



OLE for Process Control

OPC Batch

Automation Interface

Specification

Release Candidate 2

18 January, 2000

Specification Type	Industry Standard Specification		
Title:	OPC Batch Automation Interface Specification	Date:	18 January, 2000
Version:	Release Candidate 2	Software:	MS-Word
		Source:	OPCB_Auto_10_RC2
Author:	Opc Foundation	Status:	Release Candidate

Synopsis:

This specification is the specification of the interface for developers of OPC clients and OPC servers. The specification is a result of an analysis and design process to develop a standard interface to facilitate the development of servers and clients by multiple vendors that shall inter-operate seamlessly together.

Trademarks:

Most computer and software brand names have trademarks or registered trademarks. The individual trademarks have not been listed here.

Required Runtime Environment:

This specification requires Windows 95, Windows 98, or Windows NT 4.0 or later

NON-EXCLUSIVE LICENSE AGREEMENT

The OPC Foundation, a non-profit corporation (the “OPC Foundation”), has established a set of standard OLE/COM interface protocols intended to foster greater interoperability between automation/control applications, field systems/devices, and business/office applications in the process control industry.

The current OPC specifications, prototype software examples and related documentation (collectively, the “OPC Materials”), form a set of standard OLE/COM interface protocols based upon the functional requirements of Microsoft’s OLE/COM technology. Such technology defines standard objects, methods, and properties for servers of real-time information like distributed process systems, programmable logic controllers, smart field devices and analyzers in order to communicate the information that such servers contain to standard OLE/COM compliant technologies enabled devices (e.g., servers, applications, etc.).

The OPC Foundation will grant to you (the “User”), whether an individual or legal entity, a license to use, and provide User with a copy of, the current version of the OPC Materials so long as User abides by the terms contained in this Non-Exclusive License Agreement (“Agreement”). If User does not agree to the terms and conditions contained in this Agreement, the OPC Materials may not be used, and all copies (in all formats) of such materials in User’s possession must either be destroyed or returned to the OPC Foundation. By using the OPC Materials, User (including any employees and agents of User) agrees to be bound by the terms of this Agreement.

LICENSE GRANT:

Subject to the terms and conditions of this Agreement, the OPC Foundation hereby grants to User a non-exclusive, royalty-free, limited license to use, copy, display and distribute the OPC Materials in order to make, use, sell or otherwise distribute any products and/or product literature that are compliant with the standards included in the OPC Materials.

All copies of the OPC Materials made and/or distributed by User must include all copyright and other proprietary rights notices include on or in the copy of such materials provided to User by the OPC Foundation.

The OPC Foundation shall retain all right, title and interest (including, without limitation, the copyrights) in the OPC Materials, subject to the limited license granted to User under this Agreement.

WARRANTY AND LIABILITY DISCLAIMERS:

User acknowledges that the OPC Foundation has provided the OPC Materials for informational purposes only in order to help User understand Microsoft’s OLE/COM technology. THE OPC MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF PERFORMANCE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. USER BEARS ALL RISK RELATING TO QUALITY, DESIGN, USE AND PERFORMANCE OF THE OPC MATERIALS. The OPC Foundation and its members do not warrant that the OPC Materials, their design or their use will meet User’s requirements, operate without interruption or be error free.

IN NO EVENT SHALL THE OPC FOUNDATION, ITS MEMBERS, OR ANY THIRD PARTY BE LIABLE FOR ANY COSTS, EXPENSES, LOSSES, DAMAGES (INCLUDING, BUT NOT LIMITED TO, DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL OR PUNITIVE DAMAGES) OR INJURIES INCURRED BY USER OR ANY THIRD PARTY AS A RESULT OF THIS AGREEMENT OR ANY USE OF THE OPC MATERIALS.

GENERAL PROVISIONS:

This Agreement and User's license to the OPC Materials shall be terminated (a) by User ceasing all use of the OPC Materials, (b) by User obtaining a superseding version of the OPC Materials, or (c) by the OPC Foundation, at its option, if User commits a material breach hereof. Upon any termination of this Agreement, User shall immediately cease all use of the OPC Materials, destroy all copies thereof then in its possession and take such other actions as the OPC Foundation may reasonably request to ensure that no copies of the OPC Materials licensed under this Agreement remain in its possession.

User shall not export or re-export the OPC Materials or any product produced directly by the use thereof to any person or destination that is not authorized to receive them under the export control laws and regulations of the United States.

The Software and Documentation are provided with Restricted Rights. Use, duplication or disclosure by the U.S. government is subject to restrictions as set forth in (a) this Agreement pursuant to DFARs 227.7202-3(a); (b) subparagraph (c)(1)(i) of the Rights in Technical Data and Computer Software clause at DFARs 252.227-7013; or (c) the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 subdivision (c)(1) and (2), as applicable. Contractor/ manufacturer is the OPC Foundation, P.O. Box 140524, Austin, Texas 78714-0524.

Should any provision of this Agreement be held to be void, invalid, unenforceable or illegal by a court, the validity and enforceability of the other provisions shall not be affected thereby.

This Agreement shall be governed by and construed under the laws of the State of Minnesota, excluding its choice or law rules.

This Agreement embodies the entire understanding between the parties with respect to, and supersedes any prior understanding or agreement (oral or written) relating to, the OPC Materials.

Table of Contents

1 INTRODUCTION..... 5

1.1 BACKGROUND5

1.2 PURPOSE.....5

1.3 REFERENCES.....5

1.4 RELATIONSHIP TO OTHER OPC SPECIFICATIONS.....5

1.5 SCOPE.....6

1.6 AUDIENCE6

1.7 DELIVERABLES.....6

2 FUNDAMENTAL CONCEPTS 7

2.1 FUNCTIONAL REQUIREMENTS.....7

2.2 OPC BATCH AUTOMATION OBJECT MODEL.....8

3 ABOUT THE OPC BATCH AUTOMATION WRAPPER DLL 10

4 OPC BATCH SERVER AUTOMATION INTERFACE QUICK REFERENCE..... 11

5 OPC BATCH SERVER AUTOMATION OBJECTS AND INTERFACES 12

5.1 OPCBATCHSERVER OBJECT 12

5.1.1 Summary of Properties 12

5.1.2 Summary of Methods 12

5.1.3 OPCBatchServer Properties 13

5.1.3.1 Delimiter 13

5.1.3.2 OPCBatchSummaryCollection 13

5.1.4 OPCBatchServer Methods..... 14

5.1.4.1 QueryEnumerationSets 14

5.1.4.2 QueryEnumeration 14

5.1.4.3 QueryEnumerationList..... 15

5.2 OPCBATCHSUMMARYCOLLECTION OBJECT 16

5.2.1 Summary of Properties 16

5.2.2 Summary of Methods 16

5.2.3 OPCBatchSummaryCollection Properties 16

5.2.3.1 Count..... 16

5.2.4 OPCBatchSummaryCollection Methods..... 17

5.2.4.1 Item 17

5.2.4.2 GetOPCBatchSummary 17

5.3 OPCBATCHSUMMARYOBJECT 18

5.3.1 Summary of Properties 18

5.3.2 OPCBatchSummary Properties..... 18

5.3.2.1 ID 18

5.3.2.2 Description..... 18

5.3.2.3 OPCItemID 18

5.3.2.4 MasterRecipeID..... 19

5.3.2.5 BatchSize 19

5.3.2.6 EU 19

5.3.2.7 ExecutionState 19

5.3.2.8 ExecutionMode..... 19

5.3.2.9 ActualStartTime..... 20

5.3.2.10 ActualEndTime..... 20

6 SUMMARY OF OPC ERROR CODES 21

7 APPENDIX A – OPC BATCH AUTOMATION IDL SPECIFICATION 22

1 Introduction

1.1 Background

As products are developed for the batch processing industry based on the IEC 61512-1 Batch Control – Part 1: Models and Terminology standard there is an increasing need to exchange data between these products and other systems. Interfaces occur at all levels; with Field Management devices (e.g. monitoring stations, control stations...), Process Management systems (e.g. lab systems, batch control systems, loading, unloading, dispensing, weighing systems...), and with Business Management systems (e.g. ERP and MES).

The data exchange covers four basic types of information such as equipment capabilities, current operating conditions, historical and recipe contents. Version 1.0 of this specification has been limited to equipment capabilities and current operating conditions. Equipment capabilities correspond to the IEC 61512-1 Physical Model. Current operating conditions are described using a Batch List and a Batch Model which combines Batch and Control Recipe information.

Currently most batch systems use their own proprietary interfaces for dissemination and collection of data. There is no capability to augment existing solutions with other capabilities in a plug-n-play environment. This requires the developer to recreate the same infrastructure for their products as all other vendors have had to develop independently with no interoperability with any other systems.

Manufacturers and consumers want to use off the shelf, open solutions from vendors that offer superior value that solves a specific need or problem. This specification is targeted to allow Visual Basic applications, as well as other automation enabled applications, to exchange batch data with various computer systems with less complexity than the OPC Batch Custom Interface.

1.2 Purpose

To provide a means to pass runtime batch and equipment data between components which would be suitable to standardization. Additionally this document details the design of interfaces and namespaces in such a way as to complement the existing OPC Data Access Interfaces.

1.3 References

- ?? OPC Batch Custom Interface, Version 1.0, January 18, 2000.
- ?? OPC Data Access Custom Interface Specification, Version 2.0, OPC Foundation, October 13, 1998.
- ?? OPC Data Access Automation Interface Specification, Version 2.01, OPC Foundation, January 6, 1999.
- ?? OPC Common Definitions, Version 1.0, OPC Foundation, October 27, 1998.
- ?? IEC61512-1:1997, Batch control- Part 1: Models and terminology

1.4 Relationship to Other OPC Specifications

This specification defines automation support for the batch namespace defined in the OPC Batch Custom Interface 1.0 Specification. The Introduction, Fundamental Concepts and Architecture of the OPC Batch Custom Interface 1.0 Specification should be read before reading this document.

1.5 Scope

This document defines automation support for the batch namespace defined in the OPC Batch Custom Interface 1.0 Specification. No namespace changes are made in this document.

1.6 Audience

This document is intended to be used as reference material for developers of OPC Batch automation clients and servers. It is assumed that the reader is familiar with Microsoft OLE/COM technology, the needs of the process control industry, the OPC Data Access 2.0 specification and the OPC Batch Custom Specification 1.0.

1.7 Deliverables

The deliverables from the OPC Foundation with respect to the OPC Batch Automation Specification 1.0 include the OPC Specification itself and OPC Batch Automation IDL files (included in this document as Appendices. This OPC Batch Automation Specification contains design information for the following:

1. **The OPC Batch Automation Interface** - The OPC Automation Interfaces which facilitates the use of Visual Basic and other Automation enabled products to interface with OPC Batch Servers.
2. **The OPC Batch Automation Wrapper DLL** - A reference sample of the Batch Automation interface for the OPC foundation members to use in providing an automation interface to OPC Batch custom interface servers. The reference sample is provided as a DLL complete with the Visual C++ source code. Vendors may provide the DLL directly with their product.

2 Fundamental Concepts

The fundamental design goal is that this interface is intended to work as a "wrapper" for existing OPC Batch Custom Interface servers providing an automation friendly mechanism to the functionality provided by the custom interface.

The OPC Batch Custom Interface Specification version 1.0 section 3.0 Architecture provides background on the OPC batch namespace. This will be helpful in the use of the batch automation interface.

2.1 Functional Requirements

- ?? The automation interface provides nearly all of the functionality of the required and optional Interfaces in the OPC Batch Custom Interface. If the OPC Batch Custom server supports the interface, the functions and properties at the automation level will work. Automation interfaces generally do not support optional capabilities in the same way that the custom interface does. If the underlying custom interface omits some optional functionality then the corresponding automation functions and properties will exhibit some reasonable default behavior as described in more detail later in this document.
- ?? The interfaces are fully supported by VC++ and Visual Basic 6.0. They allow any application which has an OLE Automation Interface (e.g. VB, VC++, and VBA enabled applications) to access the OPC Interface, according to the limitations of the respective application.
- ?? The interface described in this specification specifically does NOT support VBScript or Java Script. A separate wrapper could be developed to accommodate the needs of VBScript and Java Script. However such an effort is outside the scope of this specification.

2.2 OPC Batch Automation Object Model

The automation interface provides all of the functionality of the required and optional interfaces in the OPC Batch Custom Interface. It additionally provides the OPC Data Access Automation Interfaces and functionality. The OPC Data Access Automation server object is used to provide the connection to the OPC Batch Server.

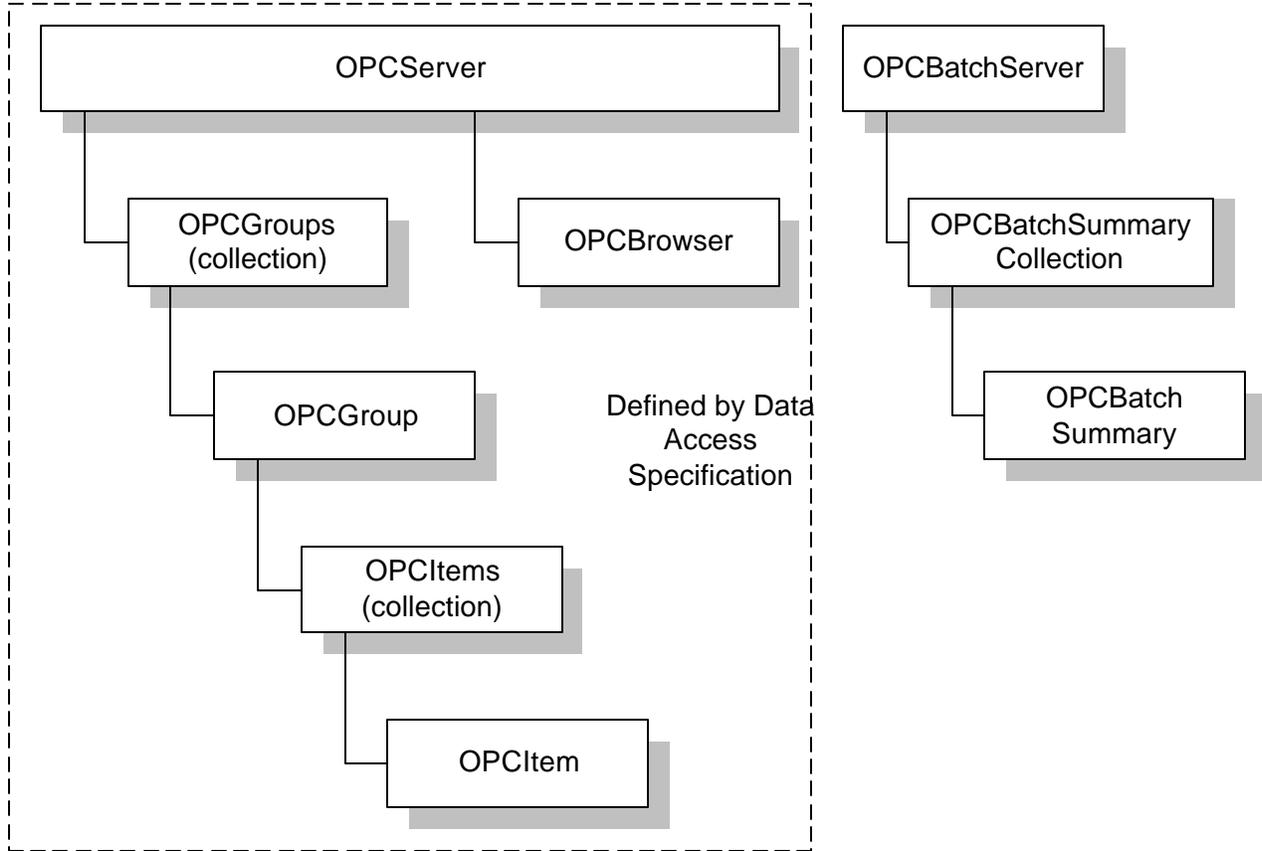


Figure 2-1: OPC Batch Automation Objects

Table 1- OPC Batch Automation Objects

Object	Description
OPCServer	An instance of an OPC Server. You must create an OPCServer object before you can get references to other objects. It contains the OPCGroups Collection and creates OPCBrowser objects.
OPCGroups	An Automation collection containing all of the OPCGroup objects this client has created within the scope of the OPCServer that the Automation Application has connected to via the OPCServer.Connect()
OPCGroup	An instance of an OPCGroup object. The purpose of this object is to maintain state information and provide the mechanism to provide data acquisition services for the OPCItem Collection object that the OPCGroup object references.

Object	Description
OPCItems	An Automation collection containing all of the OPCItem objects this client has created within the scope of the OPCServer, and the corresponding OPCGroup object that the Automation Application has created.
OPCItem	An automation object that maintains the item's definition, current value, status information, last update time. Note the Custom Interface does not provide a separate Item Object.
OPCBrowser	An object that browses item names in the server's configuration. There exists only one instance of an OPCBrowser object per instance of an OPC Server object.
OPCBatchServer	An instance of an OPC Batch Server. You must first create an OPCServer object before you can get references to the OPCBatchServer object.
OPCBatchSummary Collection	An Automation collection containing all of the OPCBatchSummary objects that represent the known batches in the batch server.
OPCBatchSummary	An instance of an OPCBatchSummary object. The purpose of this object is to maintain state information regarding a single batch entity

3 About the OPC Batch Automation Wrapper DLL

The OPC Batch Automation Interface requires full compliance with the OPC Data Access Automation Specification. The OPC Batch Automation Interface extends the functionality and support of the OPC Data Access Automation Interface. This fully compliant support provides the connection capabilities for the OPC Batch Automation Interface. It is intended that a client will connect using the OPC Data Access Automation Interface (specifically, the OPCServer objects Connect method) to establish a connection to an OPC Batch Server. Once the connection has been established, the client can create an instance of the OPCBatchServer object.

The OPC foundation provides a reference sample of the Batch Automation interface for the OPC foundation member's use in providing an automation interface to OPC Batch custom interface servers. The reference sample is provided as a DLL complete with the Visual C++ source code. Vendors may provide the DLL directly with their product.

Vendors that choose to modify the source code, or even just build the DLL from the source code (unchanged) must do the following prior to including or shipping the DLL.

1. The name of the OPC automation DLL must be changed from opcba.dll to a vendor specific unique name.
2. The name of the OPC automation IDL (opcba.idl) file should be changed to a vendor specific unique name.
3. The helpstring ("OPC Batch Automation 1.0") in the IDL file must be changed to reflect your vendor specific OPC automation interface. This is the name that shows up in the Automation Type Library. Visual Basic applications that use your vendor build OPC automation interface DLL will include the DLL by using the type library.
4. All guid's in the IDL file must be changed to new values that are generated by using the Guidgen tool. This is required to prevent the vendor built automation interface library from being confused with another vendors built automation library or the OPC foundation provided automation library.

The vendor is encouraged to not change the existing automation interfaces. If additional functionality is desired, a new object and interface should be added and should replicate all the functionality of the existing object that is being added to.

The OPC foundation has also provided a visual basic sample that demonstrates usage of the Data Access Automation interface. This sample is intended only to demonstrate the functionality of the OPC data access automation interface.

4 OPC Batch Server Automation Interface Quick Reference

This section includes a quick reference for the methods in the Automation Interface. These interfaces, their parameters, and behavior are defined in detail in section 5.

The OPC Batch Automation Specification specifies the Automation Interface and defines the following objects:

- ?? OPCBatchServer
- ?? OPCBatchSummaryCollection
- ?? OPCBatchSummary

From the OPC Data Access Automation specification, the following Objects are defined:

- ?? OPCServer
- ?? OPCGroups
- ?? OPCGroup
- ?? OPCItems
- ?? OPCItem
- ?? OPCBrowser

5 OPC Batch Server Automation Objects and Interfaces

5.1 OPCBatchServer Object

Description	A client creates an OPCBatchServer automation object from a connected OPCServer object.
Syntax	OPCBatchServer
Example	<pre> ' Create the OPC Batch Server object Dim WithEvents anOPCServer As OPCServer Set anOPCServer = New OPCServer ' Connect to the OPC Batch Server Dim aRealOPCServer As String Dim aRealOPCNodeName As String aRealOPCServer = "VendorX.BatchServer" aRealOPCNodeName = "SomeComputerNodeName" anOPCServer.Connect(aRealOPCServer, aRealOPCNodeName) ' Create the OPC Batch Server Object and instantiate Dim anOPCBatchServer as OPCBatchServer Set anOPCBatchServer = anOPCServer </pre>

5.1.1 Summary of Properties

Delimiter	OPCBatchSummaryCollection	
-----------	---------------------------	--

5.1.2 Summary of Methods

QueryEnumerationSets	QueryEnumeration	QueryEnumerationList
----------------------	------------------	----------------------

5.1.3 OPCBatchServer Properties

5.1.3.1 Delimiter

Description	(Read-only) Returns the delimiter.
Syntax	<code>Delimiter As String</code>
Remarks	This is the Unicode string that separates the elements in a fully qualified Item ID. This is used by clients wishing to compose fully qualified Item IDs from the items returned via the <code>IOPCBrowseServerAddressSpace::BrowseOPCItemIDs</code> method or by clients wishing to parse fully qualified Item IDs returned by the server.
Example	<code>Dim Delimiter As String Delimiter = AnOPCServer.Delimiter</code>

5.1.3.2 OPCBatchSummaryCollection

Description	(Read only) A collection of OPCBatchSummary objects. This is the default property for an OPCBatchServer object.
Syntax	<code>anOPCBatchSummaryList As OPCBatchSummaryCollection</code>
Example	<code>`(explicit property specification): Dim anOPCBatchSummaryList As OPCBatchSummaryCollection Set anOPCBatchSummaryList = anOPCBatchServer.OPCBatchSummaryCollection ` `(using the default specification): Dim anOPCBatchSummaryList As OPCBatchSummaryCollection Set anOPCBatchSummaryList = anOPCBatchServer</code>

5.1.4 OPCBatchServer Methods

5.1.4.1 QueryEnumerationSets

Description	Returns the list of Enumeration Set IDs and Names that the Batch Server supports.
Syntax	<code>QueryEnumerationSets (ByRef Count As Long, ByRef EnumSetID() As Long, ByRef EnumSetName() As String)</code>
Remarks	Refer to the OPC Batch Custom Interface Specification version 1.0 for details of parameters.

Part	Description
Count	Count of Enumeration Sets in the server address space
EnumSetID	Array of Enumeration Set IDs.
EnumSetName	Array of Enumeration Set names corresponding to the Enumeration Set IDs.

Example

```

Dim itemCount As Long
Dim EnumSetID() As Long
Dim EnumSetName() As String
Dim AnOPCEnumerationSet As String

AnOPCBatchServer.QueryEnumerationSets(itemCount,
EnumSetID, EnumSetName)

For i = 1 to itemCount
    AnOPCEnumerationSet = Str(EnumSetID(i)) + ":" +
EnumSetName(i)
    listbox.AddItem(AnOPCEnumerationSet)
Next i
    
```

5.1.4.2 QueryEnumeration

Description	Returns the associated Enumeration Name for the provided Enumeration Set and Enumeration Value.
Syntax	<code>QueryEnumeration(EnumSetID As Long, EnumValue As Long) As String</code>

Part	Description
EnumSetID	Enumeration Set to query
EnumValue	Enumeration Value to query

Remarks	Refer to the OPC Batch Custom Interface Specification version 1.0 for details of parameters.
Example	<pre>Dim AnOPCEnumSetID As Long Dim AnOPCEnumValue As Long ' For this example, assume that the enumeration set and enumeration ' value have been obtained as a result of a parameter value Dim AnOPCEnumerationName As String AnOPCEnumerationName = AnOPCServer.QueryEnumeration(AnOPCEnumSetID, AnOPCEnumValue)</pre>

5.1.4.3 QueryEnumerationList

Description	Returns the list of Enumeration Values and Names for the provided Enumeration Set.
Syntax	<pre>QueryEnumerationList(EnumSetID As Long, ByRef Count As Long, ByRef EnumValue() As Long, ByRef EnumName() As String)</pre>
Remarks	Refer to the OPC Batch Custom Interface Specification version 1.0 for details of parameters.

Part	Description
EnumSetID	Enumeration Set to query
Count	Count of the number of enumerations in the enumeration set
EnumValue	Array of Enumeration values associated with the specified enumeration set.
EnumName	Array of Enumeration names corresponding to the enumeration set and values.

Example	<pre>' For this example assume client wants to download all enumeration sets ' and values from the server ' First get the list of sets Dim SetCount As Long Dim EnumSetID() As Long Dim EnumSetName() As String AnOPCBatchServer.QueryEnumerationSet(SetCount, EnumSetID, EnumSetName) ' Then get the list of values for each set</pre>
---------	--

```

Dim ValueCount As Long
Dim EnumValue() As Long
Dim EnumName() As String

Dim AnOPCEnumSet As String
Dim AnOPCEnumValue As String

For i = 1 to SetCount
    AnOPCBatchServer.QueryEnumerationList(EnumSetID(i),
    ValueCount,
    EnumValue, EnumName)

    AnOPCEnumSet = Str(EnumSetID(i)) + ":" + EnumSetName(i)
    For j = 1 to ValueCount
        AnOPCEnumValue = ", " + Str(EnumValue(j)) + ":" +
EnumName(j)
        AnOPCEnumSet = AnOPCEnumSet + AnOPCEnumValue
    Next j
    MessageBox AnOPCEnumSet
Next I

```

5.2 OPCBatchSummaryCollection Object

Description	(Read only) A collection of OPCBatchSummary objects. This is the default property of the OPCBatchServer object.
Syntax	OPCBatchSummaryCollection
Example	<pre> \'(explicit property specification): Dim anOPCBatchSummaryList As OPCBatchSummaryCollection Set anOPCBatchSummaryList = anOPCBatchServer.OPCBatchSummaryCollection \'(using the default specification): Dim anOPCBatchSummaryList As OPCBatchSummaryCollection Set anOPCBatchSummaryList = anOPCBatchServer </pre>

5.2.1 Summary of Properties

Count		
-------	--	--

5.2.2 Summary of Methods

Item	GetOPCBatchSummary	
------	--------------------	--

5.2.3 OPCBatchSummaryCollection Properties

5.2.3.1 Count

Description	(Read-Only) Required property for collections.
-------------	--

Syntax Count as Long

Example For Index = 1 to anOPCBatchList.Count
 ` some code here `

 Next index

5.2.4 OPCBatchSummaryCollection Methods

5.2.4.1 Item

Description Returns an OPCBatchSummary by ItemSpecifier. ItemSpecifier is either the name or 1-based index into the collection.

Syntax Item(ItemSpecifier As Variant) As OPCBatchSummary

Part	Description
ItemSpecifier	ItemSpecifier is either the OPCBatchSummary name or the 1-based index into the collection

Example Dim anOPCBatch As OPCBatchSummary
 Set anOPCBatch = anOPCBatchSummaryCollection.Item(3)

 `Or

 Set anOPCBatch =
 anBatchOPCBatchSummaryCollection("OPCBatchModel.Batch123")

5.2.4.2 GetOPCBatchSummary

Description Returns an OPCBatchSummary object by ItemSpecifier. Item specifier is either the ID within the OPCBatchSummary object. Use Item to reference by index.

Syntax GetOPCBatchSummary(ItemSpecifier As Variant) As OPCBatchSummary

Part	Description
ItemSpecifier	ItemSpecifier is either the OPCBatchSummary name or the 1-based index into the collection

Example Dim anOPCBatchSummary as OPCBatchSummary

 Set anOPCBatchSummary =
 anOPCBatchSummaryCollection.GetOPCBatchSummary("OPCBatchModel.Batch123")

5.3 OPCBatchSummaryObject

Description	An OPCBatchSummary object within the OPC Batch Server
Syntax	OPCBatchSummary

5.3.1 Summary of Properties

ID	Description	OPCItemID
MasterRecipeID	BatchSize	EU
ExecutionState	ExecutionMode	Actual StartTime
ActualEndTime		

5.3.2 OPCBatchSummary Properties

5.3.2.1 ID

Description	(Read-Only) The vendor specific Batch Identifier
Syntax	ID as String
Example	<pre>Dim anOPCBatchSummary as OPCBatchSummary Dim anID as String anOPCBatchSummary = anOPCBatchSummaryCollection.Item(3) anID = anOPCBatchSummary.ID</pre>

5.3.2.2 Description

Description	(Read-Only) Text string describing the batch
Syntax	Description as String
Example	<pre>Dim anOPCBatchSummary as OPCBatchSummary Dim aDescription as String anOPCBatchSummary = anOPCBatchSummaryCollection.Item(3) aDescription = anOPCBatchSummary.Description</pre>

5.3.2.3 OPCItemID

Description	(Read-Only) The OPC Item id for this batch object in the batch namespace
Syntax	OPCItemID as String
Example	<pre>Dim anOPCBatchSummary as OPCBatchSummary Dim anOPCItemID as String anOPCBatchSummary = anOPCBatchSummaryCollection.Item(3) anOPCItemID = anOPCBatchSummary.OPCItemID</pre>

5.3.2.4 MasterRecipeID

Description	(Read-Only) ID of master recipe the batch was derived from
Syntax	MasterRecipeID as String
Example	<pre>Dim anOPCBatchSummary as OPCBatchSummary Dim aMasterRecipeID as String AnOPCBatchSummary = anOPCBatchSummaryCollection.Item(3) aMasterRecipeID = anOPCBatchSummary.MasterRecipeID</pre>

5.3.2.5 BatchSize

Description	(Read-Only) Amount of material that this batch will produce
Syntax	BatchSize as Float
Example	<pre>Dim anOPCBatchSummary as OPCBatchSummary Dim aBatchSize as Float AnOPCBatchSummary = anOPCBatchSummaryCollection.Item(3) aBatchSize = anOPCBatchSummary.BatchSize</pre>

5.3.2.6 EU

Description	(Read-Only) Engineering units for batch size
Syntax	EU as String
Example	<pre>Dim anOPCBatchSummary as OPCBatchSummary Dim anEU as String anOPCBatchSummary = anOPCBatchSummaryCollection.Item(3) anEU = anOPCBatchSummary.EU</pre>

5.3.2.7 ExecutionState

Description	(Read-Only) Current execution state using vendor's state names
Syntax	ExecutionState as String
Example	<pre>Dim anOPCBatchSummary as OPCBatchSummary Dim anExecutionState as String anOPCBatchSummary = anOPCBatchSummaryCollection.Item(3) anExecutionState = anOPCBatchSummary.ExecutionState</pre>

5.3.2.8 ExecutionMode

Description	(Read-Only) Current execution mode using vendor's mode names
Syntax	ExecutionMode as String
Example	<pre>Dim anOPCBatchSummary as OPCBatchSummary Dim anExecutionMode as String AnOPCBatchSummary = anOPCBatchSummaryCollection.Item(3) AnExecutionMode = anOPCBatchSummary.ExecutionMode</pre>

5.3.2.9 ActualStartTime

Description	(Read-Only) Time when batch actually started
Syntax	<code>ActualStartTime as Date</code>
Example	<pre>Dim anOPCBatchSummary as OPCBatchSummary Dim anActualStartTime as Date AnOPCBatchSummary = anOPCBatchSummaryCollection.Item(3) AnActualStartTime = anOPCBatchSummary.ActualStartTime</pre>

5.3.2.10 ActualEndTime

Description	(Read-Only) Time when batch actually ended
Syntax	<code>ActualEndTime as Date</code>
Example	<pre>Dim anOPCBatchSummary as OPCBatchSummary Dim anActualEndTime as Date AnOPCBatchSummary = anOPCBatchSummaryCollection.Item(3) AnActualEndTime = anOPCBatchSummary.ActualEndTime</pre>

6 Summary of OPC Error Codes

The OPC Batch Automation Specification does not define additional error codes beyond those defined in the OPC Batch Custom Interface Specification version 1.0 and the OPC Data Access Automation Specification version 2.0.

7 Appendix A – OPC Batch Automation IDL Specification