Opc.UaFx Namespace

The Opc.UaFx namespace defines the following members.

Classes

Name	Description
IPAddressCollection	Represents a collection of IPAddress objects.
Licenser	Defines mechanism required to register the whole OPC UA Framework Advanced as a bundle (which includes client and server functionality).
OpcAcknowledgeableCondition	Represents a snapshot of a 'Opc.UaFx.OpcAcknowledgeableConditionNode'.
OpcAcknowledgeableConditionNode	Defines a condition node which introduces additional states for acknowledgement and confirmation.
OpcActionMethodNode	Defines a method node using a Action delegate.
OpcAddAnalogItemNode	Defines a single command of the IOpcAddNodesService used to add one analog item node.
OpcAddCommentMethodNode	Defines a method node to apply a comment to a node.
OpcAddDataItemNode	Defines a single command of the IOpcAddNodesService used to add one data item node.
OpcAddDataVariableNode	Defines a single command of the IOpcAddNodesService used to add one data variable node.
OpcAddFolderNode	Defines a single command of the IOpcAddNodesService used to add one folder node.
OpcAddInstanceNode	Defines an abstract base class for a single command of the IOpcAddNodesService used to add one instance node.
OpcAddNode	Defines an abstract base class for a single command of the IOpcAddNodesService used to add one node.
OpcAddNodeCollection	Defines a collection for OpcAddNode instances.
OpcAddNodeResult	Provides information indicating and describing the outcome of an OpcAddNode command execution.
OpcAddNodeResultCollection	Defines a collection for OpcAddNodeResult instances.
OpcAddObjectNode	Defines a single command of the IOpcAddNodesService used to add one object node (nodes of the category Object).
OpcAddObjectNode.TypeDefinition	Refers to a type definition which can be used to define typed OpcAddObjectNode command instances.
OpcAddObjectTypeNode	Defines a single command of the IOpcAddNodesService used to add one object type node (nodes of the category ObjectType).
OpcAddPropertyNode	Defines a single command of the IOpcAddNodesService used to add one property node.
OpcAddReference	Defines a single command of the IOpcAddReferencesService used to add a logical reference between one and another node.
OpcAddReferenceCollection	Defines a collection for OpcAddReference instances.

Name	Description
OpcAddTypeNode	Defines an abstract base class for a single command of the IOpcAddNodesService used to add one type node.
OpcAddVariableNode	Defines an abstract base class for a single command of the IOpcAddNodesService used to add one variable node (nodes of the category Variable).
OpcAddVariableNode.TypeDefinition	Refers to a type definition which can be used to define typed OpcAddVariableNode command instances.
OpcAddVariableTypeNode	Defines a single command of the IOpcAddNodesService used to add one variable type node (nodes of the category VariableType).
OpcAggregate	
OpcAggregateConfiguration	
OpcAggregateConfigurationNode	Defines the general characteristics of a node that defines the aggregate configuration of any variable or property.
OpcAlarmCondition	Represents a snapshot of a 'Opc.UaFx.OpcAlarmConditionNode'.
OpcAlarmConditionNode	Defines an acknowledgeable condition node which introduces additional states to define active, suppressed and shelving alarm conditions.
OpcAnalogItemNode	Defines an analog item node.
OpcApplication	
OpcApplicationConfiguration	
OpcApplicationDescription	
OpcApplicationDescriptionCollection	Represents a collection of OpcApplicationDescription instances.
OpcArgument	Describes an argument used by a method either as an input or an output argument.
OpcArgumentAttribute	
OpcArgumentCollection	Representes a collection of OpcArgument instances.
OpcArgumentGlobalization	Provides mechanisms to control the localization of data and information used by an argument.
OpcArrayDimensions	Provides the lengths of array dimensions.
OpcAttributeOperand	
OpcAutomatism	Defines the behavior of global automatisms implemented in the OPC UA Framework Advanced.
OpcBidirectionalReference	Describes a bidirectional reference from a logical 'parent' to a logical 'child' AND vice versa using a specific type of reference between the two opponents.
OpcBrowseNode	Defines a single command of the IOpcBrowseService.
OpcBrowseNodeCollection	Defines a collection for OpcBrowseNode instances.
OpcBrowsePoint	
OpcCallMethod	Defines a single command of the IOpcCallMethodsService.
OpcCallMethodCollection	Defines a collection for OpcCallMethod instances.
OpcCertificateIdentity	
OpcCertificateInfo	
OpcCertificateJudgementEventArgs	Provides the event data for 'CertificateJudgement' events.
OpcCertificateManager	Provides a set of static (Shared in Visual Basic) methods to manage certificates.

Name	Description
OpcCertificateQueryInstanceEventArgs	Provides the event data for the QueryInstance event.
OpcCertificateRequestedEventArgs	Provides the event data for the CertificateRequested event.
OpcCertificateSettings	Defines the setup of an application instance certificate.
OpcCertificateStoreInfo	
OpcCertificateStores	Provides access to the set of OPC UA predefined certificate stores typically used by an OPC UA application to maintain its certificates.
OpcCertificateValidationFailedEventArgs	Provides the event data for 'CertificateValidationFailed' events.
OpcCondition	Represents a snapshot of a 'Opc.UaFx.OpcConditionNode'.
OpcConditionNode	Defines all general characteristics of a condition.
OpcConditionVariableNode	Defines a variable node which is used by various information elements of a condition that are not considered to be states. However, a change in their value is considered important and supposed to trigger an event notification. These information elements are called condition variables.
OpcContentFilter	
OpcContentFilterElement	
OpcContentFilterElementCollection	Represents a collection of OpcContentFilterElement instances.
OpcContentFilterElementFactory	
OpcContentFilterElementReadOnlyCollection	Provides a read-only collection of OpcContentFilterElement instances.
OpcContext	Provides information and services about the system which belong to the current situation (process, thread, session, etc.).
OpcData	Defines an abstract base class of data objects used and provided in the ecosystem of a specific OPC UA Server.
OpcDataChangeDataSet	
OpcDataChangeDataSetItem	
OpcDataChangeFilter	Defines the conditions under which a data change notification is to be reported.
OpcDataField	Represents a data field declared by the structured data type of a specific OpcDataObject instance.
OpcDataFieldInfo	Provides metadata of a field declared by a data type.
OpcDataItemNode	Defines a specialisation of the OpcDataVariableNode which defines the general characteristics of a data item.
OpcDataObject	Provides a generic and dynamically accessible representation of structured data types.

Name	Description
OpcDataSurrogate	Provides a surrogate used to transport the encoding of data structures its structural information is resolved dynamically at runtime. This information is then transformed into a dynamically determined subclass of the OpcData class. The SDK uses this class in turn to create an instance which is able to represent and understand the encoded structured data. This class serves mechanisms required by the OPC UA Framework infrastructure and it is not intended to be used directly in your code.
OpcDataTypeAttribute	Defines the identifier of a custom data type which is represented by the Type on that this attribute is applied.
OpcDataTypeDictionary	Represents a self-describing block which offers a set of type declarations used for data types and summarizes them within a Namespace.
OpcDataTypeDictionaryNode	Provides a node which contains a set of type descriptions.
OpcDataTypeEncodingAttribute	Defines the information about the encoding used for a custom data type which is represented by the Type on that this attribute is applied.
OpcDataTypeEncodingMaskAttribute	Defines the size of the mask used to encode optional field information of the custom data type represented by the Type on that this attribute is applied. The furthermore available Kind is used to determine whether the mask data is controlled by the data type or is to be determined and processed inline while reading or writing the custom data type. For more information about how an optional field is declared see OpcEncodingMaskKind.
OpcDataTypeIgnoreMemberAttribute	Declares a field or property of a custom data type, to that the member belongs on that this attribute is applied, to be ignored by the data type description.
OpcDataTypeInfo	Represents type declarations used for data types.
OpcDataTypeMemberAttribute	Declares a field or property of a custom data type, to that the member belongs on that this attribute is applied, with different Name and Type information to use when reading and writing the member. By default the Name and the Type are determined by the MemberInfo.Name and the FieldInfo.FieldType or PropertyInfo.PropertyType. In case there a different name or a different Type is to be used for a member use this attribute to declare them. Keep in mind to declare only different types which implement IConvertible supporting the Type specified; otherwise declare the desired type directly instead of using this attribute. Additionally its possible to derive from OpcData and use IOpcDataStore instances with properties to implement custom data convertion.

Name	Description
OpcDataTypeMemberLengthAttribute	Declares a field or property of a custom data type, to that the member belongs on that this attribute is applied, with a variable or fixed length of the member within the stream of its custom data type. Use this attribute to explicitly declare a field or property with a variable or fixed length. The length within the stream of its custom data type is declared either using the number of instances (= fixed length) using the constructor OpcDataTypeMemberLengthAttribute(Int64) or using the value of a preceding member (= variable length) using the constructor OpcDataTypeMemberLengthAttribute(String).
OpcDataTypeMemberOffsetAttribute	Declares a field or property of a custom data type, to that the member belongs on that this attribute is applied, with an absolute zero-based start-offset within the stream of its custom data type. Using the Offset specified means, that the members data starts at the byte specified by Offset. All preceding bytes are skipped; if not read for a prior member. When writing a member with a specific Offset all preceding bytes are written using zeros; if not written using a prior member. If this attribute is not defined, the first member is read and written using zero as its offset.
OpcDataTypeMembersAttribute	Defines additional information like different criterions to use to select the members of a data type which shall be respected and therefore encoded / decoded for the data type on which the attribute is applied.
OpcDataTypeMemberSwitchAttribute	Declares a field or property of a custom data type, to that the member belongs on that this attribute is applied, with conditional existence of the member within the stream of its custom data type. Use this attribute to explicitly declare a field or property as optional. Its existence within the stream of its custom data type is declared either using a bit within the encoding mask of the whole data type (see OpcDataTypeEncodingMaskAttribute) using the constructors OpcDataTypeMemberSwitchAttribute and OpcDataTypeMemberSwitchAttribute(Int32) or using the value of a preceding member using the constructors OpcDataTypeMemberSwitchAttribute(String), OpcDataTypeMemberSwitchAttribute(String, Int64) or OpcDataTypeMemberSwitchAttribute(String, Int64, OpcMemberSwitchOperator).
OpcDataTypeNode	Defines an abstract data type node.
OpcDataTypes	
OpcDataTypeSystem	Represents an encoding-based system which is organized in type dictionaries to define type information to use to understand data exchanged via OPC UA using a specific type of encoding.
OpcDataVariableNode	Provides a node to represent complex data structures expose their information as complex data types, as well as components in the address space. This allows access to simple values as well as access to the whole information at once in a transactional context.
OpcDefaultImpersonationContext	

Name	Description
OpcDeleteNode	Defines a single command of the IOpcDeleteNodesService used to delete one node.
OpcDeleteNodeCollection	Defines a collection for OpcDeleteNode instances.
OpcDeleteNodeHistoryAtTimeDetails	
OpcDeleteNodeHistoryDetails	
OpcDeleteNodeHistoryModifiedDetails	
OpcDeleteNodeHistoryRawDetails	
OpcDeleteReference	Defines a single command of the IOpcDeleteReferencesService used to delete a logical reference between one and another node.
OpcDeleteReferenceCollection	Defines a collection for OpcDeleteReference instances.
OpcDialogCondition	Represents a snapshot of a 'Opc.UaFx.OpcDialogConditionNode'.
OpcDialogConditionNode	Defines a condition node which represents itself as a dialog.
OpcDialogResponseMethodNode	Defines a method node to .
OpcDirectoryPathProbing	Provides mechanisms to evaluate the full qualified existence and accessibility of a String-based path which points to a directory in the file system.
OpcDiscreteAlarm	Represents a snapshot of a 'Opc.UaFx.OpcDiscreteAlarmNode'.
OpcDiscreteAlarmNode	Defines an alarm condition node which is used to classify types into alarm conditions where the input for the alarm may take on only a certain number of possible values (e.g. true/false, running/stopped/terminating).
OpcElementOperand	
OpcEncoding	Identifies how data is encoded on the wire and therefore describes in which way information is serialized or deserialized.
OpcEncodingMask	Defines the characteristics of the 'EncodingMask' of a type with optional members.
OpcEncodingOptions	Provides the setup used to control encoding results.
OpcEncodingStackFrame	Provides information about a OpcEncodingStackFrame, which represents the encoding / decoding operation performed on the encoding stack for the current encode / decode operation.
OpcEncodingStackTrace	Represents a stack trace of an encoder / decoder, which is an ordered collection of one or more stack frames.
OpcEndpointConfiguration	
OpcEndpointDescription	
OpcEndpointDescriptionCollection	Represents a collection of OpcEndpointDescription instances.
OpcEndpointIdentity	
OpcEndpointIdentityReadOnlyCollection	Provides a read-only collection of OpcEndpointIdentity instances.
OpcEngineeringUnitInfo	Provides information to classify a value using codes for units of measurements.
OpcEngineeringUnitInfoPropertyNode	Defines a property node for engineering unit information properties.

Name	Description
OpcEnumMember	Defines the metadata required to represent a member of an enumerated type definition.
OpcEnumMemberAttribute	Defines custom metadata which is used when a member of an enumeration is to be used as a “EnumString” or “EnumValue” of a custom data type node representing an enumerated type.
OpcEvent	Represents a snapshot of a 'Opc.UaFx.OpcEventNode'.
OpcEventCollection	Defines a collection for OpcEvent instances.
OpcEventContext	
OpcEventDataSet	
OpcEventDataSetItem	
OpcEventFilter	
OpcEventNode	Defines a generic base class for all general characteristics of an OPC UA Event.
OpcEventNodeSnapshot	Defines a lightweight immutable deep snapshot of a whole 'Opc.UaFx.OpcEventNode'.
OpcEventNodeView	Provides immutable access to the filtered data of a 'Opc.UaFx.OpcEventNode'.
OpcEventTypeAttribute	Defines the identifier of a custom event type which is represented by the Type on that this attribute is applied. If the Type which uses the attribute is an 'OpcEventNode' derivate the attribute additionally defined which type of OpcEvent instances are reported.
OpcEventTypes	Provides additional metadata and service routines for the different types of events supported and defined by the OPC UA Foundation Stack.
OpcException	The exception that is thrown when an OPC operation has failed.
OpcExclusiveDeviationAlarm	Represents a snapshot of a 'Opc.UaFx.OpcExclusiveDeviationAlarmNode'.
OpcExclusiveDeviationAlarmNode	Defines a limit alarm which utilizes multiple mutually exclusive limits.
OpcExclusiveLevelAlarm	Represents a snapshot of a 'Opc.UaFx.OpcExclusiveLevelAlarmNode'.
OpcExclusiveLevelAlarmNode	Defines a limit alarm which utilizes an alarm with multiple mutually exclusive limits.
OpcExclusiveLimitAlarm	Represents a snapshot of a 'Opc.UaFx.OpcExclusiveLimitAlarmNode'.
OpcExclusiveLimitAlarmNode	Defines an alarm condition node with multiple mutually exclusive limits.
OpcExclusiveLimitStateMachineNode	
OpcExclusiveRateOfChangeAlarm	Represents a snapshot of a 'Opc.UaFx.OpcExclusiveRateOfChangeAlarmNode'.
OpcExclusiveRateOfChangeAlarmNode	Defines a limit alarm which utilizes an alarm with multiple mutually exclusive limits using a rate of change.
OpcFileCloseMethodNode	Defines a method node to close a file.
OpcFileGetPositionMethodNode	Defines a method node to query a position in a file.
OpcFileInfo	Provides a default implementation of the IOpcFileInfo interface representing a FileInfo for the use in OPC UA.
OpcFileNode	Defines a file node.
OpcFileOpenMethodNode	Defines a method node to open a file.

Name	Description
OpcFileReadMethodNode	Defines a method node to read data from a file.
OpcFileSetPositionMethodNode	Defines a method node to update a position in a file.
OpcFileWriteMethodNode	Defines a method node to write data to a file.
OpcFilterOperand	
OpcFilterOperandCollection	Represents a collection of OpcFilterOperand instances.
OpcFilterOperandReadOnlyCollection	Provides a read-only collection of OpcFilterOperand instances.
OpcFiniteStateMachineNode	Defines a specialization of the OpcStateMachineNode that provides a mechanism to explicitly define the states and transitions. A server should use this mechanism if it knows what the possible states are and the state machine is not trivial.
OpcFiniteStateVariableNode	Defines a specialization of the OpcStateVariableNode which uses a OpcNodeId as the value of the VariableId.
OpcFiniteTransitionVariableNode	Defines a specialization of the OpcTransitionVariableNode which uses a OpcNodeId as the value of the VariableId.
OpcFolderNode	Defines a node to organise the address space into a hierarchy of nodes. They represent the root node of a subtree, and have no other semantics associated with them.
OpcHistoryConfigurationNode	Defines the general characteristics of a node that defines the historical configuration of any historical data node that is defined to contain history.
OpcHistoryModificationInfo	
OpcHistoryValue	
OpcIdentity	
OpcImpersonationContext	
OpcIncrementalNodeIdFactory	Provides a factory for NumericOpcNodeId instances which generates per namespace unique node identifiers.
OpcInstanceNode	Defines the abstract base class of all instance nodes.
OpcInstanceNodeSnapshot	Defines a lightweight immutable deep snapshot of a whole 'Opc.UaFx.OpcInstanceNode'.
OpcLimitAlarm	Represents a snapshot of a 'Opc.UaFx.OpcLimitAlarmNode'.
OpcLimitAlarmNode	Defines an alarm condition node with multiple limits.
OpcLiteralOperand	
OpcLocaleCollection	Represents a collection of String values identifying different cultures (also known as CultureInfo).
OpcMemberLength	Declares the total number of bytes or the number of instances of the type referenced by a member either direct or indirect by referencing a different member its value defines the length of the member.
OpcMemberSwitch	Declares the condition under which an optional member exists in dependency of the value of a switch member. The switch member can be either a single bit in the 'EncodingMask' or a different member its value is compared to a switch value (see Value).
OpcMethodCommands	Provides a set of static (Shared in Visual Basic) IOpcMethodCommand instances.

Name	Description
OpcMethodContext	Provides information and services about the system which belong to the current method call situation.
OpcMethodDelegateCommand	Provides an implementation of the IOpcMethodCommand interface which delegates method evaluations and method calls to a set of callback routines.
OpcMethodNode	Defines a method node.
OpcModel	Describes a specific model along with any dependencies these model have.
OpcModifiedHistoryValue	
OpcMonitoringFilter	Defines the base class of filter definitions used for node monitoring using subscriptions.
OpcName	Represents the name of any kind of OPC UA entity through qualifying its nominal information with a reference to the namespace within the name is used.
OpcNameCollection	Representes a collection of OpcName instances.
OpcNamePath	Defines a sequence of OpcName objects which can be used to browse through nodes using their Name values as the selection criteria to query node by node using the elements of the path.
OpcNamePropertyNode	Defines a property node for name properties.
OpcNamespace	Represents a nominal space used to categorize information like nodes, types, identifiers, etc. under a logical topic.
OpcNamespaceCollection	Represents a collection of OpcNamespace instances.
OpcNode	
OpcNodeChangesEventArgs	Provides the event data for the BeforeApplyChanges and AfterApplyChanges events.
OpcNodeCollection	Represents a collection of IOpcNode instances.
OpcNodeContext	Provides node sensitive information and services about the system which belong to the current situation (process, thread, session, node, etc.).
OpcNodeEventArgs	Provides generic event data for OpcNode related events.
OpcNodeExportOptions	Provides the setup used to collect the metadata of a node to export.
OpcNodeGlobalization	Provides mechanisms to control the localization of data and information used by a node.
OpcNodeId	Defines an identifier for a node in the address space of an OPC UA Server.
OpcNodeIdCollection	Represents a collection of OpcNodeId instances.
OpcNodeIdFactory	Provides an abstract base class which defines mechanisms to create new OpcNodeId instances.
OpcNodeIdPropertyNode	Defines a property node for node identifiers.
OpcNodeReference	Defines a bunch of OpcReference instances associated with a specific node using its IOpcNode instance or a OpcNodeId to identify the belonging node the references are used for.
OpcNodeReferenceCollection	Provides a collection of OpcNodeReference instances where the NodeId is used as the key of each entry.
OpcNodeReferenceExportOptions	Provides the setup used to collect the metadata of the references used by a node to export.
OpcNodeReferenceSet	Provides a sequence of OpcNodeReference instances.

Name	Description
OpcNodeReferenceSource	Provides a set of methods to construct OpcNodeReference instances using different source node information.
OpcNodeSet	Provides a static in-memory representation of an address space including its nodes their references to each other and the used types.
OpcNodeTypeInfo	Represents type declarations used for node types.
OpcNodeTypeSystem	Represents a system which defines type information used as templates to initialize different types of nodes. Such nodes can be object, variable or event nodes.
OpcNominalNodeIdFactory	Provides a generic factory for OpcNodeId instances which generates general unique node identifiers using Create(OpcContext) / Create(OpcContext, OpcNamespace) or node dependent unique node identifiers using Create(OpcContext, OpcNamespace, IOpcNodeInfo).
OpcNonExclusiveDeviationAlarm	Represents a snapshot of a 'Opc.UaFx.OpcNonExclusiveDeviationAlarmNode'.
OpcNonExclusiveDeviationAlarmNode	Defines a limit alarm which utilizes one or more non-exclusive states.
OpcNonExclusiveLevelAlarm	Represents a snapshot of a 'Opc.UaFx.OpcNonExclusiveLevelAlarmNode'.
OpcNonExclusiveLevelAlarmNode	Defines a limit alarm which utilizes one or more non-exclusive states.
OpcNonExclusiveLimitAlarm	Represents a snapshot of a 'Opc.UaFx.OpcNonExclusiveLimitAlarmNode'.
OpcNonExclusiveLimitAlarmNode	Defines an alarm condition node with multiple non-exclusive limits.
OpcNonExclusiveRateOfChangeAlarm	Represents a snapshot of a 'Opc.UaFx.OpcNonExclusiveRateOfChangeAlarmNode'.
OpcNonExclusiveRateOfChangeAlarmNode	Defines a limit alarm which utilizes one or more non-exclusive states using a rate of change.
OpcNotification	
OpcNotificationDataSetCollection	Represents a collection of IOpcNotificationDataSet instances.
OpcNotificationDataSetItem	
OpcNotificationDataSetReadOnlyCollection	Provides a read-only collection of IOpcNotificationDataSet objects.
OpcObjectNode	Defines a logical unit to represent more complex information as a OpcVariableNode. Objects are used to represent systems, system components, real-world objects and software objects.
OpcObjectTypeInfo	Represents type declarations used for object node types.
OpcObjectTypes	
OpcObjectTypes.OpcAggregateFunction	
OpcObjectTypes.OpcAlarmConditionType	
OpcObjectTypes.OpcExclusiveLimitAlarmType	
OpcObjectTypes.OpcExclusiveLimitStateMachineType	
OpcObjectTypes.OpcHAConfiguration	
OpcObjectTypes.OpcHistoricalDataConfigurationType	
OpcObjectTypes.OpcHistoryServerCapabilities	

Name	Description
OpcObjectTypes.OpcHistoryServerCapabilitiesType	
OpcObjectTypes.OpcModellingRule	
OpcObjectTypes.OpcNamespaceMetadataType	
OpcObjectTypes.OpcNamespacesType	
OpcObjectTypes.OpcProgramStateMachineType	
OpcObjectTypes.OpcServer	
OpcObjectTypes.OpcServer.OpcCapabilities	
OpcObjectTypes.OpcServer.OpcDiagnostics	
OpcObjectTypes.OpcServerCapabilitiesType	
OpcObjectTypes.OpcServerDiagnosticsType	
OpcObjectTypes.OpcServerType	
OpcObjectTypes.OpcServerType.OpcCapabilities	
OpcObjectTypes.OpcServerType.OpcDiagnostics	
OpcObjectTypes.OpcShelvedStateMachineType	
OpcObjectTypes.OpcXmlSchema	
OpcOffNormalAlarm	Represents a snapshot of a 'Opc.UaFx.OpcOffNormalAlarmNode'.
OpcOffNormalAlarmNode	Defines an alarm condition node intended to represent a discrete condition that is considered to be not normal.
OpcPathProbing	Provides an abstract base class which defines mechanisms to evaluate String-based path values used to access the file system. A path probing evaluates a specific path (source) against its existence, its acessibility and other circumstances and offers only successful evaluated path information.
OpcPathProbingCollection	Represents a collection of OpcPathProbing instances.
OpcPathProbingFailedEventArgs	Provides the event data for the Failed event.
OpcPropertyNode	Defines a property node.
OpcReadAttributeValueContext	Provides node sensitive information and services about the system which belong to the current situation (process, thread, session, node, etc.) used for reading the value of a node attribute.
OpcReadEventHistoryDetails	
OpcReadHistoryDetails	
OpcReadNode	Defines a single command of the IOpcReadNodesService.
OpcReadNodeCollection	Defines a collection for OpcReadNode instances.
OpcReadNodeHistoryAtTimeDetails	
OpcReadNodeHistoryDetails	
OpcReadNodeHistoryModifiedDetails	
OpcReadNodeHistoryProcessedDetails	
OpcReadNodeHistoryRawDetails	
OpcReadOnlyArgumentCollection	Provides a read-only collection of OpcArgument instances.
OpcReadOnlyModelCollection	
OpcReadOnlyNamespaceCollection	Provides a read-only collection of OpcNamespace instances.
OpcReadOnlyNodeCollection	Provides a read-only collection of IOpcNode instances accessible by their Id or just using an index.

Name	Description
OpcReadOnlyNodeIdCollection	Provides a read-only collection of OpcNodeId instances.
OpcReadOnlyReferenceTypeCollection	Provides a read-only collection of OpcReferenceType values.
OpcReadPropertyValueContext	Provides property node sensitive information and services about the system which belong to the current situation (process, thread, session, property node, etc.) used for reading the value of a property node.
OpcReadVariableValueContext	Provides variable node sensitive information and services about the system which belong to the current situation (process, thread, session, variable node, etc.) used for reading the value of a variable node.
OpcReference	Describes a unidirectional reference from a logical 'parent' to a logical 'child' or vice versa using a specific type of reference between the two opponents.
OpcReferenceCollection	Representes a collection of OpcReference instances.
OpcReferenceDescription	
OpcReferenceDescriptionCollection	
OpcReferenceTypes	Provides additional metadata and service routines for the different types of references supported and defined by the OPC UA Foundation Stack.
OpcRegisterNode	Defines a single command of the IOpcRegisterNodesService used to register one node for optimized node access.
OpcRegisterNodeCollection	Defines a collection for OpcRegisterNode instances.
OpcRelativePath	Defines a sequence of references and browse names to follow.
OpcRelativePathElement	Defines a reference and browse name to follow.
OpcResult	Provides information indicating and describing the outcome of an operation.
OpcResultCollection	Representes a collection of OpcResult instances.
OpcResultReadOnlyCollection	Provides a read-only collection of OpcResult instances.
OpcSecurity	Provides security options used by an application instance to control the authentication, the access and the possible operations between the opponent applications.
OpcSecurityPolicy	Describes the configuration used to secure the message exchange.
OpcSecurityPolicyCollection	
OpcServerStatePropertyNode	Defines a property node for server state properties.
OpcShelvedStateMachineNode	Defines a specialization of the OpcFiniteStateMachineNode which implements a sub-state machine that represents an advanced alarm filtering model.
OpcSimpleAttributeOperand	
OpcSimpleAttributeOperandCollection	Represents a collection of OpcSimpleAttributeOperand instances.
OpcSimpleAttributeOperandReadOnlyCollection	Provides a read-only collection of OpcSimpleAttributeOperand instances.
OpcSocketIdentity	

Name	Description
OpcSpecialFolderPathProbing	Provides mechanisms to evaluate the full qualified existence and accessibility of a String-based path which points to a directory in the file system. The thereby evaluated directories are identified by one of the values defined by the SpecialFolder enumeration.
OpcStateMachineNode	Defines the base class for all state machine nodes. It uses a single variable which represents the current state of the machine.
OpcStateVariableNode	Defines a variable node which stores the current state of a state machine as a human readable name.
OpcStatus	Provides information about the outcome of an operation.
OpcStatusCodeExtensions	
OpcStatusCollection	Representes a collection of OpcStatus instances.
OpcStatusConditionVariableNode	Defines a condition variable node for OpcStatus variables.
OpcTaskSchedule	Defines the central entry point used to schedule any kind of task to perform using custom queuing and scheduling mechanisms.
OpcText	Represents a human readable String value optionally associated with a Culture and localization relevant information.
OpcTextArrayPropertyNode	Defines a property node for text array properties.
OpcTextConditionVariableNode	Defines a condition variable node for OpcText variables.
OpcTextPropertyNode	Defines a property node for text properties.
OpcTextVariableNode	Defines a variable node for texts.
OpcTimedShelveMethodNode	Defines a method node to transition a condition into the timed shelved state.
OpcTransitionVariableNode	Defines a variable node which stores a transition within a state machine as a human readable name.
OpcTranslatePath	Defines a single command of the IOpcTranslatePathsService.
OpcTranslatePathCollection	Defines a collection for OpcTranslatePath instances.
OpcTranslatePathResult	Provides information indicating and describing the outcome of an OpcTranslatePath command execution.
OpcTranslatePathResultCollection	Defines a collection for OpcTranslatePathResult instances.
OpcTransport	Provides transport options used by an application instance to control the behavior of the operation, communication and data flow between the opponent applications.
OpcTransportProfiles	
OpcTripAlarm	Represents a snapshot of a 'Opc.UaFx.OpcTripAlarmNode'.
OpcTripAlarmNode	Defines an alarm condition node which is intended to represent an equipment trip condition. The alarm becomes active when the monitored piece of equipment experiences some abnormal fault such as a motor shutting down due to an overload condition.
OpcTwoStateVariableNode	Defines a specialization of the OpcStateVariableNode which uses a Boolean as the value of the VariableId.

Name	Description
OpcTypeDecodingFailedEventArgs	Provides the event data for events of the delegate type OpcTypeDecodingFailedEventHandler.
OpcTypeEncodingFailedEventArgs	Provides the event data for events of the delegate type OpcTypeEncodingFailedEventHandler.
OpcTypeNode	Defines the abstract base class of all type nodes.
OpcTypeResolveEventArgs	Provides the event data for events used to resolve missing Type information.
OpcTypes	Provides a set of static (Shared in Visual Basic) services to maintain different types used in the ecosystem of an OPC UA Server.
OpcUnregisterNode	Defines a single command of the IOpcUnregisterNodesService used to unregister one node to not longer optimize their node access.
OpcUnregisterNodeCollection	Defines a collection for OpcUnregisterNode instances.
OpcUpdateNodeHistoryDataDetails	
OpcUpdateNodeHistoryDetails	
OpcUpdateNodeHistoryDetailsCollection	Represents a collection of OpcUpdateNodeHistoryDetails instances.
OpcUpdateNodeHistoryStructureDataDetails	
OpcUserIdentity	
OpcValue	Stores a value optionally associated with status and timestamp information.
OpcValueCollection	Represents a collection of OpcValue instances.
OpcValueRange	Defines the range for a value.
OpcValueRangePropertyNode	Defines a property node for value range properties.
OpcVariableNode	Defines a node to use to represent values which may be simple or complex. Variables are always defined as properties or data variables of other nodes in the address space. They are never defined by themselves. A variable is always part of at least one other node, but may be related to any number of other nodes.
OpcVariableTypeInfo	Represents type declarations used for variable node types.
OpcVariableTypes	
OpcViewDescription	
OpcWriteAttributeValueContext	Provides node sensitive information and services about the system which belong to the current situation (process, thread, session, node, etc.) used for writing the value of a node attribute.
OpcWriteNode	Defines a single command of the IOpcWriteNodesService.
OpcWriteNodeCollection	Defines a collection for OpcWriteNode instances.
OpcWritePropertyValueContext	Provides property node sensitive information and services about the system which belong to the current situation (process, thread, session, property node, etc.) used for writing the value of a property node.
OpcWriteVariableValueContext	Provides variable node sensitive information and services about the system which belong to the current situation (process, thread, session, variable node, etc.) used for writing the value of a variable node.
UriCollection	Represents a collection of Uri objects.

Structs

Name	Description
Bit	Represents a Bit (high or low) value.
ByteString	Defines a sequence of Byte values.
OpcNodeValuePair	Defines a node value pair using an IOpcNodeValue object as the node and an Object as the value.

Interfaces

Name	Description
ILicenseInfo	Provides the license conditions of a single license.
ILicenseType	Provides information about the kind of license that activates according features in the product, depending on the type.
IOpcApplication	
IOpcApplicationInstance	Defines a concrete application of one of the possible OPC UA application types available.
IOpcDataFieldInfo	Defines the metadata of a field declared by a data type.
IOpcDataStore	Defines mechanisms to store and retrieve data using OPC UA defined techniques to store and locate information.
IOpcDataTypeInfo	Defines type declaration information used for data types.
IOpcFileInfo	Represents a single file in OPC UA and provides properties and instance methods for accessing files.
IOpcMethodCommand	Defines a command used as the implementation of a method.
IOpcNamespaceResolver	Defines methods to resolve metadata of an OPC UA namespace.
IOpcNode	
IOpcNodeAttribute	Describes an attribute associated with a specific node.
IOpcNodeDescriptor	Describes a specific node using the identifier of the node.
IOpcNodeInfo	Defines an interface which can be used to access basic properties used over any kind of node in an immutable way.
IOpcNodeReferenceAware	Defines the lowest common interface of types which offer OpcNodeReference instances using different circumstances and conditions to define or provide OpcNodeReference class instances.
IOpcNodeValue	Describes the Value attribute associated with a specific node.
IOpcNotificationDataSet	
IOpcReadOnlyNodeDataStore	Defines read-only mechanisms to retrieve node related data using OPC UA defined techniques to store and locate information using different OpcAttribute's and OpcName's.
IOpcType	Represents type declarations used for objects, variables, events, references and data.
IOpcTypeProvider	Provides type information used in the OPC UA address space.
IOpcTypeSystem	Provides type information used within a specific scope of OPC UA.

Delegates

Name	Description
OpcAddCommentCallback	Represents the method that will apply the `comment` to the state of a condition node.
OpcAddCommentMethodNode.AddCommentCallback	Applies the `comment` to the state of a condition.

Name	Description
OpcAddCommentMethodNode.AddCommentExCallback	Applies the `comment` to the state of a condition.
OpcCertificateJudgementEventHandler	Represents the method that will handle 'CertificateJudgementFailed' events.
OpcCertificateQueryInstanceEventHandler	Represents the method that will handle QueryInstance events.
OpcCertificateRequestedEventHandler	Represents the method that will handle CertificateRequested events.
OpcCertificateValidationFailedEventHandler	Represents the method that will handle 'CertificateValidationFailed' events.
OpcDialogRespondCallback	Represents the method that will respond on a dialog of a dialog condition node.
OpcDialogResponseMethodNode.DialogResponseCallback	Passes the selected response option and ends the dialog.
OpcDialogResponseMethodNode.DialogResponseExCallback	Passes the selected response option and ends the dialog.
OpcFileCloseMethodNode.FileCloseCallback	Closes the file represented. When a client closes a file the handle becomes invalid.
OpcFileCloseMethodNode.FileCloseExCallback	Closes the file represented. When a client closes a file the handle becomes invalid.
OpcFileGetPositionMethodNode.FileGetPositionCallback	GetPosition is used to provide the current position of the file handle.
OpcFileGetPositionMethodNode.FileGetPositionExCallback	GetPosition is used to provide the current position of the file handle.
OpcFileOpenMethodNode.FileOpenCallback	Opens the file represented. When a client opens a file it gets a file handle that is valid while the session is open. Clients shall use the Close Method to release the file handle when they do not need access to the file anymore. Clients can open the same file several times for read.
OpcFileOpenMethodNode.FileOpenExCallback	Opens the file represented. When a client opens a file it gets a file handle that is valid while the session is open. Clients shall use the Close Method to release the file handle when they do not need access to the file anymore. Clients can open the same file several times for read.
OpcFileReadMethodNode.FileReadCallback	Read is used to read a part of the file starting from the current file position. The file position is advanced by the number of bytes read.
OpcFileReadMethodNode.FileReadExCallback	Read is used to read a part of the file starting from the current file position. The file position is advanced by the number of bytes read.
OpcFileSetPositionMethodNode.FileSetPositionCallback	SetPosition is used to set the current position of the file handle.
OpcFileSetPositionMethodNode.FileSetPositionExCallback	SetPosition is used to set the current position of the file handle.
OpcFileWriteMethodNode.FileWriteCallback	Write is used to write a part of the file starting from the current file position. The file position is advanced by the number of bytes written.

Name	Description
OpcFileWriteMethodNode.FileWriteExCallback	Write is used to write a part of the file starting from the current file position. The file position is advanced by the number of bytes written.
OpcMethodCanExecuteCallback	Determines the value which decides whether a method implementation can be executed.
OpcMethodExecuteCallback	Performs the tasks necessary to implement the logic associated with a specific method using the `context` and `inputArguments` specified.
OpcNodeChangesEventHandler	Represents the method that will handle BeforeApplyChanges and AfterApplyChanges events.
OpcNodeEventHandler	Represents the method that will handle generic OpcNode related events.
OpcPathProbingFailedEventHandler	Represents the method that will handle Failed events.
OpcQueryEventsCallback	Represents the method that will refresh / (re)evaluate any node specific conditions to determine pending / occurred events the inquirer should know about.
OpcReadEngineeringUnitInfoPropertyValueCallback	Represents the method that will read the value of a engineering unit info property node.
OpcReadNamePropertyValueCallback	Represents the method that will read the value of a name property node.
OpcReadNodeIdPropertyValueCallback	Represents the method that will read the value of a node identifier property node.
OpcReadPropertyValueCallback	Represents the method that will read the value of a property node.
OpcReadServerStatePropertyValueCallback	Represents the method that will read the value of a server state property node.
OpcReadTextArrayPropertyValueCallback	Represents the method that will read the value of a text array property node.
OpcReadTextPropertyValueCallback	Represents the method that will read the value of a text property node.
OpcReadValueRangePropertyValueCallback	Represents the method that will read the value of a value range property node.
OpcReadVariableValueCallback	Represents the method that will read the value of a variable node.
OpcTaskSchedule.Scheduler	Defines a method which is used to schedule the invocation of the `action` using custom queuing and scheduling mechanisms.
OpcTimedShelveCallback	Represents the method that will shelve the condition node for a specific amount of time.
OpcTimedShelveMethodNode.TimedShelveCallback	Transitions the alarm condition into the timed shelved state.
OpcTimedShelveMethodNode.TimedShelveExCallback	Transitions the alarm condition into the timed shelved state.
OpcTypeDecodingFailedEventHandler	Represents the method that will handle events which occur when the decoding of a specific type has failed.

Name	Description
OpcTypeEncodingFailedEventHandler	Represents the method that will handle events which occur when the encoding of a specific type has failed.
OpcTypeResolveEventHandler	Represents the method that will handle events used to resolve missing Type information.
OpcWriteEngineeringUnitInfoPropertyValueCallback	Represents the method that will write the value of a engineering unit info property node.
OpcWriteNamePropertyValueCallback	Represents the method that will write the value of a name property node.
OpcWriteNodeIdPropertyValueCallback	Represents the method that will write the value of a node identifier property node.
OpcWritePropertyValueCallback	Represents the method that will write the value of a property node.
OpcWriteServerStatePropertyValueCallback	Represents the method that will write the value of a server state property node.
OpcWriteTextArrayPropertyValueCallback	Represents the method that will write the value of a text array property node.
OpcWriteTextPropertyValueCallback	Represents the method that will write the value of a text property node.
OpcWriteValueRangePropertyValueCallback	Represents the method that will write the value of a value range property node.
OpcWriteVariableValueCallback	Represents the method that will write the value of a variable node.

Table of Contents

Classes 1
Structs 15
Interfaces 15
Delegates 15