

Description: This application note describes the availability and functionality of the Extended Point Syntax provided by the applications in the GENESIS32 and BizViz suites.

OS Requirement: Windows 2000 or later.

General Requirement: GENESIS32 or BizViz version 8 or later.

Introduction

The GENESIS32 & BizViz suites use OPC Data Access Servers to read and write information. In most cases only the values of these items are necessary, but sometimes more information is needed. This additional information can be obtained from the OPC Servers using a special syntax developed for ICONICS applications, known as Extended Point Syntax.

The Extended Point Syntax Editor

All OPC items using the Extended Point Syntax (or EPS) must be prefixed with “**tag:**”. While it is possible to manually enter the Extended Point Syntax for an OPC item, the Extended Point Syntax Editor is available from the Unified Data Browser and can be used to create the necessary syntax for you. In the Unified Data Browser, select an OPC item (do not double-click), then click on the “EPS Editor” button to open the editor.

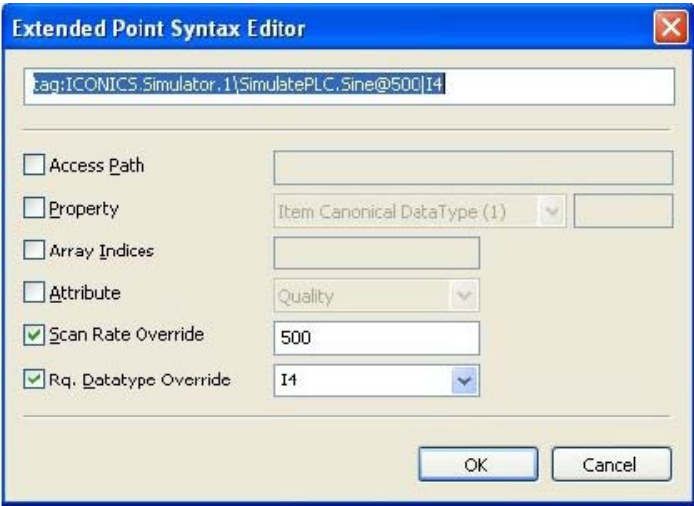


Fig 1: Extended Point Syntax Editor

NOTE: The following parameters can be used individually or combined.

Access Path

The access path allows the OPC Server to specify different methods to access the specified OPC item. The available parameters are defined by the OPC vendor.

The Extended Point Syntax for the access path is:

tag:OPCServer\OPCTag: :AccessPath

Property

In addition to the standard “value” property provided by all OPC Servers, some OPC Servers also expose a set of additional properties. When an OPC Server supports this feature it can expose a number of common properties as defined by the OPC Foundation, plus any number of custom properties. The pull-down list in the dialog provides access to all the common properties and allows you to specify your own property IDs.

The properties provided in the list are:

- **Item Canonical DataType (1)** - The item’s internal data type
- **Item Access Rights (5)** - The access rights to the OPC item as defined by the OPC Server. Typical values are: 1 – read and 3 – read/write
- **Server Scan Rate (6)** - The item’s scan rate
- **EU Units (100)** - The item’s engineering units
- **Item Description (101)** - The item’s description
- **High EU (102)** - The item’s high value in engineering units
- **Low EU (103)** - The item’s low value in engineering units
- **High Instrument Range (104)** - The item’s high value in instrument units
- **Low Instrument Range (105)** - The item’s low value in instrument units
- **Contact Close Label (106)** - The label for the item’s contact close state
- **Contact Open Label (107)** - The label for the item’s contact open state
- **Item Timezone (108)** - The item’s time zone
- **Other** - Any other custom property for which the ID can be entered manually.

The Extended Point Syntax for these properties is:

tag:OPCServer\OPCTag- ->PropertyID

Scan Rate Override

OPC items requested by an application are added to an OPC Group and each OPC Group has its own scan rate. The default scan rate used for a group is determined by the application. For example, in GraphWorX32 this is set by the display update rate and in the TrendWorX32 Logger by the data collection rate. It is possible to use this syntax item to override the scan rate at the OPC item level. This will result in one or more additional OPC Groups, each with their own appropriate scan rate.



The Extended Point Syntax for the scan rate override is:

tag:OPCServer\OPCTag@RateInMilliseconds

Requested Data Type Override

In the majority of cases, an OPC item is requested using the native data type. There is no special action required for this in any of the ICONICS applications, but in some cases it is useful to force a data type for the OPC item.

The Extended Point Syntax supports the following range data types, using the names defined in the Variant Type enumerator:

- I1 (8 bit signed integer)
- I2 (16 bit signed integer)
- I4 (32 bit signed integer)
- I8 (64 bit signed integer)
- UI1 (8 bit unsigned integer)
- UI2 (16 bit unsigned integer)
- UI4 (32 bit unsigned integer)
- UI8 (64 bit unsigned integer)
- R4 (32 bit floating point)
- R8 (64 bit floating point)
- BOOL (boolean)
- BSTR (string)
- DATE (date/time format)
- DECIMAL (decimal)
- HRESULT
- ERROR

The Extended Point Syntax for the data type override is:

tag:OPCServer\OPCTag|DataType.

Array Indices

When OPC Servers provide array data it may be necessary to extract a specific element from the array. This can be achieved using array indices in the Extended Point Syntax. The syntax supports single and multiple dimensional arrays.

The Extended Point Syntax for array indices is:

tag:OPCServer\OPCTag[index]

or

tag:OPCServer\OPCTag[index1,index2]

NOTE: Individual array elements are always read-only. Only the array tag as a whole is writeable.

Attribute

When an OPC Server sends an update to the OPC client it sends three pieces of information: the value, the quality, and the timestamp. These attributes can be visualized using the Extended Point Syntax.

The attributes provided are:

- **Quality** - The exact quality and sub-quality provided by the OPC Server. This attribute shows the value from the OPC Server, not the value internally determined in GenClient. Because of this, if the connection the OPC Server breaks this attribute will still show the last received value from the OPC Server. To get the GenClient quality, use the quality() function of the expression engine instead.
- **Good quality** - A convenience value simply returning 0 or 1
- **Timestamp** - The timestamp sent with the last callback
- **Valid** - A 0 or 1 value indicating if the tag is valid.

The Extended Point Syntax for these attributes is:

tag:OPCServer\OPCTag#attribute