapping models between Rose and the Object Builder tool	In this release, the default selection on the mapping panels of the Rose Bridge function has changed. Now, the first item is selected by default.	Windows NT
	The default selection may cause warnings or errors when the model is bridged.	
	Verify that the selection is desired before proceeding with the bridging function.	
Understanding exceptions when importing XML	When importing XML files, an exception such as the following may be displayed:	All Platforms

	Importing Build Config Exception
	java.lang.NullPointerException
	    at com.ibm.ivb.ob.model.IXODLL.bindToCDM(Compiled Code)
	    at com.ibm.ivb.ob.model.IXODLL.toCDM(Compiled Code)	

	This exception may be ignored since the model will be imported correctly.	
Importing XML for BOs associated with multiple MOs	If an XML model is imported that contains a Business Object implementation associated with multiple Managed Objects, the interface inheritances for the managed object interfaces will not be correct.	All Platforms
	To correct the interface inheritances, open each Managed Object's SmartGuide and select "Finish".	
Importing XML from a 1.2 model	When importing a 1.2 XML file into the Object Builder tool, a copy helper is not automatically selected for the BO implementation object.	All Platforms
	To update the BOImpl object after importing the XML, open the BO Implementation SmartGuide, select the copyhelper from the Key and Copy Helper page, then click finish.	
Handling the Out of Memory exception message	When using the Object Builder tool, the following exception may appear occasionally on the console:	All Platforms
	java.lang.OutOfMemoryError *at sun.awt.image.Image.<init>(Compiled Code)* *<additional lines>*	
	This exception causes a failure in repainting the display, but does not affect the model.	
	This message may be ignored.	
Overriding inherited protected attributes	The Object Builder tool provides a SmartGuide for overriding attributes on a business objects. However, if the attribute is inherited and protected, it may be selected, even though it may not be overridden. While the selection will appear in the Overrides pane, it will not appear after the SmartGuide has been closed and reopened.	All Platforms
Controlling include files added into the BO interface file	Any attributes or methods that are added via the Business Object interface will have whatever includes are required generated in the BO interface IDL file, even if the attributes or methods are protected or private access.	All Platforms
	To control what includes appear in the BO interface IDL file, add the attributes or methods to the BO implementation instead. The attributes or methods may be set for any access (including public). This approach prevents the include from appearing in the interface IDL file.	
Finishing BOImpls containing UUID DOImpls	When a Business Object Implementation contains UUID Data Object Implementations, finishing the BO Implementation Properties SmartGuide causes incorrect code to be generated for the internalizeFromCopyHelper method on the UUID DOImpl. In particular, the method does not contain the call:	All Platforms

```
IBOIMExtLocalToServer_IUUIDDataObject_Impl::
internalizeFromCopyHelper(inCopy);
```

To cause the correct generation of this method, open the DOImpl's Properties SmartGuide and select "Finish".

Defining keys with enums	If the key of an interface includes an enum, then any references to that interface that use the Home/Key mapping pattern in the DO will contain incorrect code. If this pattern is required, make the following changes to the code generated for the DOImpl that contains the reference:	All Platforms

1. In the DOImpl.ih, change the type of the protected "FK" data member from ::ByteString_var to the enum used.

   For example, change:
   ```
   ::ByteString_var iUtilitiesRes_typeFK;
   ```
   to:
   ```
   ::CityObjectInterface::ResTypeEnum iUtilitiesRes_typeFK;
   ```

2. In the DOImpl_I.cpp, locate every time the protected data member is set and add a cast to the enum type.

   For example (delegating DOs), in the DOImpl_I.cpp... in utilities() (the getter) change:
   ```
   iUtilitiesRes_typeFK = iBankAccountPO.res_type();
   ```
   to:
   ```
   iUtilitiesRes_typeFK =
   (::CityObjectInterface::ResTypeEnum)iBankAccountPO.res_type();
   ```

   In internalizeFromCopyHelper() change:
   ```
   iUtilitiesRes_typeFK = "";
   ```
   to:
   ```
   iUtilitiesRes_typeFK =
   (::CityObjectInterface::ResTypeEnum)0;
   ```

   For example (local copy DOs), in the DOImpl_I.cpp... in retrieve() change:
   ```
   iUtilitiesRes_typeFK = iBankAccountPO.res_type();
   ```
   to:
   ```
   iUtilitiesRes_typeFK =
   (::CityObjectInterface::ResTypeEnum)iBankAccountPO.res_type();
   ```

   In internalizeData() change:
   ```
   iUtilitiesRes_typeFK = iBankAccountPOCopy.res_type();
   ```
   to:
   ```
   iUtilitiesRes_typeFK =
   (::CityObjectInterface::ResTypeEnum)iBankAccountPOCopy.res_type();
   ```

Resolving model inconsistencies	The error "Invalid macro name _2boflag_" occurs when generating managed object code from a model with inconsistencies.	All Platforms

To correct this problem, export the contents of the model to XML (via the "obexport" command), and then create a new model by re-importing the XML (via the "obimport -X" command).

Note: include the names of all other projects used by the project in the obimport command.

Configuring managed objects	If the Configure Managed Object SmartGuide was opened and more than one DLL was specified for the customized home and Finish was selected, reopening the SmartGuide will show the alphabetically first DLL selected rather than the multiple DLLs specified previously.	All Platforms

Before generating DDL for this managed object, reselect the correct DLLs and select Finish.

Using the batch generation command	The obgen command will generate a single model or multiple models (incuding implicitly dependent models such as where project A depends on project B which depends on project C). However, for some cases, the code generated for implicitly dependent models may be incomplete. To ensure that all of the code is generated, make the top-level model (e.g. project A) explicitly dependent on the dependent models (e.g. projects B and C), then run the obgen command.	All Platforms
Understanding DDL generation error messages	If an error message such as "0 >= 0" pops up during DDL file generation, there is an inconsistency in the model. To correct this inconsistency, delete the managed object in the application family tree and re-add it.	All Platforms
Generating DDL files with additional executables or DLLs	When generating DDL files, any additional executables or DLLs specified on the Additional Executables page are included, without regard for the target platform. Therefore, when generating a DDL file for AIX, inappropriate executables or DLLs may be included that are intended for another platform. To avoid including inappropriate executables or DLLs, generate the application family by platform, instead of all at once. Check the Additional Executables page to ensure that the target platform-appropriate executables or DLLs are specified. In this release, always generate an application family on the platform on which the application is targeted to run.	All Platforms

Using the Event or Notification services	In this release, the Object Builder tool does not provide assistance in building applications that use the Event or Notification services.
	
	To build such applications within the Object Builder tool, import the IDL for the Event and Notification services' parent classes into the Object Builder tool and follow the example provided in chapter 11 of the *IBM CBConnector Cookbook Volume 1* book.
	
	These IDL files may be found in the CBroker\include directory. For Event services, import the following files:
	  CosEventComm.idl
	  CosEventChannelAdmin.idl
	For Notification services, import the following files:
	  CosNotifyComm.idl
	  CosNotification.idl
	  CosNotifyChannelAdmin.idl
		All Platforms

Importing Data Objects	When a Data Object is imported, it is not automatically linked with a Business Object; therefore the key and copy helpers are not set. To properly link the two objects together, do the following steps: 1. Import the Data Object. 2. Open the corresponding Business Object and select the Data Object. 3. Open the Data Object Implementation properties and select the Key and Copy Helper.	All Platforms
Creating Persistent Objects from Views	When creating POs from views, ensure that the key is set in the view prior to creating the PO.	All Platforms
Mapping the retrieve method	If a Data Object is mapped to a Persistent Object, the DO retrieve method must be mapped to the PO retrieve method. If this mapping is omitted (which may occur when inherited DOs are used), the following error will occur: *java.lang.Exception: Invalid macro name:_2POAttribName_*	All Platforms

If the error occurs, correct the mapping using the DO Implementation SmartGuide.

Mapping attributes	The documentation is incorrect regarding the mapping of DOImpl attributes to a PO in the section "Inherit View". The DO attributes for both the parent and the child need to be mapped to a SharePO and a ViewPO for the query to work.	All Platforms

Create the view to include all the attributes desired, just as in a horizontal inheritance. Do not make Partial View to be tied together later (vertical inheritance).

Mapping structs	The mapped type for a struct is not generated correctly into the DDL when using a DOImpl marked for the DB2 or Oracle caching service.	All Platforms

Such mappings may be identified by the "mt = Struct" setting:

```
MappedType.AccountDOImplDO_table_AccountModuleA_Account_AccountDetails_Details
{

// Define the attributes.
mt = Struct;
signature = "SIN integer, Balance double precision, Remarks varchar(1)";
primaryKey = "";
implementation = "select d.\"SIN\", d.\"Balance\", d.\"Remarks\" from this d";
parmList = "this AccountPO_Table";

}
```

To correct this DDL, change the parmList setting of the struct to specify the PODAO_Table instead of the PO_Table by inserting "DAO" between "PO" and "_".

Mapping key attributes	In data definition language generated from the Object Builder tool to map types may contain 'null' for key attributes in the where clause, e.g.	All Platforms

```
implementation = "select dPolicyEmSQLPO.PAMOUNT,
dCPolicyEmSQLPO.POLICYNO,dCPolicyEmSQLPO.PPREMIUM,
dCPolicyEmSQLPO.\"year\",dCPolicyEmSQLPO.MAKE,
dCPolicyEmSQLPO.MODEL,dCPolicyEmSQLPO.SERIALNUMBER,
dCPolicyEmSQLPO.COLLISION,
dCPolicyEmSQLPO.GLASSCOVERAGE from this dPolicyEmSQLPO,
CPolicyEmSQLPO_Alias dCPolicyEmSQLPO where
dPolicyEmSQLPO.null=dCPolicyEmSQLPO.POLICYNO";
```

The 'null' in the where clause at the end of the example should be 'POLICYNO'.

Edit the generated DDL to replace the 'null' with the attribute on the right side of the equality in the where clause. Take care to match the case exactly.

Mapping non-key and key attributes	When a non-key attribute of a DOImpl is mapped to the same PO attribute as a key attribute on the DOImpl, the internalize...() methods generated for the DOImpl will incorrectly initialize the PO's key attribute.	All Platforms

At runtime, this problem surfaces as a SQL -803 error during creation of a Managed Object.

This problem occurs more frequently when the DO contains object references which are mapped as foreign keys. It is often the case that there are PO attributes that are in both the foreign key and in the primary key.

To correct this problem, do the following steps for each non-key attribute on the DOImpl that maps to one or more PO key attributes (ie. PO attribute that are also mapped under a key attribute on the DO Impl):

1. Open the Method Implementation SmartGuide on the setter.

2. Select "Use the implementation defined in the editor pane" and finish the SmartGuide.
3. Use the editor pane to empty out the setter method body. This will ensure that these PO attributes are not unnecessarily reset.

Understanding the behavior of PO attributes not mapped by DO attributes	When a DO is initialized from a Copy Helper and then inserted into the persistent store, any attributes of the underlying PO not mapped to the DO will be initialized to non-NULL "sentinal" values (such as 0 for numbers and "" for strings).	All Platforms

This behavior may be unwanted in some scenarios, such as DO Inheritance with Views into a Single Table (where the parent and child DO views test for NULL and non-NULL respectively on a child column in the shared table).

To avoid this behavior, do the following steps:

1. Use the View Editor of CBToolkit to create a new View over the Table. When defining the View, select only those columns that will be mapped to the DO.
2. Open the Add Persistent Object SmartGuide on the new View. Select "View is Updatable". Indicate the appropriate key attributes. Finish the SmartGuide to create a new PO.
3. Open the properties SmartGuide on the DO Impl and associate the new View PO with the DO.
   Reminder: Make sure that the first associated PO has all the key attributes and that remaining POs are in alphabetical order.
4. Map the DO attributes to the attributes of the new View PO. Do not remove the mappings to the existing POs.
   Note: In the DO Inheritance with Views scenario, make sure that the last mapping of each DO attribute is to the original View PO from which the data is being retrieved.
5. Map the DO insert, update, del and setConnection methods to the new PO instead of the Table PO. Map the retrieve method to the original View PO.

Correcting variable declarations within the internalizeData method	Binary data items (eg. VARCHAR FOR BIT DATA in DB2) are delivered in the datasequence to the internalizeData() method as "::ByteString"s. The temporary variables used to receive these data items are declared as ::ByteString_var but should be declared as::ByteString*.	All Platforms

To correct these declarations, use either the Method Implementation SmartGuide or the Import Changes facilities of the Object Builder tool to find each chunk of code in the internalizeData() method that resembles the following and make the changes indicated:

```
...
MyPO_MyAttrDB2VARCHAR MyPO_MyAttrTemp;
// REMOVE // ::ByteString_var MyPO_MyAttrByteStringTemp;
/* ADD */ ::ByteString* MyPO_MyAttrByteStringTemp;
if (!(((((dataSeq[2]))) >>= MyPO_MyAttrByteStringTemp))
MyPO_MyAttrTemp.length = 0;
else {
MyPO_MyAttrTemp.length =
MyPO_MyAttrByteStringTemp->length();
for(short i=0; i < MyPO_MyAttrTemp.length ; i++) {
// REMOVE // MyPO_MyAttrTemp.data[i] =
MyPO_MyAttrByteStringTemp[i];
/* ADD */ MyPO_MyAttrTemp.data[i] =
(*MyPO_MyAttrByteStringTemp)[i];
}
...
```

Associating multiple POs with a DO	If multiple POs are associated with a DO, the order of the POs as shown in the Associated Persistent Objects page must conform to the following rules or the createIterator() method on the home may throw a CORBA::Exception with id CORBA::UNKNOWN:	All Platforms

- The first associated PO must be one to which all the DO key attributes are mapped.
- The remaining associated POs must be in alphabetical order by their instance names.

When associating POs with a new DO, obey the above rules.

For a DO that already has associated POs that do not agree with the order described above, do the following steps:

1. Open the Properties smartguide on the DO Implementation object.
2. Go to the Associated Persistent Objects page.
3. Identify any one PO that persists all the DO key attributes, then delete all OTHER POs and re-add them with the same instance names as before and in alphabetical order by instance name.

   Note: If you choose different instance names when re-adding the POs then you will need to recreate the mappings on the Attributes and Methods Mapping pages of the smartguide and regenerate the SM/DDL under the Application Configuration folder.
4. Finish the DO Implementation SmartGuide and regenerate the DOImpl.

Creating transactional managed objects	There is a restriction on the type of container that may be configured for transactional managed objects. When Use RDB Transaction Services is checked on the Container - SmartGuide frame for Services, there are three types of behavior for methods called outside a transaction. 1. Start a new transaction and complete the call. 2. Throw an exception and abandon the call. 3. Ignore the condition and complete the call. The third option is not supported in this release and should not be selected. While the Object Builder tool will successfully generate the model with this container, the application may experience intermittent failure while executing methods outside of a transaction. The exception from the server is IBOIMException::IDataObjectFalied and the DB2 database error is *SQL0998N, Reason Code = 5. Subcode = "". SQLSTATE=58005 routine invoked in an improper context.* The reasons for this restriction are complex. Using SQL outside of a transaction requires that the thread making the request have its own non-transactional connection established. We continue to support customization of data objects to do calls to databases, but not as part of a transactional being external commit coordinated by Component Broker. Do not select the "Ignore the condition and complete the call" behavior type.	All Platforms
Using the horizontal partitioning pattern	In this release, when defining queryable Business Objects that use Caching service (DAO) Data Objects, the DOs may not inherit via the attribute duplication (or horizontal partitioning) pattern.	All Platforms
Mapping schema types for Oracle databases	When creating a PO and Schema top-down from an Oracle Caching Services DO Impl, the Object Builder incorrectly presents DB2 column types as the defaults instead of Oracle types. To select appropriate column types, do the following: Before finishing the Add Persistent Object and Schema SmartGuide, click on each PO attribute. Select the desired Oracle column type from the "SQL Type" combo-box and, where applicable, the desired Length and Scale. Typical Oracle alternatives to DB2 types are:	Windows NT

DB2	**Oracle**
INTEGER	NUMBER, length 9, scale 0
SMALLINT	NUMBER, length 4, scale 0
DECIMAL(p,s)	NUMBER, length p, scale s
TIME	DATE Enter a size of at least 27 in the Size field for the PO attribute.
TIMESTAMP	DATE Enter a size of at least 27 in the Size field for the PO attribute.

Building models with both Java and C++ components	Since each DLL unit's makefile assumes that all its dependencies are available, building one DLL unit (say, in C++) that depends upon another DLL unit (in Java) will fail if the **Build->Out-of-Date Targets->Default** option is selected in the Object Builder tool's Build Configuration node. This occurs since this option only builds the default target for each DLL unit. To avoid this problem, use the **Build->Out-of-Date Targets->All** option.	All Platforms
Using the Composite Business Object Builder tool	The following information is in addition to that provided in the documentation for the Composite Business Object Builder tool: Java BO or a Java User-defined Composition When using a Java BO or a Java User-defined Composition the IDL for the interfaces that are used by the User-defined Composition must be specified in the "Client Source Files" Smart Guide of the Client DLL of the Java BO under the "Build Configuration" node. If they are not specified the Server with the Java BO will not find the java class files needed to run the Java BO. Exceptions When the object references to the components in a composite are created, all exceptions are caught and a "BAD_OPERATION" system exception is rethrown to the client. Many times these exceptions will be caused by incorrect input on the Location Smart Guides. The developer should look in the activity log for the server to determine the original problem that caused the exception. Adding or Renaming a Managed Object After a Composite Is Built When you add, rename or change a composited component in the Composition Editor, Composition page (Objects to Composite list), you must do the following: 1. Open any business object interfaces that are based on the group, and select Refresh from Composition. 2. Update the key, if necessary. 3. Visit the BOImpl page; edit location information if necessary; always hit Finish. Delegation between BO and DO The "Composite Component Creation Scenario" states that only Caching is supported for handling state data. This is not true. Delegation is supported. Lazy Evaluation The "Composite Component Creation Scenario" states that you must deselect the Lazy Evaluation flag. This is not true. The Lazy Evaluation flag does not have to be deselected. The Lazy Evaluation flag does not affect the creation of the component of a composite. They are always created at "initForCreation" or "initForReactivation" time.	All Platforms

When adding a Managed Object to an existing composite, do the following steps:

1. Visit the BO Implementation SmartGuide book and select "Finish". This will cause default location information to be created for the MO that was added. To change the default location information, use the Location page in the BO Implementation SmartGuide.
2. Visit the BO Implementation SmartGuide and go to the "Data Object Interface" page. On that page move any key attributes for the additional MO over to state data so they will be included in the DO Implementation.
3. Select "Finish".

Selecting the "Add..." button on the Composition page of the Composition Editor causes erratic population of the Composition Palette. No managed objects, the appropriate two managed objects, or many copies of the managed objects may appear.

Invoking the palette or the composition tool again will resolve the problem. No user data is lost.

Including a CICS component in a composite	When adding a component into a composite, the Object Builder tool will check to see if any of the Data Objects defined for the component use PAA Services. If so, the Object Builder tool will generate code in the composite to create the component using a copy helper. The initial values for the attributes in a component that use PAA Services must be set using the "Interface Smart Guide" for the component. When creating the component the composite will create a CopyHelper, set the attributes from the composite Key, and create the component. The other attributes on the CopyHelper will be set to the initial values specified on the "Interface Smart Guide".	All Platforms
Creating debug compilations	Code generated from the Object Builder tool is normally compiled using the -Ti+ option to produce a debug version. The VisualAge C++ Version 3.5.5 compiler provided with this release will automatically switch to limited debug information if the type information exceeds 64KB. If this occurs, the debug version will only support line-number-only debugging. To disable this automatic switch and generate the full debug information, specify the /yTo compiler option. This option will produce the following message, which may be ignored: *warning EDC4015: Error occurred while writing object file <source_file>.o: change to HLL5.*	Windows NT
Creating optimized compilations	Compiling code with the optimization flag turned on (O+) may cause the application server to have segmentation faults if the following code pattern is present in the application:	Windows NT

```
start loop
try
{
[do some stuff]
}
catch
{
[make a method call to an imported function
i.e. not in this DLL]
[segmentation fault occurs here due to compiler generated code]
}
end loop
```

A compiler fix is available (PMR 20453,L11,000). Contact your IBM sponsor to

get the fix.

Until applying the fix and recompiling the application containing this code pattern, do not compile the application with optimization turned on.

Debugging on AIX	On AIX, a timing problem may cause the server process (somsrsm) to be terminated while being debugged. While debugging a C++ or Java BO, the debugger will catch a SIGTERM exception. If the exception handler is run, the debug session will terminate, hanging the client.	AIX
	To recover, stop the application server, the name server, and the client from the System Management User Interface.	
	While stepping a multi-threaded application, some of the threads (other than the thread being stepped) may enter a 'blocked' state after the step completed. Most likely, the blocked threads entered a blocking call (such as sleep()) during the step but did not exit the call yet. The blocked thread cannot be stepped.	AIX
	Continue stepping the non-blocked thread until the blocked thread(s) becomes unblocked.	
Using Object Level Trace without JIT	On AIX, if the Java Just-In-Time (JIT) compiler is turned off, starting the Object Level Trace will cause a core dump exception from the OLT Client Controller.	AIX
	To avoid this problem, edit the CBToolkit/bin/ivbtrc file to remove the '-nojit' statement.	
Using Object Level Trace with Java BOs	Due to a problem with the debug API in the Java virtual machine, the OLT debugger may not honor breakpoints set within a Java Business Object. Furthermore, if a breakpoint is honored, the client application and the server may hang.	All Platforms
	The problem is being pursued with the JVM supplier. Contact your IBM sponsor for updates about this problem.	
Using the idlc tool	The -mmo option of the idlc tool is not supported in this release.	All Platforms
Avoiding certain IDL styles	Certain forms of IDL, particularly involving circular includes, that are not supported by the IDL or VisualAge C++ compiler should be avoided.	All Platforms

These forms should be avoided:

- The use of forward declares of module scoped interfaces.
  If an interface is declared in module scope, it cannot be forward declared. This can occur if two module scoped interfaces reference each other. For example, the following form cannot be made to work:
  File1.idl
  module m1
  {
  interface iface1
  {
  attribute m2::iface2 attr1;
  };
  };
  File2.idl
  module m2
  {
  interface iface2
  {
  attribute m1::iface1 attr2;
  };

```
};
```
- The use of forward declares of interfaces that are used in conjunction with inheritance.
  This can occur when inheriting from an interface that contains a reference to the derived interface type. For example, the following form cannot be made to work:
  File1.idl
  interface derived : Base
  {
  };
  File2.idl
  interface Base
  {
  attribute derived attr;
  };

Using the IDLtoJava tool	The IDLtoJava tool will fail on files that do not have an EOL/EOF combination on the last line, even if the IDL in the file is syntactically correct. The error message "Expected ';'; encountered 'EOF'." is generated under these circumstances. Add a blank line at the end of the file to correct this problem.	All Platforms
Creating application install packages	When creating application install packages usable with InstallShield 5.1, the Object Builder tool places the generated setup.exe file into a directory with a very long path. When the length of the path exceeds 65 characters (including directory separators), Windows NT cannot launch the setup.exe program. By default, the setup.exe file is placed in the directory [drive]:\[project]\Working\NT\[application family name]\IS5.1\ [application family name]\Media\Default\Disk Images\disk1, where drive, project, and application family name are provided by the user. There are three options to work around this limitation: 1. Set the Object Builder tool's project directory as the root directory and keep the application family name short, thus keeping the directory path within the 65 character limit. 2. After each Generate and Build action for an Application Family, move the subdirectory starting with "disk1" to another directory with a shorter path. 3. Start InstallShield 5.1 by double clicking on the generated .ipr InstallShield 5.1 project file in the lower-level application family directory, then launch the setup.exe within InstallShield.	Windows NT
	The Object Builder tool generates two DDL files for an application family, allowing one or both files to be installed. However, the application install package generated for InstallShield 5.1 does not support selective installation. For this release, expect to install both the Application Family and Specific Application Family subcomponents with each application install package.	Windows NT
	When an application install package is created using InstallShield 5.1 for an application containing a Composite Business Object implemented in C++ that uses a Composite Group implemented in Java. For example: Application Family: CompositeAppFamily Application: CompositeApp If you created an Application Family "CompositeAppFamily" that contained a "CompositeClient"/ "CompositeServer" using a Composite BO implemented in C++ with a Composite Group implemented in Java you will generate the following outputs:	Windows NT

CompositeServer.dll
CompositeClient.dll
CompositeServer.jar
CompositeClient.jar
The generated application install package will not include the
CompositeClient.jar or the CompositeServer.jar.

The application install package is generated correctly if the Composite
Business Object using the Composite Group is implemented in Java.

To add the CompositeServer.jar and CompositeClient.jar to the application
install package, do the following steps:

1. Start the Object Builder.
2. Edit the properties of the "CompositeApp" Application.
3. Page to the "Additional Executables" SmartGuide, select "Browse..."
   and add the CompositeClient.jar and CompositeServer.jar.
4. Select "Generate" on the "CompositeAppFamily" and then recreate the
   application install package.

The DDL is correct and does not need to be changed.

Creating application packages containing multiple, interacting Java BOs	When generating an application install package containing multiple Java BOs that interact with each other, the xxxxxC.jar (where xxxxx is the class name) is not packaged by default. Manually add this jar file to the package.	All Platforms

Samples

Updating the inheritance sample	The View part of the inheritance sample located in the directory samples/TutorialSamples/CB_Fundamentals/Inherit needs to change as follows:	All Platforms

1. In the Application Configuration delete the
   ViewDataModuleMOBeneficiary.
2. In the DBA_Defined Schemas - ShareDataGroup delete the
   CBSampDB.BView.
3. Create the view again by doing the following steps:
   a. View Name - BVIEW.
   b. Tab to View Work Area. Select S_DINH. Choose all the columns in
   the ViewEditorColumn Tool.
   c. In the ViewEditorList Tool go to the Where tab.
   d. Select the claimsPayments in the ViewEditorColumn Tool.
   e. In the ViewEditorList Tool condition choose "Is Not NULL".
   f. Press OK.
4. Edit the BView and mark the Not Null columns to be also DBKey (ssNo
   and name).
5. Add Persistent Object .
6. In the User-Defined Data Objects - ViewDataFileDO -
   ViewDataModuleDO.
   a. In the PersonDOImpl Attribute Maping page map all the attributes
   to iSharePO and iPersonPO.
   b. In the BeneficiaryDOImpl Associated Persistent Objects page add
   the iBeneficiaryPO Type BViewPO.
   c. In the BeneficiaryDOImpl Attribute Mapping page map all the
   attributes to iSharePO and iBeneficiaryPO.
   d. In the BeneficiaryDOImpl Methods Mapping map the retrieve to the
   iBeneficiaryPO and all the others to iSharePO.
7. In the Application Configuration add the BeneficiaryMO.

Updated sample code will be provided by the IBM Component Broker Support team web site. Contact your IBM sponsor for assistance in obtaining the latest sample.

Building the inheritance sample	The makefile to build the Inherit sample (part of the CB_Fundamental samples) is missing. Therefore, a make of all of the CB_Fundamental samples does not build the Inherit sample.
	
	To build the Inherit sample, do the following steps:
	
	1. On Windows NT, in the directory
	   samples/TutorialSamples/CB_Fundamentals/
	   Inherit/BusinessObjects/Working/NT
	   run the command
	    nmake -f all.mak
	   to build the objects.
	
	   On AIX, in the directory
	   samples/TutorialSamples/CB_Fundamentals/
	   Inherit/BusinessObjects/Working/AIX
	   run the command
	    make -f all.mak
	   to build the objects.
	2. Build the client application (Applcations/Java/QuickTestClient) by running the build command file.
		All Platforms

Creating children in the Inherit Key duplication sample	If the parent is already created in the Inherit Key duplication sample and the child is created using the createFromCopyHelper() method, the attributes stored in the Parent table will not be updated because the Parent insert fails due to the Parent table already existing.	All Platforms
Updating the Java Foreign Key sample	The Java Foreign Key sample model has a problem with the addCustomer method that requires regeneration. To correct the model located in samples\TutorialSamples\CB_Fundamentals\jforeignkey\BusinessObjects, open the Properties for the csAgentBO and press Finish.	All Platforms
Using the query evaluator in the TwoPO sample	The QuickTest client's Query Evaluator feature has an extra underscore ("_") in the query string built from the home name. Remove the extra underscore to enable the query to work correctly.	All Platforms
Running the cache sample on AIX	The run.ksh script used to run the QuickTest client for the cache sample does not set the **CLASSPATH** environment variable correctly. Edit the run.ksh script to change the **CLASSPATH** environment variable to include the ivbjfc.jar file from the $IVB_DRIVER_PATH path rather than the $SOMCBASE path.	AIX
Preparing the Personal Life Insurance sample for use	The compiled versions of the Personal Life Insurance application included with the Component Broker Toolkit will not run. The sample application must be regenerated to work correctly. To regenerate the sample, do the following steps: 1. Create the database and tables, create the "LifeIns" database and its tables using the sql files located in samples\TutorialSamples\BusinessObjects\Insurance_Application\Working\Nt. 2. Follow the instructions in the section "Regenerating the Life Insurance Application Code" in the README file located in the samples\TutorialSamples\PersonalLifeInsuranceSample directory.	All Platforms

Correcting the Sample Insurance Application appendix in Quick Beginnings	In the Sample Insurance Application appendix of the *IBM Component Broker Quick Beginnings Guide*, please make the following corrections: 1. The paths referred to are incorrect. The pathname samples/pli/... is now samples/TutorialSamples/PersonalLifeInsuranceSample. 2. The clients located in this directory have been repositioned into the samples/TutorialSamples/ PersonalLifeInsuranceSample/Applications directory. 3. The non-GUI Java client is no longer shipped on Windows NT. 4. The GUI Java client has been replaced with a QuickTest client provided on both Windows NT and AIX. For information on this client, see the samples/TutorialSamples/ PersonalLifeInsuranceSample/Docs/PLI.html file.	All Platforms
Updating the Personal Life Insurance sample	The following corrections to the Personal Life Insurance sample are needed: • The QuickTest build.bat file contains an extraneous goto statement that should be removed before executing the file. The complete path to this file is samples/TutorialSamples/PersonalLifeInsuranceSample/ Applications/Java/QuickTestClient/build.bat.	All Platforms
	• The documentation for the Java client is missing some GIF files. These files are available from the IBM Component Broker Support web site. Contact your IBM sponsor for assistance in obtaining these files.	All Platforms

Programming Model

Using local references to Java Business Objects	Local references to Java Business objects are not valid outside the remote method call which created the reference. For example, it is not legal to set the value of a Java Business Object into a static variable and then retrieve that object from the static on a subsequent method call and use it. The passivation scheme for Java BO's and the mixin technology of the server require that interposing being done by the C++ MO before running methods against the Java BO. Thus passing references to self around outside the context of the initiating remote method is not allowed.	All Platforms
Using security - delegation	Several areas of the documentation discuss "delegation" support. This capability is not present in this release. Please ignore those areas of the documentation referring to "delegation" support.	All Platforms
	In this release, the method SecurityLevel2::Credentials::refresh() is not functional.	All Platforms
Matching key strings to their underlying resource manager fields	In this release, the length of character strings used as key attributes in business objects must match the length of the corresponding attributes in the underlying resource manager (DB2, CICS, IMS, Oracle). For example, DB2 supports both variable- and fixed-length string formats. VARCHAR table columns are variable-length while CHARACTER table columns are fixed-length. Business object key attribute character strings are always variable-length. There are three ways to match the formats: • Use variable-length fields in the underlying resource manager and ensure that all uses of the key string in the application contain exactly the same number of characters (including blanks).	All Platforms

- If the resource manager field must be fixed-length, always pad the key strings in the application with trailing blanks to make the string length match the length of the resource manager field.
- Create a mapping helper to strip trailing blanks from the fixed length string attribute as it is read into the Data Object from DB2. This meets the "same length" requirement by shortening the fixed length string from DB2 to match the variable length string used within your business object. Use the file DB2MappingHelper.hpp found in the include directory where CBConnector was installed as an example to create your own mapping helper.

  Do the following steps:
  a) Create a new .hpp file and define your own mapping helper class.
  b) Define a method in your new class with an inline implementation that strips trailing blanks from your string attribute.
  c) Copy your new .hpp file into the "working" directory of your object builder development environment.
  d) Install your new mapping helper by using the Object Builder tool to select the "properties" pop-up window for your Data Object implementation object.
  e) Click "next" several times until you reach the "Attributes Mapping" smart guide.
  f) Click on the string attribute that you want to install the mapping helper for. A "Mapping Helper Class" dialogue box will appear within the smart guide.
  g) Fill in the "Class Name" line of this box with the name of your new mapping helper class.
  h) Fill in the "PO to DO Method" line of this box with the name of your mapping helper method.
  i) Regenerate and recompile the code for your Data Object implementation.

These approaches may also be applied for other resource managers if you have persistent business objects which are not stored in DB2.

The default mapping of a DECIMAL column in a DB2 table is to an attribute of type double. If a DECIMAL column must be mapped to an attribute of type string in a Business Object, contact your IBM sponsor for assistance, referencing defect 50057.

Handling nulls in ByteStrings	When a ByteString data type in a Business Object is mapped to a CHARACTER data type in a database using the PO caching pattern, the data is truncated at the first NULL byte.	All Platforms
	To avoid this truncation, map the ByteString data type to a VARCHAR FOR BIT DATA or LONG VARCHAR FOR BIT DATA data type in the database.	
Using Factory Finding with Reference Collections	Reference collections are configured to be available on all application servers that run the DB2 Adaptor. In situations where there are multiple servers on a host configured with reference collections, the decision about which server to create reference collections on is based on the factory which is found via the factory finder. If the normal host scoped factory finder is used, the server returned is indeterminate.	All Platforms
	Using various scopes on the factory finder can alter this pattern. See the online help for factory finders for more information.	
	In release 1.2, the following strings were used in the factory finder interface for	All

reference collections:

Platforms

DB2

"IManagedCollections::IReferenceCollectionRDB.object interface"
"IManagedCollections::IKeyedReferenceCollectionRDB.object interface"

Transient

"IManagedCollections::IReferenceCollection.object interface"
"IManagedCollections::IKeyedReferenceCollection.object interface"

Iterators

"IManagedCollections::IIteratorRDB.object interface"

In this release 1.3, these strings will be changed to comply with CORBA Life Cycle specification. The new strings will be:

DB2 Collections

"IManagedCollections::IReferenceCollection.object interface/DB2ReferenceCollectionFactory.object home"

keyed

"IManagedCollections::IKeyedReferenceCollection.object interface/DB2KeyedReferenceCollectionFactory.object home"

Transient Collections

"IManagedCollections::IReferenceCollection.object interface/TransReferenceCollectionFactory.object home"

keyed

"IManagedCollections::IKeyedReferenceCollection.object interface/TransKeyedReferenceCollectionFactory.object home"

Collection Iterator

"ICollectionsBase::IIterator.object interface"

Release 1.2 code that uses reference collections should change to use the new strings. An additional DB2 home is provided in this release, however, which allows the user to access DB2 collections using the 1.2 version of the factory finding strings.

When finding Transient collections using 1.2 version of the factory finding strings , Release 1.3 will return a DB2 collection. To find a Transient collection, use the 1.3 version of the factory finding strings.

Mapping object references to database fields

Limitations imposed by mapping stringified Object References to VARCHAR fields suggest that object references be mapped as follows:

All Platforms

- Normal Object References should be represented using a home and key.

    This allows queries to be run on those fields. Also, the record lengths may be reduced since keys typically don't require as much space as a stringified Object Reference.

- When using Object References as "back pointers" for Foreign Key Object Relationships, the Object References must be represented using a home and key.

    Note: in this release, Object Builder does not require this approach. Not

following this approach may cause runtime failures.

Using the Query service	In this release, the Query service has several restrictions:	All Platforms

- Support for relationships in queries is limited to 1 to 1 and 1 to N relationships. Relationships with outer joins are not supported.
- Host variables in queries are not supported.
- Queries with deferred cache updates are not supported; queries are pushed down to the database irrespective of whether the cache contains deferred updates or not.
- Distributed query processing is not supported; query evaluation is local to the Component Broker Connector server that is executing the query. Nevertheless, queries can span multiple back-end databases provided that the Component Broker Connector server has local connections to these databases.
- Extended data types such as DB2 LOBs and user-defined DB2 data types are not supported.
- The default ORB request timeout value of 30 seconds may be insufficient when executing queries from a client. Reset the timeout to either a higher value or to zero to wait indefinitely
- For a query such as retrieving handles on related department objects in the empHome home collection along with the avg int_sal of employees in each department (**select e.depSet, avg (e.int_sal) from empHome e group by e.depSet**), an incorrect error message is generated.

  Possible workarounds include selecting related department numbers instead of handles on departments (**select e.depSet..no, avg (e.int_sal) from empHome e group by e.depSet..no**).
- All uses of the query service must be within the scope of a transaction.

- The methods get_field_type and get_field_class_name on the query data array iterator object return incorrect results when the SELECT clause of the query statement specifies users methods as in "SELECT x.methodA() FROM homeCollection x;"

- A query statement that references a struct embedded within a struct such as "SELECT x.struct1..struct2 FROM myHomeCollection x:" does not working correctly.

Using query iterators	In this release, the behavior of iterators returned by the query service has changed.	Windows NT

This pertains to iterators returned by Home Collection methods createIterator and evaluate as well as evaluate_to_iterator and evaluate_to_data_array methods on the query evaluator interface.

The interaction between iterators and transactions commits was not documented in prior releases but the implementation was such that

a query iterator would remain valid if the caller performed a transaction commit and started a new transaction.

With this release, iterators returned by query service performing a query over a home collection or a view of a home collection now have undefined behavior after the caller performs a commit. The caller could be returned an InvalidIterator exception but an outstanding problem in the query iterator could also cause invalid data to be returned.

This may effect ManagedObjects whose methods return query iterators if the ManagedObject is configured into an atomic container.

See chapter 6 "More on Iterators" in the Programming Guide.

Improving query performance	The performance of queries may be improved by doing the following:	All Platforms

- Use the IExtendedQuery::ParameterList interface to construct a parameter list of collection names with collection references.

    This performance improvement will also work with the following interfaces:
    QueryEvaluator::evaluate_to_data_array
    QueryEvaluator::evaluate_to_data_array
- In the client application, instead of doing
    ahome->evaluate ("att1 >0");
    do instead
    CosQuery::QLType_ptr ql_type = NULL;
    IExtendedQuery::ParameterList collection_names;
    IExtendedQuery::ParameterList params;
    CORBA::ULong how_many = 0;

    IExtendedQuery::MemberList*x;
    IExtendedQuery::MemberList*& members = x;
    CosQuery::QLType_sequence * l_type;
    IManagedCollections::MemberList*x1;
    IManagedCollections::MemberList*& members1 = x1;

    IExtendedQuery::ParameterListBuilder *pb =
    IQueryLocalObjectImpl::ParameterListBuilder::_create();

    pb->add_object_parm("ahome", ahome_ptr);
    collection_names = *(pb->get_parm_list());
    qe->evaluate_to_iterator ("select x from ahome x
    where x.att1 > 0",
    ql_type,
    collection_names,
    params,
    how_many,
    members,
    result_iterator);

Avoiding data corruption when using atomic transactions	When using the atomic mixin with the caching service, in certain very specific situations, Business Object (BO) updates may be lost. The client program may execute methods on the BO to change its data, but the corresponding DB2 database record is not updated. No errors or messages indicate the failure.	All Platforms

This error occurs (intermittently) when ALL of the following conditions are met:

1. The BO Container property: "Behavior for methods called outside of a transaction" is set to "Start a new transaction."
2. The BO property: "Pattern for handling state data" is set to "Caching".
3. The DOImpl property: "Persistent behavior" is set to "DB2 Caching services."
4. The client program starts a transaction, and then obtains an object reference (using either createFromCopyString, findByPrimaryKeyString, or createFromPrimaryKeyString) and then executes methods to update the object within the SAME transaction context.

To avoid this problem, change one of the four conditions.

If changing any of the first three properties is not possible, changing the way the client program is written may be an option. One solution is to remove the start-transaction from the client program. It is not necessary, because the BO Container is configured to "start a new transaction" if none is active. Another solution is to divide the client transaction into two. Use one transaction to obtain the object reference (e.g. using createFromPrimaryKeyString) and then

start a second transaction to execute methods on the BO. The first transaction must be committed before starting the second one.

Using non-queryable BOs in a reference collection	When a non-queryable business object is placed into a reference collection, the reference collection may be iterated but the business object may not be queried.	All Platforms
Understanding differences between DataObject implementation approaches	A DataObject implemented using the embedded SQL option behaves differently from a DataObject implemented using the caching service option. Migrating a DataObject from one approach to another may require changes to applications using the DataObject. For assistance in migrating between these approaches, contact your IBM sponsor, referencing defect 49094.	All Platforms
Handling Floating Point Underflow exceptions	On Windows NT, "float underflow" exceptions may appear in javai.dll. This exception occurs due to loading a VAC++ DLL into a JVM where this exception has been unmasked by DLL initialization when the JVM did not expect it to be unmasked. The unmasking occurs in DLLs compiled with VisualAge C++ prior to the 3.5.4 level	Windows NT

The JDK sets the control word to 0x27F when the VM is started and the Component Broker runtime calls FIX_FLOAT (see below) to reset the control word before every method invocation in a Java BO.

To locate all DLLs on your system that might be calling _fpreset, use the Windows NT Start->Find->Files or Folders... tool and to search for files named "*.dll" that include the text "fpreset" on all drives and subdirectories.

Examine the setting of the FPU Control Word in the idebug tool by opening the Registers monitor and select Options->Display style. This will open another pane in the Registers window for the "FP Stack". The "FPCW" field is the control register.

For JDK to run correctly, the control register should be set to something like 0x27F, with the last 5 bits masked.

The incorrect setting that _fpreset sets on DLL load is 0x362.

To avoid these exceptions, choose from the following:

- For your own Java native method DLLs written in VisualAge C++, recompile them with the VisualAge C++ 3.5.5 compiler provided with this release of Component Broker.
- If recompiling your DLLs is not possible, apply the workaround described below.
- For other Java native method DLLs that refer to _fpreset, contact the supplier and request a recompiled DLLs that doesn't use _fpreset.

There are a few DLLs supplied with Component Broker 1.3 that call _fpreset. They will be fixed in a future release of CB. The CBConnector runtime has applied the workaround described below for all DLLs provided with the product.

Workaround

Re-mask the floating point exceptions after a VisualAge C++ DLL has called _fpreset. The following VAC++ code does this remasking:

```
// code in header
#if (defined(_WIN32) && defined(__IBMCPP__))
#include <float.h>
#define FIX_FLOAT() _control87(EM_UNDERFLOW, EM_UNDERFLOW);
```

```
#else
#define FIX_FLOAT()
#endif

... //code that loads a DLL
FIX_FLOAT();
```

When using the Java "System.loadLibrary" command to load a VAC++ DLL containing Java native methods, define an additional "fix_float" static Java native method in the DLL that contains the C++ code above. Call this method after the System.loadLibrary call to reset the control word.

Developing Resource Objects	Resource Objects, as described in Chapter 8 of the *Advanced Programmers Guide*, are not supported in this release.	All

Component Broker Connector

General

Coexisting with the Hummingbird SOCK5 server	Under particular circumstances, running Component Broker simultaneously with the Hummingbird SOCK5 server results in a General Protection Fault. This problem does not occur when the SOCK5 server is not running. To remove the SOCK5 server, do either of the following steps: • Execute the SOCK5DEL.EXE program found in the Program Files\hummingbird directory. • Rename the HCLSOCK5.DLL file in the winnt\system32 directory. Restart the system. Web browsers or other programs dependent upon SOCKS servers will be affected by the absence of the SOCK5 server. To reinstate the SOCK5 server, do either of the following steps; • Execute the INSTALL.BAT program found in the Program Files\hummingbird directory. • Rename the renamed HCLSOCK5.DLL file in the winnt\system32 directory to its original name.	Windows NT
Managing the AIX semaphores	In this release for AIX, Component Broker Connector does not free semaphores correctly. Over time, the system-wide pool of 4096 semaphores will be exhausted, causing system failures. To manage the semaphores, do the following steps while the system is quiesed: 1. Use the **ipcs -s** command to capture a list of semaphores in use before starting any Component Broker processes. 2. After running Component Broker, use the shell script **ipcrmls** provided by Component Broker. This will create a list of commands to delete ALL the semaphores in the system. Do not run this list of commands. 3. Edit the list of commands created in step 2 to remove the commands deleting semaphores you want to keep. The list created in step 1 contains semaphores that should be kept. 4. Execute the remaining commands to remove the semaphores allocated by Component Broker.	AIX
Switching code pages	Switching code pages for a started server process will result in invalid object references. Before changing a code page, ensure that the server process is stopped.	All Platforms

Using Remote Name Contexts	Remote Name Contexts, as described in the *System Administration Guide*, page 313, are not supported in this release. If a Remote Name Context is defined, the Name Server may not be able to start.	All Platforms
Changing trace levels	In this release, changing the component trace level directly on a running server will cause the server to fail.	All Platforms
	To avoid such a server failure, change the trace level by editing the Model using the System Management User Interface, then reactivate the configuration.	
Stopping DCE gracefully	DCE does not always clean up its state properly when Windows NT is shut down. Consequently upon restart, DCE does not start all of its daemons properly.	Windows NT
	Before shutting down, issue the following DCE commands from a DCE command line:	
	>>dcecp stop.dce *>>dcecp clean.dce*	

Systems Management

Finding the preinstalled CB Service applications	On AIX, the preinstalled CBConnector service applications (e.g. iDB2IMServices) do not appear in the **Available Applications** folder in the System Management User Interface.	AIX
	To add these applications to a configuration, do the following steps:	
	1. Go to the host image for the system in the UI. 2. Expand **Application Family Installs/<App Family>/ Application Installs** and drag the application from there as you would from the **Available Applications** folder.	
Resolving activation hangs	If an activation hangs with no messages appearing after the "Verification completed -- configuration valid" message appears on the activation console, stop and restart the CB Connnector Service, then reactivate the configuration.	All Platforms
Restarting the CB Connector service	When stopping the CB Connector Service and restarting it, the service may fail to restart. To recover, stop the ORB Daemon (somorbd process) and any servers (somsrsm processes) that are running, then restart the CB Connector service.	Windows NT
Resolving "could not open file" errors	If the DDL file is on a mapped network drive, a "could not open file" error may be generated when performing a Load Application action on the file. This may be due to a mismatch in permissions between the CBConnector service and ID/service used to map the drive.	Windows NT
	Validate that the permissions match before retrying the Load Application action.	
Configuring multiple server host networks	Component Broker networks with multiple server hosts present a challenge to configure and activate successfully.	All Platforms
	If you experience problems setting up and running such an environment, contact your IBM sponsor for the latest information on multi-host operations.	
	During activation of a multiple-server configuration, the activation may enter a loop repeatedly attempting to start the Name Server, the Location Service Daemon, or application servers.	All Platforms
	To correct this problem, do the following steps:	
	1. Using the System Management user interface, become "Super User"	

by selecting "View," and then "User Level".
2. Select the offending Host under the "Hosts" folder. Right-click and selete "Edit".
3. Select the "Main" tab and change the setting for "health monitor polling interval" from its default value of 10 seconds to the maximum value of 300.
4. Apply the changes, leave Super User mode, and re-activate.

Refreshing specific applications	Refreshing a "specific" application will fail when objects in the "base" application become dependent on objects in the "specific" application once the "specific" application is installed on a server. To refresh such an application, uninstall and reinstall the "specific" application.	All Platforms
Recovering from System Management User Interface timeouts	The System Management User Interface client will receive a "Remote server probably inactive" error message if the server process bgmain is not running. To recover from this timeout, stop the CBConnnector service and restart it. The User Interface may then be started.	All Platforms
Stopping a reluctant server	Under certain error conditions, normal stopping of an application server via the System Management User Interface may not complete and the server will start to consume much of the CPU cycles. This condition may be verified through the Windows NT Task Manager or the "ps" command on AIX. To correct this condition and properly recover the application server, do the following steps: 1. Issue Stop Immediate from the System Management User Interface on the application server. Normally, this will successfully stop the application server. If the Stop Immediate doesn't work, then use the Windows NT Task Manager or the AIX "kill" command to stop the process. 2. After the application server has successfully ended, clean up any open connections to DB2 databases by opening a DB2 Command Line Processor window and issuing the 'force applications all' command. 3. Restart the application server.	All Platforms
Uninstalling an application family	In some cases, uninstalling an application family may fail if the application is on a running server. Stopping the server prior to uninstalling the application family will resolve this failure.	All Platforms

Server

Resolving Interface Repository errors	When an application is installed and its IR population program run, the server where the application was installed should be activated or reactivated. Otherwise, the IR definitions for the application may not be found for a query, resulting in error messages such as: *Query error message = Error - Class ::sampleJBO::csAgent is not defined in the interface repository*	All Platforms
Understanding log messages when DCE credentials are refreshed	When DCE credentials are automatically refreshed, the following extraneous message is written to the activity and error logs: *File: src/objsvcs/security/authn/ISecurityInit.cpp* *Function: authn_and_refresh(void*)* *Message: should contain "sec_login_refresh_identity()"* This message may be ignored.	All Platforms

Managing secure and unsecure servers	In a mixed environment where some servers are secure and others are not, every server will act as a secure client, regardless of whether the server is configured to be security enabled. Thus, applications on unsecure servers may access secure servers, allowing clients to the unsecure server to drive requests which affect a secure server. Ensure that applications that reside on unsecure servers do not access secure servers.	All Platforms
Understanding server hangs on AIX	When using Java BOs on AIX, the server may hang if the Java Virtual Machine's garbage collector releases memory related to the Java BO. To minimize the likelihood that this hang occurs, increase the heap size for the Java Virtual Machine by opening the System Management User Interface and setting the Java Virtual Machine minimum heap size to 20 MB and the maximum heap size to 100 MB.	AIX
	At server startup, a hang may occur due to an initializer not completing within its allocated time. If the Activity Log shows the server waiting on an initializer, the initializer is not completing its initialization. Contact your IBM sponsor for assistance if you observe this problem.	AIX
Setting database access parameters	In this release, the application definition in a DDL was changed to accommodate multiple relational database adaptors. The model migration steps update the DDL to reflect this change. However, the change also surfaces in the System Management End User Interface, particularly regarding how database access parameters are set. XA Resource Manager Images are no longer used. The information in those images has been moved to the RDB Connection Image. Changes to the open string for a particular database (such as setting the user ID or password) must now be made to the RDB Connection Image. Also, the container definitions no longer use the data store qualifier to identify the database via the XA Resource Manager. Instead, the container is related to the RDB Connection Image through a relationship called Configured RDB Connections. Finally, this release introduces a XA Connection Image. No user interaction with this image is required.	All Platforms
Removing Server Group Gateway servers	When a configuration which defined a controlled server group is deactivated the Server Group Control Point and Server Group Gateway servers are also deactivated and may be removed. However, when a Server Group Gateway server is removed it fails to unbind an object reference from the Name Service. The object will have been bound at the path /workgroup/resources/server_groups/<server-group-name>/gateways.This redundant binding will cause a problem if the controlled server group configuration is reactivated with the same workgroup but with a different Managing Host different from the one originally specified. To allow the configuration to be successfully reactivated, remove the object reference from the Name Service entries in the DCE CDS. Please refer to the *System Administration Guide* for instructions regarding how to use the DCE Director to remove these entries.	All Platforms
Avoiding next key locking problems	Due to a feature in DB2 known as "next key locking", deadlocks are possible when multiple, concurrent transactions are run. Symptoms of a potential next key deadlock or race condition include the Component Broker application appearing to hang indefinitely or appearing to hang for a period of time before	All Platforms

catching one of the following exceptions:
  CORBA::NO_RESPONSE
  CORBA::PERSIST_STORE
  CORBA:TRANSACTION_ROLLEDBACK

Testing has found that next key deadlocks can be encountered using both embedded SQL and DB2 caching service DO implementations regardless of the current isolation level being used. This includes both pessimistic and optimistic locking used by the DB2 caching service.

The "next key" deadlock occurs when a transaction attempts to insert a row into an indexed table immediately after a row already locked by another transaction. This problem is most likely to appear with tables containing few rows or with long-running transactions that deal with many rows of a table.

Next key locking is required to guarantee ANSI and SQL 92 standard repeatable read (RR) isolation level. Component Broker currently does not support the next key locking feature of DB2. In testing CB R1.3 with DB2 we have encountered several deadlock and race conditions that are directly attributed to DB2's next key locking.

Next key locking is enabled by default in UDB Version 5.0 in order to guarantee the ANSI/SQL 92 standard RR isolation level. This is attained by automatically locking the next key for all INSERT and DELETE statements for all isolation levels and the next higher key value above the result set for SELECT statements for RR isolation level only. The next key locking done during the INSERT and DELETE statements is required regardless of current isolation level to ensure that ANSI/SQL 92 standard RR behavior is possible for SELECT statements that may be executed by other transactions.

Next key locking can be disabled for an individual database in UDB by setting the DB2_RR_TO_RS flag to YES for the instance(s) containing the database(s) effected. This disables the next key locking for all applications accessing all databases within the specified instance(s). DB2 has recommended using a single database per DB2 instance in order to minimize the impact on other applications.

The DB2_RR_TO_RS flag may be set in the registry using the db2setcommand from a shell command prompt on your DB/2 server before starting DB/2. For example:

db2set DB2_RR_TO_RS=YES -i <DB2 instance>

If any applications that share these databases with CB require the ANSI/SQL 92 standard RR isolation, then setting the DB2_RR_TO_RS flag may not be a viable solution to this problem. If that is the case your CB application(s) must be modified to handle encountering a potential next key deadlock condition. Contact your IBM sponsor for assistance, referencing defect 49296.

In some circumstances when doing multiple client work, two or more clients may appear to be waiting for the same object and the clients will seem to hang. Terminating the applications leaves the database in an inconsistent state and locks are outstanding.   All Platforms

There is an option in DB2 to limit the amount of time an application will wait for a lock, the default is infinite. To set the wait time, use the following DB2 command:

UPDATE DATABASE CONFIGURATION FOR CLAIMDB
USING LOCKTIMEOUT xx

(where xx is the amount of time in seconds). This will cause DB2 to generate the following message when the timer is exceeded:

SQL0913N Unsuccessful execution caused by deadlock or timeout.

This will be mapped to a CORBA::PERSIST_STORE exception by Component Broker and returned to the client application. Additional information will be logged in the activity log and this exception can be caught and dealt with in the application appropriately.

Often the client can just retry after a rollback and continue on. (See the next key lock description above for additional suggestions).

Understanding xa_close messages	When a server has been connected to an XA-compliant database such as DB2, messages such as	All Platforms

Server testsrv received an error from xa_close when closing a connection to the XA Resource Manager database CBSRFCDB. The return code was XAER_RMERR.

will be written to the activity log when the server is shut down.

These messages are harmless and may be safely ignored.

Clients

General

Understanding DCE "gss authentication" messages	The following DCE message may appear on the client's console:	All Platforms

PID XXXX ERROR gss authenticatoin
msgID = 0x0DCEA113
Can't get server ticket in gss_init_sec_ctx

This message may be ignored.

Debugging on an AIX client	Default settings for data size, physical memory, and stack size may be insufficient when using the debugger on an AIX client. The debugger will fail if these settings are insufficient.	AIX

To unset these settings, add the following lines to your login script file (.profile):

ulimit -d unlimited # to reset limits on data size
ulimit -m unlimited # to reset limits on physical memory
ulimit -s unlimited # to reset limits on stack size

Also, minimize the use of paging space while running the debugger. The AIX command "lsps -a" reports the current paging space usage.

ActiveX Client

Installing the ActiveX client	For this release, the ActiveX client requires that Microsoft Internet Explorer 3.0 or higher is installed.	Windows NT

Internet Explorer 3.0 is provided as an optionally installable feature of Windows NT Version 4.0. If Internet Explorer is not currently installed, install it prior to attempting to install the ActiveX client.

Generating interfaces using the IDL2COM tool	When an IDL file is processed by the IDL2COM tool, a unique GUID (-g parameter) is required. This GUID is used to register within the Windows system registry, the various interfaces, etc produced by IDL2COM. Many interfaces may be contained within the IDL, and IDL2COM uses the provided GUID parameter as a starting point for the to-be registered interfaces. If multiple items produced from the IDL file need to have GUIDs, IDL2COM increments the 7th and 8th digits (31 in the example below) as needed. So if the IDL file results in three interfaces being registered, the following GUIDs would be used:	Windows NT

AE3E2131-C6DE-11d0-92AF-08005ACE818D
AE3E2132-C6DE-11d0-92AF-08005ACE818D
AE3E2133-C6DE-11d0-92AF-08005ACE818D

This is important to know since all registered items need to be unique within the registry. Use the guidgen.exe program supplied by Microsoft to provide the -g parameter, but be careful not to conflict with the internally-generated values that IDL2COM will use based on the -g parameter.

Be sure to include a space between the -g and the GUID value on the IDL2COM command line. For example,
idl2com -v -g AE3E2131-C6DE-11d0-92AF-08005ACE818D myIDL.idl

Mapping data types	The IDL2COM tool has the following limitation: • IDL constants of datatype float or double are not supported.	Windows NT
Generating, registering or unregistering DLLs	To produce a DLL containing a COM/OLE Automation interface to a CORBA object, do the following steps: 1. Use the IDLC emitter to produce the client side bindings from the IDL, using the -mcpponly flag on the command. For example, *idlc -mcpponly -suc:hh Policy.idl* 2. Use the IDL2COM tool to produce the COM/OLE Automation files from the IDL. 3. Use nmake -f [filename] to compile and link the COM/OLE Automation DLL for the CORBA object. Use the IDL name, appended with the .mak extension for [filename]. 4. Use regsvr32 [filename] to register the DLL in the Windows registry. Use the IDL name, appended with the .dll extension for [filename]. 5. Use regsvr32 /u [filename] to unregister the DLL if no longer needed. Use the IDL name, appended with the .dll extension for [filename]. If the DLL produced in step 3 is to be moved to a different directory, do the following steps: 1. Unregister the DLL. 2. Move the DLL file and its corresponding TLB file to the new directory. 3. Reregister the DLL. When producing the DLL for the CORBA object, ensure that all required IDL for the object (including IDL referenced by the object's IDL) has been processed by the IDL2COM tool into the same directory. This ensures that the correct header and library files are available when the .mak file is processed.	Windows NT
Using COM or OLE objects	Conformance to the OMG's COM-CORBA Interworking Part A specification is not complete. CORBA objects can be accessed through COM and OLE automation-produced interfaces, but CORBA objects can not access COM or OLE objects.	Windows NT
Using OLE automation interfaces	While the produced OLE Automation interfaces are intended to be generic OLE automation interfaces available to any OLE controller, only Visual Basic 4.0 has been used to test these at this time.	Windows NT
Using remote CORBA objects	During installation, several OMG COM-CORBA Interworking Specification interfaces are installed and registered. Of key importance are the CORBA Factory interfaces: GetObject and CreateObject. Use these interfaces to get started using remote CORBA objects. The samples provide guidance on using these interfaces.	Windows NT
Using IBM-supplied COM wrappers	During installation, several pre-built COM/OLE Automation interface DLLs for some of the Object Services are provided and registered for you. These DLLs are contained within the installed `bin` directory, and supporting TLB, LIB, and header files are installed as well.	Windows NT

The specific list of DLLS which are shipped is contained in the `RegActX.bat` file found in the installed `bin` directory. This file registers the DLLs.

Using VBScript and Internet Explorer	A sample is provided that shows the use of a CORBA object from VBScript and Internet Explorer. This sample is available in the samples subdirectory (InstallVerification/ProgrammingModel/Applications/ActiveX) under the directory where Component Broker was installed. Only the following data types may be passed to a CORBA object from a VBScript script: • primitives (ie. long, short, char, boolean, etc) • strings • objects • structures containing only primitives, strings or objects • unions containing only primitives, strings or objects Do not use arrays or sequences in a VBScript for this release.	Windows NT
Accessing AIX servers	In this release, an ActiveX client does not communicate with an AIX server.	Windows NT

Java Client

Accessing ORB and object services API documentation	HTML documentation for the ORB and object services bindings is available by executing the self-extracting ZIP file apidocs.exe, located in the subdirectory doc\client\javacl under the directory where the Java Client is installed. Using a browser, open the HTML file "packages.html" to view a listing of available Java packages for the Java client API.	Java
Finding the somojij.zip file on AIX	In the 1.3 release, the somojij.zip file was moved to a different directory on AIX. It may now be found in the $IVB_DRIVER_PATH/lib directory. Check any scripts that modify **CLASSPATH** to ensure that the correct directory is used.	Java
Compiling IDL files	Customer-written IDL files that #include IBM-provided IDL files such as orb.idl, Cos<xxx>.idl, IExtended<xxx>.idl may cause the IDL-Java compiler to issue error messages to the effect of "IDL file not found" if the compiler is run on a machine on which only the Java client SDK installation option was chosen. This is because that installation option does not install the global set of IBM-provided IDL files on the target machine in this release. There are two workarounds available: 1. Copy the contents of the include directory (or at least the *.idl files in that directory) from a C++ client machine or a server machine, to a suitable directory on the Java client development machine. (This only needs to be done once per Java client development machine.) When running the IDL-Java compiler, instruct the compiler to search that directory for #include files by specifying that directory after the -I option, for example: java com.ibm.idl.toJava.Compiler -I <dirname> <other options> <idl file> or, 2. Copy the customer IDL file to a machine on which the C++ client is installed, or a machine on which a Component Broker server is installed, and emit the Java bindings using the above syntax using the -I option. Then copy the resulting .java files back to the Java client development machine.	Java

Using a downloaded Java client	Downloading a Java client causes the Java ORB classes to be downloaded. Due to Java security, these classes may only open socket connections to the same physical IP address from which the classes were downloaded. This means that ORB requests may only flow to the web server.	Java
	Placing all the resources that the Java client needs to access at the same physical IP address as the web server allows the downloaded Java client to fully participate in the Component Broker environment. This means placing the web server, the Component Broker Connector server (including the Location Service Daemon and the Common Naming Service bootstrapping "agent" daemon) and all Application, Business, and Data Objects to be accessed by the Java client at the same physical IP address. Delegated requests to third tier resources are supported, as long as the Java client does not have to directly access the third tier resource.	
	A second approach is to set up a "redirector" on the web server to redirect CORBA interactions between the client and the server. This approach tunnels the CORBA (IIOP) request by wrapping it inside a HTTP request and sending the request to a Java servlet on the web server. The servlet then makes a connection to the Component Broker Connector server and forwards the CORBA request. The reply flows to the servlet, which then returns the reply to the client. This approach incurs a performance penalty; therefore, it should not be used for applets that require a high volume of client/server interactions. Additional information on this approach is available in the online documentation.	
	To run the Java client as a downloaded applet, other Component Broker clients (such as the C++ client or the ActiveX client) may not be installed on the machine to which the Java client is being downloaded. The Java client will fail if run as an applet on a machine where another Component Broker client is installed.	Java
	Before downloading the Java client to run as an applet, check to ensure that no other Component Broker client is installed.	
	Since only one Object Request Broker (ORB) is started per Java Virtual Machine (JVM), the ORB will be shared among all applets running in the JVM. Applets intended to run with other applets in the same JVM should be designed to efficiently use the shared ORB.	Java
Understanding error messages when running as an Applet	The following error messages will be generated when using the Java client as an Applet:	Java
	Security exception: netscape.security.ForbiddenTargetException: User didn't grant the UniversalConnect privilege.	
	Security exception netscape.security.ForbiddenTargetException: User didn't grant the UniversalPropertyRead privilege.	
	Security exception on user.home netscape.security.AppletSecurityException: security.checkpropsaccess.key	
	Security exception netscape.security.ForbiddenTargetException: User didn't grant the UniversalFileAccess privilege.	
	Exception creating logfile directorynetscape.security.AppletSecurityException: security.checkread: Read of 'AppletRASLogs\CBUser' not permitted	
	The client will still operate properly when these messages appear.	
Using QuickTest applications on AIX	Due to a problem in Java on AIX, when running Quick Test generated clients, there may be problems with the focus on text fields. Although a text field seems to have focus, typing may appear in a different text field.	Java / AIX
	To correct this, select a different object such as the pulldown menu next to the text field. Focus will now follow where it is requested. This is a limitation in	

Java and cannot be fixed in Quick Test.

Using the Remote Debugger from a Java client	The user interface for the Remote Debugger requires the ICLUI class libraries provided with the VisualAge C++ compiler. If the Java client machine does not have these libraries, the Java client code on this machine may not be debugged locally using this debugger.	Java
	The Java client code may be debugged remotely, where the Remote Debugger user interface is used on a machine that has the ICLUI class libraries.	
	Alternatively, the VisualAge C++ compiler may be installed on the Java client machine, thereby installing the necessary ICLUI class libraries to support local use of the Remote Debugger.	
Using the Java client on Windows 95 with SSL	To use the Java client on Windows 95 with SSL, the **CLASSPATH** environment variable must be set to include the following: [drive]:\cbroker\data\keyrings where [drive] is the drive where the Java client is installed.	Java

CICS / IMS Application Adaptor

Updating the Quick Beginnings Guide	The *CICS and IMS Application Adaptor Quick Beginnings* guide should be updated with the following changes:	Windows NT

- On page 48, in the section "Configuring the Application with System Management", add the following step after step 2d:
  Change the Security Mechanism to IMS.
- On page 52, in the section "Creating CICON Transaction Objects", add the following step prior to step f:
  On the Advanced Pages (page 2), select the MSG field under the literal column and change "PRESS CLEAR TO EXIT" to nil (as defined in the DFHDGA2 field). This will allow the program to query the status of this field to check whether or not the create of the record in the CICS datastore has succeeded. In the failure case, this field will return a "DUPLICATE RECORD" message.
- On page 64 in the section "Creating the Other CICON Transaction Objects", in the bullet for the creating the transaction object for retrieve, do the steps in the following order:
  1.
  4.
  2.
  3.
  5.
  6.
  7.
- On page 91 in the section "Define on a Procedural Application Object", add the following at the end of the section:
  In addition, this constructor also needs to set the LogonInfoClassname. For example, code similar to the following is needed:
  ECI.setLogonInfoClassName(
  "paa.mysamples.cics.eci.acct.BeCashAcctLogonInfo")
- On page 93 in the section "Importing the Bean", replace the first sentence with:
  The bean to import is the BeCashAcct from the paa.mysamples.cics.eci.acct package.
- On page 93 in the section "Importing the Bean", replace step 3a with:
  Type paa.mysamples.cics.eci.acct.BeCashAcctPAO in the Class Name field.
- On page 151, Appendix B, "When you to import a package..." should be "When you need to import a package...".

Using CICS/IMS applications	Applications that access CICS or IMS will work better when the Java VM heap size is increased from its default 32 MB setting to 64 MB.	Windows NT

To change the heap size, do the following steps:

1. Open the System Management User Interface.
2. Select the application server running this application, right click on it and select **Edit**.
3. In the window, select the Java VM Machine page.
4. Enter 64000000 in the "maximum heap size" field.
5. Click on the **Apply** and **OK** buttons.

Finding missing TO files	For the CICSHOD sample, three files are missing:
	  MenuCustomerRetrieveTO.java
	  MenuCustomerUpdateTO.java
	  MenuCustomerInsertTO.java
		Windows NT

These files are available from the IBM Component Broker Support web site. Contact your IBM sponsor for assistance in obtaining these files.

Place these files into the directory
CBroker/samples/InstallVerification/PAA/BusinessObjects/Acct/
Working/NT/paa/samples/cics/ivp/menu/
and proceed with the directions in the *CICS and IMS Application Adaptor Quick Beginnings* guide to compile and run the sample.

Creating a CICON Transaction Object	The *CICS and IMS Application Adaptor Quick Beginnings* guide, chapter 5 is missing a critical step. In section "Creating the Other CICON Transaction Objects" on page 64, add a step 8 - " In creating MenuCustomerRetrieveTO class user must select MSG from the OUTPUT page for both Menu2 and Invalid transaction records."	Windows NT

This is needed for the application to retrieve the output message from the screen and determine the next step.

Developing objects involving the CICS or IMS application adaptor	When compiling objects developed to use the CICS or IMS adaptor, the current directory must be specified in the **CLASSPATH** environment variable for the object to build correctly.	Windows NT

Ensure that the current directory is specified in the **CLASSPATH** environment variable by including a period (".") in the variable's value, separated from other directories using the semicolon (";").

Handling embedded NULLs	CORBA IDL does not allow embedded NULLs in Strings. However, values retrieved from CICS via ECI may contain such NULLs.	Windows NT

A value containing an embedded NULL will generate a DATA_CONVERSION message.

To handle this case, a converter routine is needed to either truncate the String at the embedded NULL or to replace the NULL with some other character. The following sample demonstrates what such a converter routine must do:

```
public class MyConverter {
public static String trimNull(String instr) {
String retval = instr;
int ind = instr.indexOf(0);
if (ind != -1)
retval = instr.substring(0,ind);
return retval;
};

public static String replaceNull(String instr,
char newchar) {
String retval =
instr.replace(java.lang.Character.forDigit(0,0),
newchar);
```

```
return retval;
};
};
```

This routine must be placed into the **CLASSPATH** as MyConverter.java and compiled using the Java compiler.

To use this converter, do the following steps:

1. Add the following line to the yyyyyPAO.java file:
   import MyConverter;
2. In the PAO, under the retrieve method, and after the data has been successfuly retrieved, modify the fields to be converted by adding the following code to replace the null with a space:
   setXxxxx(MyConverter.replaceNull(getXxxxx(),' '));
   or to trim the string at the null:
   setXxxxx(MyConverter.trimNull(getXxxxx()));
   where Xxxx is the name of the PAO attribute.

Managing connections	If a CRUD method is called outside of the scope of a Session, a connection is allocated during the execution of the method. At the conclusion of the method, the connection is marked for garbage collection by the Java virtual machine. Since the connection may not be garbage collected immediately, it will count towards the connection limit. Especially for IMS connections which allow only one connection at a time, this causes subsequent CRUD method calls to hang while waiting for a connection. To avoid this problem, always call a CRUD method within the scope of a Session.	Windows NT
Using the resetSession method	In this release, the resetSession method on the session control does not reset the values to the last checkpointed values.	Windows NT
Loading PAA DDL	When loading a PAA Application, or when loading the DDL for a PAA application, the following warning message may appear: *Hostl.<YourHost>/ApplicationFamily.iDB2IMApplications (line): The object which was expected to already exist is missing.* So long as the warning involves iDB2IMApplications, it may be ignored. The application or the DDL loads successfully. To avoid receiving this warning, remove the following two lines from the DDL for the PAA application: ApplicationFamily.iDB2IMApplications; ApplicationFamily.iDB2IMApplications/Dll.somids1i;	Windows NT
Installing an application on multiple servers that uses APPC	An application that uses APPC communication must be configured with the Local Logical Unit (LU) name of the server. Since multiple servers cannot share the same LU name, additional configuration must be done after server activation. Once activation is complete, modify one of the servers' image to specify a different LU name for the APPC connection configured for that server.	Windows NT
Handling initial 3270 screen navigation	When using 3270 communications, it is possible for the navigation script's sends and receives to get out of sequence, i.e. a reply is sent before the request is received. The first interaction of a navigation script normally assumes the initial screen will be "clear" and that the script will initiate a CICS transaction by sending a script including a transaction identification. If the initial screen is not "clear", the problem may appear, depending upon system response times.	Windows NT

To force a wait for the initial screen to be fully received, set the ScreenOnConnect attribute of the HODConnectionSpec to true. The technique for setting parameters of the HODConnectionSpec is described in the subsection "Define on a Procedural Adaptor Object" contained under the section "Connection Specifications in the Unit Test Environment" in the *CICS and IMS Application Adaptor Quick Beginnings* manual.

Using Optimistic inserts or deletes	In this release, optimistic locking over APPC for insert or delete operations is not supported.	Windows NT
Building the IMS APPC sample	After installing the typical server installation option with the PAA runtime and samples, move the file	Windows NT

c:\CBroker\samples\InstallVerification\PAA\BusinessObjects\ IMSAPPC\Working\NT\DelTO.java

to the

c:\CBroker\samples\InstallVerification\PAA\BusinessObjects\ IMSAPPC\Working\NT\paa\samples\ims\appc\pbe

directory before building the sample.

Understanding APPC messages	Several messages related to APPC connections to third-tier systems were not included in the *System Administration Guide*. These messages are described in the online help, however.	Windows NT
Using the IMS APPC sample	Please note the following changes to the *CICS and IMS Application Adaptor Quick Beginnings* manual in the section "Developing an IMS APPC Application":	Windows NT

- In the section "Creating Other CICON Transaction Objects", the DelTO Transaction Object should have "DELETE" as the Parameter constant on the Transitions Page.
- In the section "Defining the Business Object", the implementation of the showAll user-defined method was omitted. After defining the method and hitting Finish, choose the PhoneBookRecBO business object, then choose showAll() under the User-Defined Methods in the rightmost panel. Under the Source panel at the bottom, add the following 4 lines of code into the showAll() method:
  lnm = iDataObject->lastName();
  fnm = iDataObject->firstName();
  ext = iDataObject->extNumber();
  zip = iDataObject->internalZip();

In this release, when using the IMS APPC sample, fields retrieved through the IMS APPC adaptor are padded to their defined field length with blanks. Since the adaptor compares the retrieved key field with the requested key value, the comparison will fail when the requested key value is shorter than the key field length. This causes the persistent application object associated with the requested key value to not be filled with the correct data from IMS.

Windows NT

In the IMS APPC sample, this problem occurs particularly with the LastName field in the IMS Phone Book IVP, where the LastName field is defined to be 10 characters. When a shorter name (such as "SMITH") is specified, the returned data is associated with the padded key ("SMITH   "), which fails to match the requested key.

This problem may be avoided by taking either of the following steps:

- In the client program, ensure that the requested key value is always the same length as the key field.
- Modify the setLastName method in the PAO object to insert the line
  lastName = LastName.trim();

at the very beginning of the method body.

Understanding non-zero IMS return codes	The tier3 IMS console will show non-zero return codes (such as 18 and 4) when the transaction in the IMS APPC sample completes against the IMS phonebook IVP. These return codes result from differences between the way Component Broker and IMS commit transactions and deallocate conversations. Similar non-zero return codes may occur with other IMS transactions. These return codes may be ignored, since they have no effect on the processing.	Windows NT

Oracle Application Adaptor

Configuring the default database	When configuring the default database that is created as part of the Oracle Application Adaptor installation, there is an SQL Plus 8.0 restriction that affects the configuration. Step 11 in the section "Configuring an Oracle Database" in the "Installing the Component Broker Application Adaptor for Oracle" chapter of the *Oracle Application Adaptor Quick Beginnings* manual should read as follows: 11. Create any required tables in the user schema. For the Oracle sample application, create the OPolicy table in the OPolicy database: a. At a Windows NT command prompt, enter:	Windows NT

```
   copy x:\CBroker\samples\EarlyTestFunction\OAA\BusinessObjects\
   OPolicy\Working\NT\OPolicy.sql x:\CBroker\samples
```
  Note: This step is necessary because SQL Plus 8.0 limits commands to 79 characters.
    b. In an Oracle SQL Plus 8.0 window, enter:
```
      @x:\CBroker\samples\OPolicy.sql
```

VisualAge Java and CICON information

This release of Component Broker includes a version of VisualAge Java that includes support for CICON, a tool for mapping 3270 screens into JavaBeans. Useful information about these tools is included in the readme.htm file provided with the tools. Included below is a portion of that information that is particularly relevant in the Component Broker environment.

General VisualAge Java information

- While installing this version of VisualAge Java, we recommend making a backup copy of the file ide.icx. To create a backup copy, do the following steps:
  1) Change to the directory to where VAJava was installed.
  2) Change directory to ide\program.
  3) Copy the ide.icx file to ide.old.
- Do not install to an HPFS network or local drive (since Windows NT4.0 has trouble handling long file names on such a drive).
- You must have at least 2MB free on your Windows system drive, and must have your environment variable TEMP or TMP pointing to a valid temporary directory with at least 5MB free. This also applies to the UnInstall program.
- Windows NT4.0 has a registry path restriction of 511 characters.
  If your current path is greater than 450 characters then you should remove entries from it prior to installing.
- TCP/IP must be installed and configured in order for IBM VisualAge for Java to function properly. TCP/IP must be enabled as follows:
  1) For a LAN Adapter configuration:
  - You must have DNS enabled with a valid host and domain name.
  - Your LAN DNS must resolve "localhost" to 127.0.0.1.
  - You cannot run disconnected with a LAN adapter configuration.
  2) For a Dial-Up Adapter configuration:
  - You must have DNS disabled.

- Your TCP/IP Address must be obtained automatically.
Note: these configuration options will apply to all TCP/IP adapters even though they have only been changed for this one. You will not be able to use both LAN and Dial-Up without reconfiguring.

Dial-Up networking TCP/IP properties for your internet service provider(s) (ISP) must be configured as documented by the ISP. The Dial-Up networking TCP/IP properties will override the properties in the Dial-Up Adapter TCP/IP properties configured via the 'Network' icon in the Windows 95 Control Panel. The overriding of the properties will only take place so long as the Dial-Up Adapter TCP/IP properties are configured as above. Enabling the DNS in the Dial-Up Adapter TCP/IP properties or setting an IP address in the Dial-Up Adapter TCP/IP properties will interfere with the Dial-Up networking configuration for the ISP and must be avoided.

- If you are running standalone, you can also enable the MS Loopback Adapter without the other two adapters.
- If you get a message that indicates that the install has detected a "Shell Extension" for Windows NT, the install will not be able to proceed and will terminate. In order to get VisualAge for Java installed, you should perform the following tasks:
1) Make sure you have an Emergency Recovery Disk. Instructions for
creating this are available in the Windows operating system
documentation.
2) Invoke "regedit.exe" from a command prompt.
3) Expand the key
\\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
4) Select the SHELL name in the name/data pairs for the above key.
5) Make a note of the 'data' recorded for this 'NAME' as you will need it after installing
IBM VisualAge for Java.
6) Select Edit->Modify from the menu bar pulldown for the SHELL name/data pair.
7) Set the value for the SHELL name to Explorer.exe, and then select the OK
pushbutton.
8) Select Registry->Exit from the menu bar pulldown.
9) Restart and complete IBM VisualAge for Java installation.
10) Once installation is complete restore the previous registry entry as follows:
- Invoke "regedit.exe" from a command prompt.
- Expand the key
\\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
- Select the SHELL name in the name/data pairs for the above key.
- Select Edit->Modify from the menu bar pulldown for the SHELL name/data pair.
- Restore the value for the SHELL name to the value that was recorded in Step (5),
and then select the OK pushbutton.
- Select Registry->Exit from the menu bar pulldown.

Specific CICON information

- When parsing BMS or MFS files from the Parser Page of the Settings notebook, always close the parser output window via the top left button of the window. **Do not close the window with the OK button or allow the window to remain open when exiting VisualAge for Java**. This will result in the inability to parse BMS or MFS files from the Parser Page on subsequent entry into the VisualAge for Java environment.

If this problem occurs, do the following steps to recover:
1) Version each project to be retained.
- Go to the Workbench window.
- Click on the Projects tab.
- Select all the unversioned projects that are to be retained.
- Click on Selected>>Version...
- Select Automatic to allow the system generate version names.
- Click on Finish.
2) Exit VisualAge for Java
3) Change directory to the root of the VisualAge for Java install tree
4) Change directory to ide\program
5) Delete the ide.icx file
6) Copy the ide.old file to ide.icx
7) Start up VisualAge for Java.
8) Add back the versioned projects from step 1
- Go to the Workbench window.

- Click on the Projects tab.
- Click on Selected>>Add Project...
- Select Add project(s) from the repository
- Click on Browse...
- Select the projects and then select the desired edition you wish to add.
Move them to the right side. Click OK.
- Click on Finish

- The generation of a transaction object method can only be done once its bean class has been saved.
- The generated transaction object method may not be visible in its class. Should this occur, refresh the Workbench by going to the Window pull down menu and selecting Refresh.
- There is no online help from within the development dialogs.
- Transactions used via ECI have to be compiled with an IBM COBOL compiler.
- LogonLogoff class and LogonInfo class have to be specified with full package name in a connection specification property sheet .
- When implementing a logon method of a LogonLogoff class for an IMS 3270 transaction , only navigate up to a non-conversational state.
- For RPC records setting constant values on a transition page is only possible if the underlying data type is string. If not you have to leave it nil, and by that propagate it into the transaction object method interface, where you can assign a type correct value. There is no converter support for RPC transaction record. For ANY, CICS, and IMS transaction records converter support is limited to the following converters:
string --> java.lang.Boolean
string --> java.lang.Integer
string --> java.lang.String (by delimiter, by offset, by trim, and by substitution)
The customizer for Connection and Interaction specifications do not prevent the entering of not supported values. Property editors would be necessary to prevent this. We will have a list containing valid values in the future.
- When the attribute type on the Business Object differs from the type retrieved from the backend system, do the following steps to match the types
1) Define the set of PAO properties (key and general) which match the types
in the Business Object.
2) For each general property defined in step 1, define a second PAO property
which matches the type of the backend system.
3) For each property defined in step 2, remove its field definition and change
the code of the accessor method to convert from/to the fields defined in step 1.
4) In the connection view, map the fields to the general properties defined in
step 2, and the key attributes defined in step 1.
- Projects in the Workbench that do not have a corresponding directory name in <VisualAge for Java root directory>\ide\project_resources will cause two problems: An exception to be thrown when renaming them and a dialog appearing when exporting them. To avoid this java.lang.InternalError exception and the dialog ensure that the name of the project is listed as a directory in
<VisualAge for Java root directory>\ide\project_resources, and that it contains the appropriate resource files.
- There are five unresolved problems associated with classes found in the Workbench. This problem of AAract method may not be private, final, synchronized, static or native can be ignored. The classes exhibiting this error are:
ibm.cics.jgate.client.JavaGatewayInterface
ibm.cics.jgate.server.ProtocolHandler
java.text.Collator
- Importing the package COM.ibm.ivj.examples.vc.beandemo 1.0 into the Workbench without the packages sunw.demo.juggler 1.0 and sunw.demo.molecule 1.0 will create many unresolved problems.

For other useful information on creating CB Procedural Adapter Objects, how to get non-mapped output, i.e. output not connected to PAO attributes, ways of defining a connection specification, and building an ECI scenario for CICS, see the readme.htm file in the root of the VAJ/CICON CD.

IMS configuration information

The following information is useful in setting up IMS to be accessed from Component Broker:

Two idle IMS Message Processing Regions (MPRs) must be available for use by the IMS transactions used by that DO. If a CB application uses 10 DOs that are in IMS this means 20 IMS regions (MPRs) will be used by the CB initiated IMS transactions. Failure to have enough MPRs available will cause some CB DO requests to remain queued for execution within IMS which will prevent successful execution of the CB application.

The IMS system definition (IMS gen stage 1) must include a TRANSACT macro defining the transaction code used for the CB DO request and this must have the operands MAXREGN and PARLIM specified appropriately. MAXREGN must be greater than 2*(# of DOs in IMS for the CB application) and PARLIM=0. You can also specify MAXREGN=0 which allows an arbitrary number of regions to be used. The SCHDTYPE=PARALLEL operand must also be specified on the corresponding APPLCTN definition.

The CB application uses a distributed syncpoint. This will cause the IMS transactions representing the DOs to be idle but still scheduled in their dependent regions until the commit point is reached for the entire application. The requirement to have a sufficient number of IMS MPRs available stems from this need for all IMS transactions that are initiated by the single CB application instance to be concurrently executing. Having too few MPRs will cause the CB application instance to never complete (waiting for a new MPR to start to execute the DO). The application programs making up the DO transactions in IMS are very light in resource usage compared to most IMS transactions -- they are waiting much of the time. These new MPRs that must be added to the IMS system (or existing ones reserved for use by the CB applications) are less of a resource impact on the performance and capacity of the IMS system than other MPRs. The primary impact is in Real and Virtual storage use with much less impact on CPU and IO workload. Storage tends to be plentiful on recent-vintage machines; this mitigates the impact of the additional storage used as it is a plentiful resource.
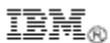
The IMS transaction program implementing the Data Object access to IMS controlled data must by defined with a suitable PROCOPT in the PSBGEN and must issue suitable locking for the data segments being processed. This is very important for distributed sync point to work correctly as the IMS data retrieval and data update transactions need to sync together. IMS has added special processing to recognize all the transactions that are part of the same distributed logical unit of work and automatically propagates the locks from the first transaction to the second one in order to have a consistent unit of work and to prevent

deadlocks. This simplifies the application program responsibility for data access control but still leaves two responsibilities. These are the processing option (PROCOPT parameter in PSBGEN) and locking of data by the retrieval transaction to prevent other updaters.

The exclusive processing option (PROCOPT=E) must never be specified for CB transactions! This option specifies that only one transaction can be running at a time accessing the database; this prevents the second CB transaction from being concurrently processed and prevents reaching the coordinated sync point. Almost always you should specify PROCOPT=A for both transactions (this allows all update, insert and delete activity). Using other processing options, particularly PROCOPT=GOx, can result in deadlock or waiting for locks held by other (non-CB) transactions and should be done only after careful analysis of the impact. It is valid in some special situations but usually has undesirable results.

The first transaction doing the IMS data retrieval should lock the data for update by issuing a GHU or GHN call. The update is done by the second update CB transaction and the locks obtained by the GHx call are transferred to this second transaction as part of the special IMS support for distributed sync point. This is a new function in IMS and only for transactions using distributed sync point (LU 6.2 sync level of syncpt). Other transactions doing data retrieval continue to be prohibited from using the hold form of get calls if you want to avoid lock interference and its adverse performance impact.

There is a possibility of deadlock or lock conflict for inserted data. This varies widely by database definition (both logical and physical) and database type. In general inserting data next to updated data is safe (the locking for update will also implicitly lock the insert) but conflicts in free space access can still occur. Normal IMS tuning and lock conflict diagnosis techniques apply with no special considerations needed for CB.