



User Guide Part 10

Object Model Reference

Contents

- 1 TERMS, DEFINITIONS AND ABBREVIATIONS 4**
- 1.1 Terms.....4
- 1.1.1 Asset..... 4
- 1.1.2 Data Model..... 4
- 1.1.3 Mimic..... 4
- 1.1.4 Mimic Template..... 4
- 1.1.5 Status Object Model..... 4
- 1.1.6 Workspace..... 4
- 1.2 Abbreviations.....4
- 2 OVERVIEW..... 5**
- 2.1.1 OPC UA..... 5
- 2.1.2 SResult..... 5
- 2.2 StatusServerClient Class.....5
- 2.2.1 Properties..... 5
- 2.2.2 Methods..... 6
- 2.2.3 Events..... 6
- 2.3 LoginDialog.....6
- 2.4 StatusObjectModel Class.....7
- 2.4.1 Properties..... 7
- 2.4.2 Methods..... 7
- 2.4.3 Events.....10
- 2.4.4 Query Syntax for GetNodeIdsFromQuery.....10
- 3 OBJECT MODEL BASE CLASSES.....11**
- 3.1 Object Base.....11
- 3.1.1 Properties.....11
- 3.1.2 Methods.....11
- 3.1.3 Events.....12
- 3.2 RootCollectionBase.....12
- 3.2.1 Methods.....12
- 3.3 StatusObjectType.....13
- 3.3.1 Properties.....13
- 3.3.2 Methods.....13
- 3.3.3 Events.....14
- 4 TYPES AND PROPERTIES.....15**
- 4.1 Asset Type.....15
- 4.1.1 Properties.....15
- 4.1.2 Methods.....15
- 4.1.3 Events.....16
- 4.2 Data Item.....16
- 4.2.1 Properties.....16
- 4.2.2 Methods.....17
- 4.3 Analog Item.....17
- 4.3.1 Properties.....17
- 4.4 Discrete Data Item.....17
- 4.4.1 Properties.....17
- 4.5 Byte Array Item.....18
- 4.6 Enumeration Item.....18
- 4.6.1 Properties.....18

5	ASSET	19
5.1.1	Properties.....	19
5.1.2	Methods	19
6	FILES	20
6.1	UserFile.....	20
6.1.1	Properties.....	20
7	LOGGING	21
7.1	LoggingConfiguration.....	21
7.1.1	Properties.....	21
7.1.2	Methods	21
8	SECURITY AND WORKSPACES	22
8.1	User Account.....	22
8.1.1	Properties.....	22
8.1.2	Methods	22
8.2	Role.....	22
8.2.1	Properties.....	22
8.3	Workspace	22
8.3.1	Properties.....	23
9	SUBSCRIBING TO ALARMS AND DATA CHANGES	24
9.1	Subscription.....	24
9.1.1	Properties.....	24
9.1.2	Methods	24
9.1.3	Events.....	25
10	ADDITIONAL INFORMATION AND SUPPORT	26

Figures and Tables

Figure 1 - Status Server Login6

1 Terms, Definitions and Abbreviations

This document outlines the user interface experience for the Windows Runtime Client of Status Enterprise, known as the Status Solution Explorer.

1.1 Terms

1.1.1 Asset

An instance of a Type. A representation of a specific 'thing' in the system you are monitoring.

1.1.2 Data Model

A Data Model is a logical representation of a system that you wish to monitor, defined as types, properties, alarms and assets.

1.1.3 Mimic

A graphics screen with images and various user interface controls connected to real-time data, used to monitor and control external data, usually data from equipment and processes. Sometimes called HMI (Human Machine Interface) screen or dashboard.

1.1.4 Mimic Template

A mimic that is associated with a specific type of Asset. Mimic Templates can be opened with any Asset that is of the matching type or Assets that are sub types of the matching type.

1.1.5 Status Object Model

A .NET Library that can be used to interact with your Data Model allowing the creation of workflow and other complex interactions in your system.

1.1.6 Workspace

An accessibility object for restricting access in the Data Model. Users and roles can be associated with a workspace. Properties, assets and other objects in the Data Model can also be associated with a workspace. Users can only access things that are within the workspaces they belong to.

1.2 Abbreviations

Abbreviation	Definition
DA	Data Access
HMI	Human Machine Interface
ODBC	Open Database Connectivity
OPC	OPC Foundation
PLC	Programmable Logic Controller
SCADA	Supervisory Control and Data Acquisition
UA	Unified Architecture
XAML	Extensible Application Markup Language from Microsoft

2 Overview

Status Object Model is a .NET library that allows you to programmatically access the Status Server with higher level classes. You can monitor properties on assets for value changes, create new objects on object model, monitor alarms, update property values and generate a data model.

The Status Object Model is useful for creating your own custom applications that communicate with the Status Server. Using the Object Model you could create workflow applications, manufacturing execution engines, batch applications and link other systems in your organization with data from Status Server.

2.1.1 OPC UA

Many of the types used as parameters in method calls in the object model are types that belong to the OPC UA.

2.1.2 SResult

SResult is a structure used for error notifications. Unless specified otherwise, all methods in the object model return SResult.OK on success. For unsuccessful calls, SResult returns OPC UA status codes and exception information outlining the reason a successful call could not be completed.

2.2 StatusServerClient Class

The StatusServerClient Class is responsible for connecting to the server and holding the OPC UA Client session. This is the class that is returned from the LoginDialog, but you could use it directly to login headless.

To get the StatusObjectModel class from the StatusServerClient, call the GetObjectModel() method. To end a session with the StatusServer call EndSession();

2.2.1 Properties

Property	Type	Description
AuthenticationType	AuthenticationType	The type of authentication to use when connecting to the Status Server.
Certificate	X509Certificate	An optional certificate to use during authentication with the server.
MessageSecurityMode	MessageSecurityMode	Specifies the type of message security to use between the client and the server.
Password	string	The user's password.
SecurityPolicy	SecurityPolicy	The type of security policy to use when connecting to the server.
UserName	string	The name of the user account to use when authenticating.

2.2.2 Methods

Method	Parameters	Description
Connect	- none -	Connects to the Status Server with the current session.
EndSession	- none -	Ends the session with the server and disconnects the client.
GetObjectModel	out StatusObjectModel statusObjectModel - the object model to be returned to the caller.	Supplies the StatusObjectModel for the current session.

2.2.3 Events

Event	Type	Description
PreClientDisconnect	PreClientDisconnectHandler	Fires an event when the client is subscribed to an event when it disconnects. The subscribed event will be called before disconnecting.
StatusServerClientChanged	StatusServerClientChangedHandler	Fires an event when the client has changed.

2.3 LoginDialog

The ServerLogon class displays the user interface for logging into the Status Server. It creates a StatusServerClient and calls the appropriate methods to connect to the Status Server.

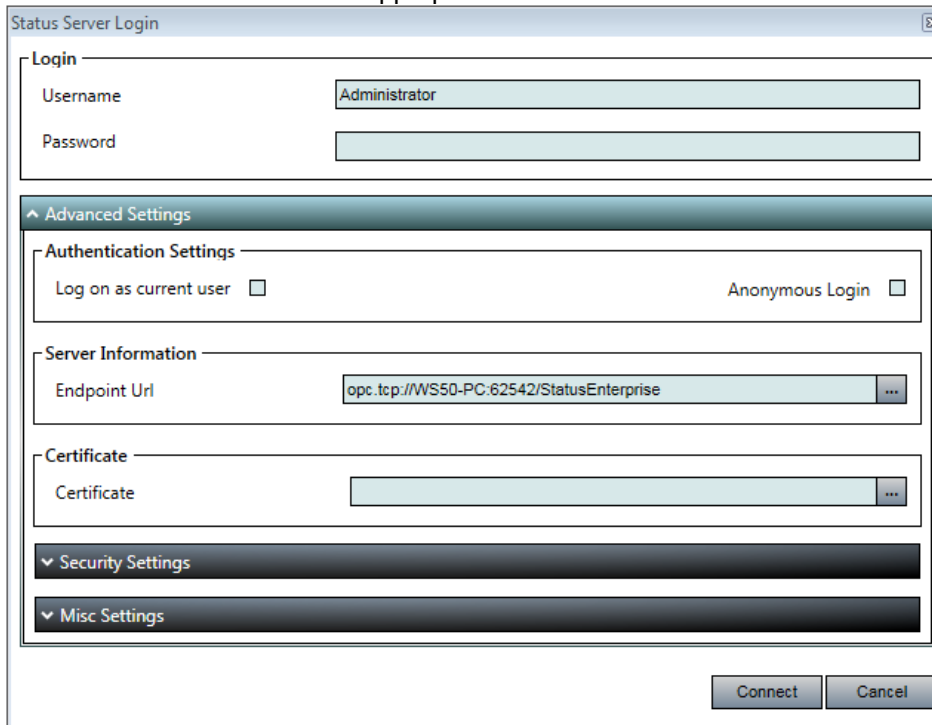


Figure 1 - Status Server Login

```

ServerLogon serverLogon = new ServerLogon();
StatusServerClient statusServerClient = serverLogon.Logon();

StatusObjectModel objectModel;

statusServerClient.GetObjectModel(out objectModel, false, true);
foreach (var view in objectModel.Views.Components)
{
    System.Diagnostics.Debug.WriteLine(view.DisplayName.Text);
}

```

2.4 StatusObjectModel Class

StatusObjectModel is the root of the object model for Status Enterprise. It is used for accessing the root collections of the object model and provides some system wide method calls.

2.4.1 Properties

Property	Type	Description
AlarmNotifications	AlarmNotifications	A collection of AlarmNotification in the object model.
AssetTypes	AssetTypes	A collection of the root types on the object model.
Contacts	Contacts	A collection of contacts in the object model.
DiscreteltemTexts	DiscreteltemTexts	A collection of DiscreteltemText in the object model.
EngineeringUnits	EngineeringUnits	A collection of EngineeringUnit in the object model.
Enumerations	Enumerations	A collection of Enumeration in the object model.
LoggingConfigurations	LoggingConfigurations	The logging configurations available on the object model.
MessageTemplates	MessageTemplates	A collection of message templates in the object model.
Roles	Roles	The Roles configured for the object model.
UserAccounts	UserAccounts	The user accounts configured for the model.
Views	Views	A collection of View in the object model.
Workspaces	Workspaces	The Workspaces configured for the model.

2.4.2 Methods

Method	Parameters	Description
BulkWrite	WriteValuesCollection writeValues - descriptions of the properties and the values being written. out DiagnosticsInfoCollection diagnostics - diagnostic information on each item being written. out StatusCodeCollection statusCodes - status codes for each item being written.	Writes a block of property value changes to the server.
CreateSubscription	out Subscription subscription - the newly created subscription.	Creates a new subscription that can be used for monitoring property changes.
DeleteAllSubscriptions	- none -	Deletes all subscriptions and unsubscribes all data items.

DeleteSubscription	Subscription subscription - the subscription to delete.	Deletes the subscription and unsubscribes all data items.
DeliveryReceived	NodeId dataItemId – the NodeId for a data item. DataValue value – the resulting value for DeliveryReceived.	Stores the NodeId and DataValue for Guaranteed Delivery to confirm that the sent and received data is either good or bad.
GetAttributeValues	List<NodeId> nodeIds – List of node ids to get values uint attribute - The attribute id of the attribute record. Use Opc.Ua.Attributes for values out DataValueCollection dataValueCollection – the resulting values for each node	Directly gets the Value for each node from the server without creating ObjectBase given a specific AttributeId(DisplayName, Value, etc).
GetAttributeValues	List<NodeId> nodeIds – List of node ids to get values List<uint> attribute – List of attribute ids that corresponds to the node list. Use Opc.Ua.Attributes for values out DataValueCollection dataValueCollection – the resulting values for each node	Directly gets the Value for each node from the server without creating ObjectBase given a specific AttributeId(DisplayName, Value, etc).
GetAttributeValues	List<NodeIdCollection> nodeIdCollections – List of node id collections to get values. List<uint> attributes – List of attribute ids that corresponds to the node list. out List<DataValueCollection> dataValueCollections – the resulting list of values for each node collection.	Directly gets the Value for each node from the server without creating ObjectBase given a specific AttributeId(DisplayName, Value, etc).
GetReferences	List<NodeId> nodeIds – The list of node ids. NodeId referenceTypeId – The NodeId reference type to browse out BrowseResultCollection – The list of BrowseResult containing node reference.	Gets the references for each given node based on the given referencetypeId
GetNodeIdFromBrowsePath	String browsePath - the path to the node. out NodeId nodeId - the node if found.	Gets the node id from the specified browse path using the root as the starting point.
GetNodeIdsFromBrowsePaths	NodeId startNode – The node id as the starting node List<string> relativePaths – the relative browse paths out List<NodeId> nodeIds – the list of node found out SResult<SResult> results – list of SResults for each path	Gets the node ids from relative browse paths using the provided starting node.
GetNodeIdsFromBrowsePaths	List<string> browsePaths - the browse paths out List<NodeId> nodeIds - list of node found out List<SResult> results – list of success or failure of each nodeIds	Gets the node ids from browse paths using the root folder as the starting node.
GetNodeIdsFromBrowsePaths	List<NodeId> startingNodes – List of node ids as the starting nodes. List<string> relativePaths – List of relative browse paths. out List<NodeIdCollection> nodeIdCollections – List of node id collections. Out List<SResultCollection> - list of SResults for each path collection.	Gets the node ids from relative browse paths using the provided starting nodes.
GetNodeIdsFromQuery	NodeId startingNodeId – The starting node string query – the query to filter items of the node out List<NodeId> - The resulting list of nodes found	Gets the list of NodeIds from a given NodeId filtered by a query

GetNodeIdsFromQuery	string browsePath – The path to the node string query – the query to filter items of the node out List<NodeId> - The resulting list of nodes found	Gets the list of NodeIds from a given path filtered by a query
GetSubscriptions	Out List<Subscription> subscriptions - the subscriptions found.	Gets the current subscriptions on the object model.
GetValuesFromQueryMethod	NodeId startingNode – the node id as the starting node. NodeId queryMethod – the node id to determine the queryMethod. String query – the query to filter the values from the query method. out VariantCollection methodResults – results from the query.	Gets the values from the query method used filtered by a query.
HistoryRead	List<NodeId> nodeIds - the nodes to read the history for. readDetails HistoricalReadDetails - the details of the history to read. out List<DataValueCollection> historicalValues	Gets the history of the specified items.
MoveType	AssetType type – The type to move. AssetType newParentType – The new parent type	Moves the specified type from being a subclass of its supertype to being a subclass of newParentType
ObjectFromNodeId	NodeId nodeId - the id of the object model object to find. out SResult sresult - the success of failure of the operation.	Returns the object model object for the specified node.
ObjectFromNodeId	NodeId nodeId - the id of the object model object to find. out ObjectBase objectBase - the object model object.	Gets the object model object for the specified node.
ObjectsFromNodeIds	List<NodeId> nodeIds - the node ids of the object model objects to get. out List<ObjectBase> objects - the objects found.	Gets the object model objects for the specified node ids.
Refresh	-none-	Refreshes the collections in the model from the server
UpdateMajorProgress	-none-	Updates the upper loading progress bar in the Mimic Designer splash screen.
UpdateMinorProgress	-none-	Updates the current process for the lower loading bar in the Mimic Designer splash screen.
ValidateSession	-none-	Validates the session to see if we have a valid connection
ValidateSessionWithException	-none-	Validates the session to see if we have a valid connection, throws an exception if session is not valid
ValidateSessionWithMessage	-none-	Validates the session to see if we have a valid connection, shows SResult message if session is not valid
WriteValue	NodeId nodeId – the NodeId of the object uint attribute – the attribute for writing unit sCode – the quality of data to be written object value – the value to be written	Writes a value to the server using the node ids

DateTime timestamp – the source timestamp

2.4.3 Events

Event	Parameters	Description
PropertyChanged	PropertyChangedEventHandler	Fired whenever a property changes on the item.
NotifyMajorProgressEvent	ProgressNotificationHandler	Fires whenever there is an update to the upper loading bar in the Mimic Designer splash screen.
NotifyMinorProgressEvent	ProgressNotificationHandler	Fires whenever there is an update to the upper loading bar in the Mimic Designer splash screen.

2.4.4 Query Syntax for GetNodeIdsFromQuery

Syntax 1

?StartRow={starting row number}**?RowsToReturn**={number of rows to return}

Example:

?StartRow=1 ?RowsToReturn=50

Syntax 2

?StartRow={starting row number}**?RowsToReturn**={number of rows to return}**?Filter**={property name}{operator}{value};**[Optional {Join};{property name}{operator}{value};]**

Example:

?StartRow=1?RowsToReturn=50?Filter=Priority|=|High;And;Project|=|Status Enterprise;

3 Object Model Base Classes

3.1 Object Base

Object Base is the base class of all object model items defined in the object model.

3.1.1 Properties

Property	Type	Description
ArrayDimensions	IList<uint>	Gets or sets the list of array dimensions.
BrowseName	QualifiedName	Gets or sets the browse name of the object model item.
DataType	NodeId	Gets the NodeId of the type for the value of this object model item, typically used with DataItem derived classes.
DataTypeDisplayName	string	Gets the display name of the data type.
Description	LocalizedText	The description of the object model item.
DisplayName	LocalizedText	Gets or sets the display name of the object model item.
EventNotifier	bool	Gets or sets a flag indicating if this node fires events
ExFromAlarmTreeNotifier	bool	Gets or sets a flag to exclude the client from being notified by the server if the asset has alarms on.
FullBrowsePath	string	Gets the full browse path of this object relative to the root of the object model.
IsHistorizing	boolean	Gets or sets a flag indicating if this item is to be logged. Only used on DataItem derived classes.
ModelingRule	ModelingRules	Gets or sets the modeling rule for the item, can be Optional or Mandatory.
NodeId	NodeId	Gets the unique NodeId of the data item.
ObjectModel	StatusObjectModel	Gets the StatusObjectModel instance
Owner	ObjectBase	Gets the object model item that is the parent or owner of this item, null for root items.
RawValue	DataValue	Gets the raw value of the data item.
TypeDefinition	NodeId	Gets the NodeId of the object model item that defines the type of this item.
TypeDefinitionDisplayName	string	Gets the display name of the type definition.
Value	object	Gets or sets the value of this item, usually used with items subclassed from DataItem.
ValueRank	int	Gets or sets the value rank, used with collections. Currently unused.

3.1.2 Methods

Method	Parameters	Description
CompareTo	object obj – the object to compare against.	Compares the browse name of an asset to another assets browse name.
Delete	-none-	Method used to delete an object from the object model.
Dispose	-none-	Method used to free an objects resources.

Equals	Object obj – object to compare equality against.	Compares equality with another object. Returns True if node ids or objects are equal.
GetHashCode	-none-	Gets the hash code of an object.
GetNodeidFromRelativePath	string browsePath - the relative browse path to the desired node. out Nodeid nodeid - the matching node found.	Gets the child node of this item provided relative browse path to the item.
GetNodeidsFromRelativePaths	List<string> browsePaths - the relative browse paths to the desired nodes. out List<Nodeid> nodeids - the matching nodes found.	Gets the child nodes of this item provided relative browse paths to the items.
IsInputParameter	MethodParameter methodParameter – the method parameter for method call optional bool bRemove – if true the parameter will be removed	Marks this object as being an input parameter for a method call
ToString	-none-	Formats a string to display an objects display name and type definition.

3.1.3 Events

Event	Parameters	Description
PropertyChanged	PropertyChangedEventHandler	Fired whenever a property changes on the item.
ObjectDeleted	ObjectBaseEventHandler	Fired whenever an object is deleted from the object model.

3.2 RootCollectionBase

RootCollectionBase is a based class used for the major collections of object model items. These include Assets, AssetTypes, UserAccounts, Workspaces, Roles, RemoteServers and LoggingConfigurations. The bulk of the properties and methods on this class relate to finding items and adding and removing items from the collection.

3.2.1 Methods

Method	Parameters	Description
AddComponent	StatusObjectType asset – an existing asset to add	Adds an existing asset as a component of this asset
FindItem	Nodeid nodeid - the NoeidId of the object model item to find in this collection. out SResult result - error information.	Returns the object model item matching the requested node id if it exists.
FindItem	String browseName - the browseName of the item to find. out SResult - error information.	Returns the object model item matching the requested browse name if it exists.
FindItemByDisplayName	string displayName – the display name of the item to finde out SResult – success or failure of the operation	Gets the requested items that match the requested display name.
FindItems	List<Nodeid> nodeids - the Nodeids of the items to find. out SResult result - error information.	Returns a ReadOnlyObservable Collection of the object model items

		matching the specified NodeIds.
FindItems	List<string> browseNames - the browseNames of the items to find. out SResult sresult - error information.	Returns a ReadOnlyObservableCollection of the object model items matching the specified browse names.

3.3 StatusObjectType

StatusObjectType is a subtype of ObjectBase. Many of the higher level objects like Asset, AssetType, FileFolder, UserFile, LoggingConfiguration, RemoteServerConfiguration, Role and Workspace are a subtype of StatusObjectType.

3.3.1 Properties

Property	Type	Description
Components	ReadOnlyObservableCollection<ObjectBase>	Gets the collection of components owned by this item.
IsOverride	boolean	Gets a flag that indicates if this object is overridden. Applies to data items and alarms, see Part 2 - Data Model Designer for more information.
IsRootItem	boolean	Gets a flag that indicates if this is a root item in the data model.
MethodCalls	ReadOnlyObservableCollection<MethodCall>	Gets the collection of MethodCall that are owned by this item
Properties	ReadOnlyObservableCollection<DataItem>	Gets the collection of DataItems that are owned by this item.
ReadOnlyWorkspaces	ObjectModelCollection<Workspace>	Gets the read only workspaces of this item.
Workspaces	ObjectModelCollection<Workspace>	Gets the collection of workspaces that are owned by this item.

3.3.2 Methods

Method	Parameters	Description
AddComponent	ObjectBase asset - the object to add.	Adds a component to this item.
CreateAlias	string browseName – The browse name for the asset Nodeid targetType – the type id of the asset Nodeid target – the owner of the asset, or null for a root asset out Alias alias – the newly created asset	Creates a new asset as an asset on this asset
CreateComponent	string browseName – the browse name of the component string displayName – the display name of the component StatusObjectType assetType – the asset type out SResult – success or failure of the operation	Creates a new asset based on the asset type and adds it as a component of this item.
CreateComponent	string browseName – the browse name of the	Creates a new asset of the specified type and adds it as a

	<p>component</p> <p>string displayName – the display name of the component</p> <p>Nodeld assetTypeld – the type id of the asset</p> <p>out SResult sresult – success or failure of the operation</p>	<p>component of this item.</p>
CreateMethodCall	<p>string displayName – the name of the method</p> <p>Assembly assembly – the actual assembly of the method</p> <p>Type classType – the Type class</p> <p>MethodInfo method – the method information</p> <p>out SResult – success or failure of the operation</p>	<p>Creates a new Method Call</p>
CreateMethodCall	<p>MethodCall typeMethodCall – the method call type</p> <p>out SResult sresult – success or failure of the operation</p>	<p>Creates a new Method Call from an existing Method Call information</p>
DeleteMethodCall	<p>MethodCall methodcall – the method call to delete</p>	<p>Deletes a method call from this asset type</p>
FindComponent	<p>string browseName - the browse name of the component to find.</p>	<p>Finds the component with the specified browse name.</p>
FindProperty	<p>string browseName - the browse name of the property to find.</p>	<p>Finds the property with the specified browse name.</p>
IsSubtypeOf	<p>StatusObjectType parentType - the type of object to compare.</p> <p>out SResult sresult – success or failure of the operation</p>	<p>Returns true if the item is a matching type or subtype of the type provided.</p>
TranslateBrowsepathToNodeld	<p>string relativePath – the sub path of this type</p> <p>out Nodeld selectedObjectBase</p>	<p>Translate a Type browse path to the correct node</p>
TranslateBrowsePathToObjectBase	<p>string relativePath – the sub path of this type</p> <p>out ObjectBase selectedObjectBase -</p>	<p>Translate a Types browse path to the correct object</p>

3.3.3 Events

Event	Parameters	Description
ComponentCollectionChanged	NotifyCollectionChangedEventHandler	Fired whenever a change in components

4 Types and Properties

4.1 Asset Type

The AssetType class defines the types of objects that live in the data model. Inherited from StatusObjectType.

The AssetType class can be used for programmatically building the data model type hierarchy.

4.1.1 Properties

Property	Type	Description
IsDeleted	bool	Gets a flag indicating if the item has been deleted.
IsSystemType	bool	Gets a flag indicating if this item is a system type.
SubTypes	AssetTypes	Gets a list of types that are a subtype of this item.
SuperType	AssetType	Gets the type that this item is subclassed from if this item is not a root type.

4.1.2 Methods

Method	Parameters	Description
CanRename	string proposedName – a proposed new name	Indicates if this object can be renamed to the proposed name
CreateComponent	string browseName – the name of the asset StatusObjectType assetType – the asset type out Asset asset – the newly created asset	Creates an asset based on the supplied type
CreateComponentReference	AssetType assetType - the type the reference points to. String browseName - the browse name of the reference. out ObjectReference - the newly created reference.	Creates a reference to an object.
CreateProperty	string browseName - the browse and display name of the property. NodeId dataTypeId - the node id of the property type to create. out DataItem dataItem - the newly created data item. Object defaultVal - the default value for new instances of the data item, default is null. string description - the item description.	Creates a new property of the specified type. Types include DataItems, AnalogItems and Discrete Items. The data type ids for these items are defined in the OPC UA libraries.
DeleteComponent	ObjectBase asset - the object to delete.	Deletes the asset from the server and removes it from the asset type description.
DeleteComponentCollection	StatusObjectList statusObjectList - the object to delete.	Deletes the list from the server and removes it from the asset type description.
DeleteDataItem	DataItem dataItem - the data item to delete.	Deletes the data item from the server and removes it from the asset type description.

DeleteProperty	Dataltem asset – the component to delete	Deletes a component from this asset type
FindProperty	string componentName – the name of the component to find	Finds the component with the specified browse name
IsSubTypeOf	StatusObjectType parentType – the type of object to compare. out SResult sresult – success or failure of the operation	Returns true if this type is a subtype of the specified type
RefreshComponents	- none -	Refreshes components for this type, its subtypes and super types on the client.
RefreshProperties	- none -	Refreshes the properties for this type, its subtypes and super types on the client.
Rename	string proposedName – a proposed new name	Renames this asset

4.1.3 Events

Event	Parameters	Description
Deleted	EventHandler	Fired whenever an asset is deleted

4.2 Data Item

A Dataltem is both a property definition of a type and an instance of a property on an asset. Dataltem is a subtype of ObjectBase. Other more specialized data items like AnalogItem and DiscreteItem are subtypes of Dataltem. The Dataltem class is used with non-numeric property values like string and Date Time.

A data item can maintain a BindingConfiguration which defines what property the data item is bound to on which remote server. Updates from the remote server are updated on the data item by the Status Server.

4.2.1 Properties

Property	Type	Description
AccessLevel	byte	Gets or sets the access level.
Alarms	ReadOnlyObservableCollection<Alarm>	A collection of alarm on this item
BrowseName	QualifiedName	The BrowseName of the object
IsAnalog	bool	Indicates that this is an AnalogItem type
IsDiscrete	bool	Indicates that this is an DiscreteItem type
IsEnumeration	bool	Indicates that this is an EnumerationItem type
IsGuaranteedDelivery	bool	Gets or sets the flag indicating that the asset is marked for Guaranteed Delivery.
IsOverride	boolean	Gets a flag that indicates if this property is a property override of a property on the super type.
IsVolatile	bool	Gets or sets the flag indicating the value of the data item is stored in memory and never saved to the database.
LogginConfiguration	LoggingConfiguration	Gets or sets the logging configuration for this item
ReadOnly	bool	Read only set or gets the current write bit of the AccessLevel
ReadOnlyWorkspaces	ObjectModelCollection<Workspace>	Gets the collection of workspace

Value	object	Gets or sets the value of the data item.
Workspaces	ObjectModelCollection<Workspaces>	Gets the collection of workspaces this data item is a member of. Clients accessing this property must also be a member of the specified workspaces or a member of a role that is a member of one of the specified workspaces.

4.2.2 Methods

Method	Parameters	Description
CanRename	String proposedName	The method used to rename the DisplayName property of an object.
CreateAlarm	string browseName – the browse name for the alarm string description – the description of the alarm AlarmType alarmType – the type id of the alarm out Alarm alarmCondition – the newly created alarm	Creates the specified alarm type and attaches it to this asset type
CreateExistingAlarm	string browseName – the browse name for the alarm out SResult sresult – success or failure of the operation	Creates a new data item for the specified property. If the property already exists, it will be returned. The property must already exist on the type definition
DeleteAlarm	Alarm alarm – the alarm to delete	Deletes an alarm from this asset type
GetParent	out ObjectBase parent – the parent of this item	Gets the parent of this item
HistoryRead	HistoricalReadDetails readDetails - a description of what history should be read from the server. out DataValueCollection historicalValues - the historical data retrieved from the server.	Reads a block of historical data from the server based on the constraints defined in the read details.

4.3 Analog Item

Analog Item is a subtype of Data Item. It is used with numeric property values like integer and double.

4.3.1 Properties

Property	Type	Description
EngineeringUnit	EngineeringUnit	Gets or sets the engineering unit associated with the data item (meters, yards).
High	double	Gets or sets the upper limit for the item
Low	Double	Gets or sets the lower limit for the item

4.4 Discrete Data Item

Discrete Item is a subtype of Data Item. It is used with Boolean property values. It has no additional methods.

4.4.1 Properties

Property	Type	Description
----------	------	-------------

DiscreteltemText	DiscreteltemText	Gets or sets the DiscreteltemText profile to use
ValueText	LocalizedText	Gets or sets the value text for the text from DiscreteltemText TrueText or FalseText.

4.5 Byte Array Item

Byte Array Item is a subtype of Data Item. It is used with user file types loaded in the system. It has no additional properties or methods.

Property	Type	Description
Value	Object	Gets or sets the DataValue of an object.

4.6 Enumeration Item

Enumeration Item is a subtype of Data Item. It is used with Enumeration type properties. It stored an integer value as an index with the string value of the enumeration stored in the ValueText property.

4.6.1 Properties

Property	Type	Description
Enumeration	Enumeration	An enumeration profile for an Enumeration Item
ValueText	LocalizedText	Gets or sets the value text for the enumeration item, the actual text value of the enumeration.

5 Asset

An Asset is an instance of an Asset Type. It represents a specific real world object in the object model. Asset is a subtype of StatusObjectType.

5.1.1 Properties

Property	Type	Description
AssetType	AssetType	Gets the type information (the properties, methods, workspaces, alarms, and default values) for the asset.

5.1.2 Methods

Method	Parameters	Description
CreateProperty	string browseName – the name of the property	Creates a new data item for this asset.
DeleteProperty	Dataltem dataltem – the data item to delete	Deletes the data item or property of this asset.
RemoveComponent	ObjectBase asset – the component to delete	Deletes a component from this asset type

6 Files

Classes for defining files and folders stored in the object model, and their association with other object model objects.

6.1 UserFile

UserFile is a subtype of SystemType. It is a file stored in the object model that can be retrieved. Images saved as a UserFile can be bound to the BindableImage control on a mimic.

6.1.1 Properties

Property	Type	Description
CanCache	Bool	Gets or sets a value that indicates if the client can cache the file
Content	byte[]	Gets or sets the binary content of the file as a byte[].
DisplayName	String	Gets or sets the display name of the object
FileType	string	Gets or sets the file type (png, bmp, pdf).
MetaData	string	Gets or sets additional metadata to associate with the file, user-defined.
MimeType	string	Gets or sets the mime type for the file.
OwnerType	string	Gets or sets the type of the owner for the document
OwnerTypeId	NodeId	The node id for the owner of the document
Size	ulong	Gets the file size in bytes.

7 Logging

Classes used for configuring the logging of asset property values to the data log database.

7.1 LoggingConfiguration

LoggingConfiguration is a subtype of SystemType. It defines parameters for how a property should be logged.

7.1.1 Properties

Property	Type	Description
DeviationFormat	ExceptionDeviationFormat	The logging type
ExceptionDeviation	double	The ExceptionDeviation
MaxTimeInterval	double	The max logging interval in ms
MinTimeInterval	double	The minimum logging interval in ms

7.1.2 Methods

Method	Parameters	Description
CanRename	string proposedName	Indicates if this object can be renamed to the proposed name.

8 Security and Workspaces

Classes to control access to the server and restrict access based on permissions.

8.1 User Account

A user account associated with the system, a SystemType. The username for the account is the browseName since browse names must be unique.

8.1.1 Properties

Property	Type	Description
AuthenticationType	UserAuthenticationType	Gets or sets the authentication type (Built-in or Windows).
ConnectionInfo	string	Stores the connection information for validation against an LDAP server.
Email	string	Gets or sets the email address for the user.
FirstName	string	Gets or sets the first name for the user.
IsEnabled	boolean	Gets or sets a flag that enables or disables the users account.
LastName	string	Gets or sets the last name of the user.
Password	string	Sets the user's password.
Phone	string	Gets or sets the phone number for the user.
Roles	ObjectModelCollection<Role>	Gets the roles the user is associated with.
UserName	string	Gets or sets the UserName property of a user.

8.1.2 Methods

Method	Parameters	Description
CanRename	string proposedName	Indicates if this object can be renamed to the proposed name.

8.2 Role

A SystemType that groups UserAccounts together to form a collection of related users.

8.2.1 Properties

Property	Type	Description
UserAccounts	ObjectModelCollection<UserAccount>	Gets the user accounts associated with this role.

8.3 Workspace

A SystemType that groups UserAccounts, Roles, Types and Properties together to restrict access and direct specific content to particular users.

By default, assets and properties have no workspaces assigned to them and are accessible by all users. Once an asset or property is associated with a workspace, only users that are members of that workspace will be able to access the item.

8.3.1 Properties

Property	Type	Description
IsReadOnly	boolean	Gets or sets a value that indicates if the items associated with this workspace are read-only.
Roles	ObjectModelCollection<Role>	Gets the roles that are members of this workspace.
UserAccounts	ObjectModelCollection<UserAccount>	Gets the user accounts that are members of this workspace.

9 Subscribing to Alarms and Data Changes

9.1 Subscription

Subscriptions are client side objects that are used to subscribe to property changes on data items in the model, or to subscribe to alarming. To subscribe to a data item value change or an alarm notification, create a subscription using `StatusObjectModel.CreateSubscription` and add the node ids of the objects you wish to monitor to the subscription. Subscription is not a subclass of any object model class.

9.1.1 Properties

Property	Type	Description
Id	uint	The id of the underlying OPC UA subscription.
ItemCount	uint	The number of monitored items we are subscribed to
PublishingEnabled	bool	Gets or sets the flag indicating enabling the subscription to enable publishing.

9.1.2 Methods

Method	Parameters	Description
AlarmConditionRefresh	- none -	Refresh Alarm Conditions. Make request to resend current alarms
Dispose	- none -	
Resume	- none -	Resumes the current suspended subscription
Subscribe	Nodeld nodeld - the id of the node to subscribe to. out uint clientHandle – the handle when a subscribe is created int updateRate – the rate in which we want the subscription to get updates from the server MonitoringMode monitoringMode – the monitoring mode for the subscription uint attributeld - the attribute id on the node that you want to subscriber to, the default 13 which is the Value attribute.	Subscribe to a data item
Subscribe	List<Nodeld> dataItemIds - the ids of the nodes to subscribe to. List<int> updateRates – the update rates for each node ids List<MonitoringMode> monitoringModes – the monitoring modes for each ids out List<uint> clientHandles – the individual handles for each subscribe uint attributeld - the attribute id on the node that you want to subscriber to, the default 13 which is the Value attribute.	Subscribe to data items
Subscribe	ObjectBase dataItem - the object model object to subscribe to. out clientHandle – the handle when a subscribe is	Subscribe to a data item

	<p>created</p> <p>int updateRate – the rate in which we want the subscription to get updates from the server</p> <p>MonitoringMode monitoringMode - the monitoring mode for the subscription</p> <p>uint attributeld - the attribute id on the node that you want to subscriber to, the default 13 which is the Value attribute.</p>	
Subscribe	<p>List<ObjectBase> dataItems - the object model items to subscribe to.</p> <p>out List<uint> clientHandles – the individual handles for each subscribe</p> <p>int updateRate – the update rates for each node ids</p> <p>MonitoringMode monitoringMode – the monitoring mode, default is MonitoringMode.Reporting</p> <p>uint attributeld - the attribute id on the node that you want to subscriber to, the default 13 which is the Value attribute.</p>	Subscribe to data items
SubscribeAlarm	<p>Noeld nodeld - the node id of the item to subscribe to.</p> <p>Monitoring Filter - an optional filter to restrict the alarms returned.</p>	Subscribe to a data item for alarming notifications
SubscribeAliasChanged	<p>Nodeld nodeld – the node id of the data item.</p> <p>out uint clientHandle – parameter modifier reference to the client handle.</p> <p>MonitoringFilter filter – event filter for the MonitoredItem.</p>	Subscribes to a data item for alarming notifications.
Suspend	- none -	Suspends the current subscription
UnSubscribe	uint clientHandle – the handle of the subscribed data item	Unsubscribe to a data item
UnSubscribe	List<uint> - a list of data item handles to unsubscribe	Unsubscribes to data items

9.1.3 Events

Event	Parameters	Description
AlarmNotification	AlarmEventNotification	Fired whenever there is a change in the alarm state.
AliasChangedNotification	AliasChangedNotification	Fired whenever the data item is subscribed to an alias.
ValuesChanged	ValueChangedEvent	Fired whenever the value of the subscription changes.

10 Additional Information and Support

For more information in using Status Enterprise, please visit us at www.scadauniversity.com. This will give you access to instructional videos, documentation and our online forum. You can also email B-Scada support at support@b-scada.com.



B-Scada, Inc.
9030 W. Fort Island Trail
Building 9
Crystal River, FL 34429 USA
+1 352-564-9610
www.scada.com