### Table of Contents

[Table of Contents 1](#_Toc469299582)

[IVI Shared Component Installer Revision History 2](#_Toc469299583)

[Version 1.0.4 2](#_Toc469299584)

[Version 1.0.7 3](#_Toc469299585)

[Version 1.0.8 3](#_Toc469299586)

[Version 1.0.9 3](#_Toc469299587)

[Version 1.0.10 3](#_Toc469299588)

[Version 1.1.0 4](#_Toc469299589)

[Version 1.1.1 4](#_Toc469299590)

[Version 1.1.2 4](#_Toc469299591)

[Version 1.1.3 5](#_Toc469299592)

[Version 1.2.0 5](#_Toc469299593)

[Version 1.2.1 5](#_Toc469299594)

[Version 1.3.0 5](#_Toc469299595)

[Version 1.4.0 6](#_Toc469299596)

[Version 1.5.0 6](#_Toc469299597)

[Version 1.5.1 7](#_Toc469299598)

[Version 2.0.0 7](#_Toc469299599)

[Version 2.1.0 8](#_Toc469299600)

[Version 2.1.1 9](#_Toc469299601)

[Version 2.2.0 9](#_Toc469299602)

[Version 2.2.1 9](#_Toc469299603)

[Version 2.3.0 10](#_Toc469299604)

[Version 2.4.0 10](#_Toc469299605)

[Version 2.4.1 10](#_Toc469299606)

[Version 2.4.2 11](#_Toc469299607)

[IVI Cleanup Utility Revision History 12](#_Toc469299608)

[Version 1.0.5 12](#_Toc469299609)

[Version 1.0.6 12](#_Toc469299610)

[Version 1.1.2 12](#_Toc469299611)

[Version 1.2.0 12](#_Toc469299612)

[Version 2.0.0 13](#_Toc469299613)

[Version 2.2.0 13](#_Toc469299614)

[Known Issues 13](#_Toc469299615)

### IVI Shared Component Installer Revision History

#### Version 1.0.4

**Updates**

1. Added Click License Agreement.
2. Added updated IviConfigServerCAPI.dll version 1.0.3 to installation.

**Cosmetic/UI Changes**

1. Changed of ‘IVI Standard Root Directory Selection Dialog’ so that an appropriate text message is displayed depending on if the Browse button is enabled or disabled.
2. Disabled the ‘Back’ button in the first Maintenance Mode installation dialog.
3. Changed ‘Cleanup utility’ to ‘Cleanup Utility’ in all dialog text messages.
4. Changed all references to ‘Wise Installation Wizard’ to ‘Installation Wizard’.

**Bug Fixes**

1. On NT systems, the installation was registered as a per-user installation instead of a per-machine installation.
2. On NT systems, if a user without administrative privileges ran the installer, the installer did not abort with an appropriate error message. Instead, the installer attempted to install the product and failed much later in the installation process.
3. On NT systems, if a non-default directory was selected for the IVI Standard Root Directory using the ‘Browse’ button, the ‘Bin’ folder was still installed to the default location ‘Program Files\IVI’.
4. If the VXIPNP directory was not defined on the system, attempting to select another directory using the ‘Browse’ dialog always resulted in a ‘Choose another directory other than the VXIPNP directory’ error.
5. Duplicates of the MSVC compatible import libraries were installed in the ‘Lib\bc’ folder. Borland compatible import libraries for the IVI C Shared, IVI Float, and IVI Configuration Server C API do not exist.
6. On Windows XP, the ‘Exit Installation’ dialog appeared without any bitmaps or buttons. Only moving the mouse over the dialog caused the ‘Finish’ button to appear.

#### Version 1.0.7

**Updates**

1. Added the following files to the installer (installed in the Include folder):

IviEventServer.h

IviEventServer\_i.c

IviEventServerDLL.h

IviEventServerDLL\_i.c

IviSessionFactory.h

IviSessionFactory\_i.c

1. Updated with latest ConfigServer C API (1.0.6.0) and C Shared (1.0.238.0) Component files.
2. Modified installer source to use relative paths for all files.

**Bug Fixes**

1. The master configuration store file would get over-written on a reinstall after the user had performed a 'Partial Cleanup' using the IVI Cleanup Utility.

#### Version 1.0.8

**Updates**

1. Rebuilt with version 1.0.7.0 of IviConfigServerCAPI.dll.

#### Version 1.0.9

**Updates**

1. Rebuilt with version 1.0.239.0 of IviCShared.dll.

#### Version 1.0.10

**Updates**

1. Rebuilt with version 1.0.240.0 of IviCShared.dll.

#### Version 1.1.0

**Updates**

1. Added the IVI Primary Interop Assemblies.
2. Added Type Library Help for all the classes.

**Bug Fixes**

1. On Windows 9x the long path of the Bin directory was added to the system PATH environment variable. This caused the subsequent additions to this environment variable to fail.
2. Running the clean up utility after manually removing “IviEventServer.exe” would cause the clean up utility to crash.
3. IviConfigurationStore.xml and IviConfigurationStore.xsd files would not get repaired if they were missing from the Data folder. IviConfigurationStore.xsd should always be repaired and IviConfigurationStore.xml should only be repaired if it is missing from the Master Config Store location.
4. Upgrading to a newer version of the IVI Shared Components, would not correctly register the IVI Event Server.

#### Version 1.1.1

**Updates**

1. On an upgrade, uninstall the older version at the end of the installation sequence to increase efficiency and prevent any changes to the feature state before the install sequence.

**Bugs That Were Fixed**

1. Upgrading to a newer version of the IVI Shared Components would not install the IviConfigServerCAPI.dll.

#### Version 1.1.2

**Updates**

1. Rebuilt installer to include the following updates to the shared components:
   * IviConfigurationServer.dll: version 1.3.2.4
   * IviCShared.dll: version 1.1.0.0

#### Version 1.1.3

**Updates**

1. Rebuilt installer to include the following updates to the shared components:
   * IviDmmTypeLib.h: Generated based on revision 13 of IDL

#### Version 1.2.0

**Updates**

1. Rebuilt installer to include the following updates to the shared components:
   * IviScopeTypeLib.h
   * IviSessionFactory.dll: version 1.0.1.2
   * readme.txt (for IviSessionFactory)
   * Additional registry entries for IviSessionFactory.dll under HKEY\_CLASSES\_ROOT\Interface\{DE217CF0-2F0C-4EB5-B435-E69400C467EC}
2. Implemented the following changes requested at the January 2004 IVI Foundation Meeting:
   * Added the IVI Cleanup Utility to the IVI Shared Component installation.
   * Registered the IVI Shared Components with the Windows Add/Remove Programs Utility to call the IVI Cleanup Utility in partial mode when invoked from the Windows Add/Remove Programs Utility.

#### Version 1.2.1

**Updates**

1. Rebuilt with version 1.1.2 of the IVI Cleanup Utility.

#### Version 1.3.0

**Updates**

1. Added IviLxiSync 1.0.4.0 components.

#### Version 1.4.0

**Updates**

1. Compiled with version 1.2.0 of the IVI Cleanup Utility.
2. Implemented a custom action to modify the security attributes of the IVI Standard Root directory on Windows Vista to grant full access to all authenticated users (and therefore disable virtualization).
3. Replaced IVI Event Server self-registration (via custom actions) with MSI tables to create/remove the registry entries at install/uninstall time.
4. Modified installer behavior to completely uninstall any previous version before installing the new version. This is needed to prevent IVI Event Server self-registration in older versions from removing registry entries after the new version has been installed.
5. Removed pre-installation of MSI Engine for Win9x (from EXE).

#### Version 1.5.0

**Updates**

1. Changed default directory to <**CSIDL\_PROGRAM\_FILES>**\IVI Foundation\IVI
2. Changed install location of Master Configuration Store and schema files to **<CSIDL\_COMMON\_APPDATA>**\IVI Foundation\IVI. This only applies to clean installations. The file locations remain the same on upgrades.
3. Unlocked the Master Configuration Store file on Vista. The schema file will remain locked if installed to **<CSIDL\_COMMON\_APPDATA>**\IVI Foundation\IVI.
4. Added HKLM\SOFTWARE\IVI IviDataDir registry key value to store location of the 'Data' folder.
5. Updated the following files:
   1. IviConfigServer version 1.6.0
   2. IviConfigServerCAPI version 1.6.0.0
   3. IviCShared version 1.2.0.0
   4. IviCSharedSupport.dll version 1.0.238.0
   5. IviFloat version 1.0.238.0
   6. IviCleanupUtility version 1.2.1.0
   7. Ivi.ConfigServer.Interop.dll version 1.6.0.0
   8. IviPiaRegistration.bat
6. Added the following files:
   1. policy.1.0.Ivi.ConfigServer.Interop.dll
   2. Ivi.ConfigServer.Interop.config
7. Replaced IviConfigurationStore.xsd with IviConfigurationStore\_1-6.xsd.
8. Removed MSVC 6.0 runtime merge modules.

#### Version 1.5.1

**Updates**

1. Updated the IviCShared version 1.6.0.0

#### Version 2.0.0

**Major Update: Updated to comply with version 2.0 of the IVI-3.1 specification.**

Starting from this version, the IVI Shared Component installer will consist of the following two installers:

1. A 32-bit installer for installation of 32-bit shared components on 32-bit operating systems. This installer will not run on 64-bit operating systems.
2. A 64-bit installer for installation of both 32-bit and 64-bit shared components on Vista 64. This installer will not run on 32-bit operating systems.

**Note:** Both installers will always be maintained at the same version number. The term ’32-bit installer’ and ’64-bit installer’ will be used in this document when needed to distinguish between the two. Otherwise, the generic term ‘installer’ will be used to refer to both the 32-bit installer and the 64-bit installer.

**Updates**

The following updates were made to version 1.5.1 (the 32-bit installer):

1. Updated to be compliant with revision 2.0 of the IVI-3.1 specification:
   1. Does not run on 64-bit operating systems.
   2. Installs 64-bit import libraries to the <IviStandardRootDir>\Lib\_x64\msc directory.
2. Updated the following files:
   1. IviConfigServer: version 1.7.0. Includes bug-fixes.
   2. IviConfigServerCAPI : version 2.0.0. This version is a new native C/C++ implementation of the API and is not dependent on the COM Configuration Server.
   3. IviCShared, IviFloat: version 1.7.0
   4. IVI CleanupUtility: version 2.0.0
   5. TypeLibraries: versions 2.0.x.x
3. Modified installation of .NET primary interop assemblies (PIAs):
   1. Removed installation of version 1.0 PIAs from the <IviStandardRootDir>\Bin\Primary Interop Assemblies directory (including policy files). Version 1.0 PIAs are still installed to GAC (including policy files) if .NET framework 1.1 or higher is installed.
   2. Added installation of version 2.0 PIAs to the <IviStandardRootDir>\Bin\Primary Interop Assemblies directory.
   3. Added installation of version 2.0 PIAs to the Global Assembly Cache (GAC) if .NET framework 2.0 or higher is installed.

**Note:** All of the primary interop assemblies (PIAs) installed in the <IviStandardRootDir>\Bin\Primary Interop Assemblies directory have been built with .NET 2.0. These PIAs are also installed in the Global Assembly Cache (GAC). The IVI Foundation recommends that all future development, including rebuilding existing .NET source, use these PIAs.

The older versions of the PIAs, which were built with .NET 1.0 and 1.1, are being installed in the GAC only in order to preserve backwards compatibility at runtime with assemblies that (1) were built with .NET 1.0 or 1.1, and (2) reference older versions of the PIAs. These older versions are not installed in the the <IviStandardRootDir>\Bin\Primary Interop Assemblies directory.

Please note that the older versions of the PIAs are only installed if .NET 1.0 or 1.1 is installed. If you install .NET 1.0 or 1.1 after installing the IVI Shared Components, and wish to use the older versions of the PIAs, perform the following steps:

1. Uninstall the IVI Shared Components from the system by selecting the ‘Partial Cleanup’ mode from the Windows Add/Remove Program or by running the IVI Cleanup Utility.
2. Install the IVI Shared Components.

**64-bit IVI Shared Component Installer**

Implemented the ‘unified’ 64-bit IVI Shared Component installer as required by version 2.0 of the IVI-3.1 specifications. The following are some of the most important features of the installer. For more details, refer to Section 6 of IVI-3.1:

1. Does not run on 32-bit operating systems
2. On a clean install, allows the user to specify the value of both the 32-bit and 64-bit IVI Standard Root directories from the user interface or from the command line. Checks that the specified values always meet the following conditions:
   1. The 32-bit IVI Standard Directory is not equal to or a subdirectory of the 64-bit Progam Files directory.
   2. The 64-bit IVI Standard Directory is not equal to or a subdirectory of the 32-bit Progam Files directory.
   3. The 32-bit IVI Standard Directory is not equal to or a subdirectory of the 64-bit VXIplug&play directory.
   4. The 64-bit IVI Standard Directory is not equal to or a subdirectory of the 32-bit VXIplug&play directory.
   5. The 32-bit IVI Standard Root Directory is not equal to or a subdirectory of the 64-bit IVI Standard Directory and vice versa.
3. Installs all 32-bit files to the 32-bit IVI Standard Root directory.
4. Installs 64-bit import libraries to the <IviStandardRootDir32>\Lib\_x64\msc directory.
5. Installs all 64-bit files to the 64-bit IVI Standard Root directory.
6. Creates identical sets of registry keys in both the 32-bit and 64-bit registry hive.
7. Installs 64-bit PIAs to the GAC if the 64-bit .NET framework 2.0 or higher is installed.

#### Version 2.1.0

**Updates**

1. Added new type libraries and primary interop assemblies to the installer for the following classes:
   1. IviCounter
   2. IviDigitizer
   3. IviDownconverter
   4. IviUpconverter

#### Version 2.1.1

**Updates**

1. Included the following updated type libraries and primary interop assemblies to the installer:
   1. IviCounter
   2. IviDigitizer
2. Removed the IviCounter.chi and IviDigitizer.chi from the installer as those files are now embedded.

#### Version 2.2.0

**Updates**

1. Included new ACPwr type libraries and primary interop assemblies in the installer
2. Added version of IVI Shared Components to the entry in Add/Remove Programs
3. Added build versioning to the installer parts so it is easier for us to know what versions of the installer people currently have
4. Cleaned up various resource information in binaries

#### Version 2.2.1

**Updates**

1. Included performance fixes of IVI C Shared Components (IVICShared.dll) in the installer

#### Version 2.3.0

**Updates**

1. Replaced NI copyright information in header files with IVI copyright.
2. The PIA registration batch file has been updated with an extra parameter -codegen for the regasm tool.
3. Removed the PrimaryInteropAssemblyCodeBase registry keys from the installer per Microsoft's recommendation.
4. Corrected the registry entries for Counter and Digitizer PrimaryInteropAssemblyName to 1.0.1.0.

#### Version 2.4.0

This version was not publicly released by the IVI Foundation and was immediately superseded by the 2.4.1 version.

#### Version 2.4.1

**Updates**

1. Removed support for Windows XP and Windows Vista.
2. All components were migrated to Visual Studio 2015, which results in some changes to generated assemblies. The minimum required version of the .NET Framework (for interop assemblies) is 2.0.
3. Added the Fgen Extensions to the IVI-COM type library. This includes the following new interfaces:

* IIviFgen2
* IIviFgenOutput2
* IIviFgenArbitrary2
* IIviFgenArbitraryWaveform2
* IIviFgenArbitrarySequence2
* IIviFgenTrigger2
* IIviFgenTriggerStart
* IIviFgenTriggerStop
* IIviFgenTriggerHold
* IIviFgenTriggerResume
* IIviFgenTriggerAdvance
* IIviFgenDataMarkers
* IIviFgenSparseMarkers
* IIviFgenSampleClock

Interfaces that end in 2 are revised versions of previously defined interfaces. The current API matches version 5.2 (May, 2016) of the *IVI-4.3: IviFgen Class Specification*.

1. Added a policy file for the Ivi.Fgen.Interop assembly so that references to older versions of the interop assembly will be redirected to the new version. This reflects the fact that the new version of the API is backwards compatible with older versions.
2. Updated IviVisaType.h to:
   1. Include the warning and error code declarations from vpptype.h
   2. Include the error and CVI declarations from visatype.h

#### Version 2.4.2

**Updates**

1. Changed the ThreadingModel of the COM IviSessionFactory to Both.
2. Corrected a casing issue in the RFSigGen PIA file name.
3. In the Fgen IVI-COM type library, moved data and sparse marker Configure methods from the respective collection interfaces to the repeated capability instance interfaces.

### IVI Cleanup Utility Revision History

#### Version 1.0.5

**Bug Fixes**

1. The IviStandardRootDir registry key value would not get deleted on a ‘Complete’ uninstall if the HKEY\_LOCAL\_MACHINE\IVI key had other sub keys.

#### Version 1.0.6

**Updates**

1. Update the product Name field in the file summary page to “IVI Shared Components Cleanup Utility” as specified in the specification.

#### Version 1.1.2

**Updates**

1. Changed positions of OK and Cancel buttons on main dialog of cleanup utility.

#### Version 1.2.0

**Updates**

1. Rebuilt EXE with Visual Studio 2003.
2. Added Application Manifest to:
   1. Use the Windows XP Style common controls
   2. Request for Admin privileges on launch
3. Updated copy right and company information in the version resource.

**Bug Fixes**

1. Some types of exceptions were not properly handled by the application, resulting in unhandled exceptions (crashes) in some cases. Fixed program logic to use return status codes to indicate success or failure instead of throwing exceptions.

#### Version 2.0.0

**Updates**

1. Updated to be compliant with revision 2.0 of the IVI-3.1 specification:
   1. Removes the 64-bit IVI Shared Components on Vista 64 in addition to the 32-bit components in partial and full cleanup mode.
   2. Removes the Lib\_x64 directory from the 32-bit IVI Standard Root Directory in full cleanup mode.
   3. Removes the 64-bit IVI Standard Root Directory and 64-bit IVI registry keys in full cleanup mode.
2. Cosmetic fix: Updated the confirmation message at the end to say if the shared components were ‘partially’ or ‘completely’ installed.

#### Version 2.2.0

**Updates**

1. Updated copyright information to match other binaries. This will now be automatic as we have a formal build process to build this binary

### Known Issues

**February 2015, Version 2.3.0**

The IVI COM Configuration Server, which is also used by many COM instrument drivers, leaks memory. The size of the leak depends on the size and makeup of the configuration data, but the scale of the leak for a relatively modest set of configuration data is 500-2000 bytes. Larger sets of configuration data may leak larger amounts of memory. The leak is repeated each time the COM configuration data is read from disk, so long running processes that repeatedly read or write the configuration store or repeatedly open and close IVI-COM drivers may experience significant leakage over the life of the process. The IVI Foundation is currently investigating the source of the leak.