

What's new in QuickOPC.NET 5.00

Technology

- All components are in managed code, i.e. purely .NET solution.
- Uses strongly typed parameters and properties.
- Designed in accordance with Microsoft recommendations for component development, and Code Analysis (FxCop) rules.
- There is no longer a COM process (middle layer) between your application and the OPC server.
- Windows Vista, Windows 7 and Windows Server 2008 support.
- Unicode is now used throughout the product.

Components

- Added ability to specify percent deadband with subscriptions.
- User interface components (dialogs) are now pure Windows Forms components.
- There is a new dialog for browsing for an OPC property.
- New method `ChangeItemSubscription` allows modification of subscription parameters without having to unsubscribe first.
- New `BrowseNodes` method combines capabilities of both `BrowseLeaves` and `BrowseBranches` methods.
- When browsing for OPC servers, version independent `ProgId` of the server can now be retrieved if available.
- The functionality of `OPCDAComponent` (in `OPCDAControls` assembly), providing event notifications on the UI thread for Windows Forms, has been replaced by the use of standard `SynchronizationContext`. This enables proper synchronization not only in Windows Forms, but in other environments as well, now and in the future.
- All methods are provided with set of useful overloads, simplifying their usage and allowing shorter and more readable code.
- Instead of specifying individual parameters for connection to OPC server, it is now possible to use a `ServerDescriptor` object. Similarly, parameters determining an OPC item can be passed using a `DAItemDescriptor` object. Use of these objects reduces the number of method arguments and makes the code shorter.
- Conversions from `ServerElement` and `DANodeElement` to `String` are provided, simplifying the use of browse results for further method calls.
- When browsing for OPC servers, a `ServerCategories` object is now provided, with information about specifications that the OPC server claims to support.
- New `VarType` value type makes it easy to specify data types used in OPC.

- New DAQuality value type manipulates OPC quality a whole, but also gives easy access to its individual bit fields. For display purposes, DAQuality can be directly converted to a string describing the quality.
- DAVtq object (holding value, timestamp and quality combination) can be converted to a string, fully describing contents of all its elements.
- New DAAccessRights, DABrowseFilter, DADataSource, DANodeFilter, and DAPropertyId types facilitate specifying the values using symbolic constants.
- A new DAGroupParameters object combines together parameter of a subscription.
- Method that deal with multiple elements at once return an array of OperationResult objects or objects derived from it, such as ValueResult or DAVtqResult. These objects contain error (exception) information as well as the actual result.
- Methods that deal with multiple elements at once can be passed a single argument, which is an array of objects derived from OperationArguments, simplifying argument preparation. An arbitrary State object can be used to hold application-defined information for each element, and this State object is then available in the OperationResult objects returned by the method call.
- All global parameters are accessible from your code as properties, and they are no longer stored in registry.
- OPC interoperability has been improved, based on tests with many OPC servers, and participation in OPC Interoperability Workshops.
- One-shot operations use a set of OPC groups separate from those that are created for subscription-based operations, resulting in more predictable behavior.
- Error messages have been improved and enhanced.

Documentation

- Detailed IntelliSense and Object Browser information is available.
- The reference documentation is now provided in standard Visual Studio (MSDN) layout.

Packaging

- OPC Core Components that are redistributed with the product have been updated to a newer version.
- Examples, and other parts that do not directly belong to the core software, have been moved to a separate installation, QuickOPC.NET Bonus Pack.
- Components providing automation (COM) interfaces have been moved to a separate product installation, called QuickOPC.
- The Demo application, supplied with the product, has been enhanced to show many features of the product.

- Easier redistribution: Simply include the QuickOPC.NET assemblies with your application (Visual C++ Redistributables may also be needed).
- "Hidden" licenses are now being supported for OEM usage.

Removed

- Components providing Web service interfaces were removed, and will possibly be re-designed and become a separate product installation.
- Event Log Utility is no longer necessary and has been removed.
- Easy-DA Options Utility is no longer necessary and has been removed.
- OPC Item Generator utility has been removed.
- Support for operating systems older than Windows XP, or Windows Server 2003, has been removed.
- Methods that provide server-wide operations are now considered as "advanced", and were removed.

