Tech Note 765 Implementing Block Reads Using the OPCClient Object

All Tech Notes, Tech Alerts and KBCD documents and software are provided "as is" without warranty of any kind. See the Terms of Use for more information.

Topic#: 002544 Created: March 2011

Introduction

The OPCClient object is a DeviceIntegration (DI) object that allows access to a running OPC Data Access (DA) Server. An OPCClient object supports the following operations on I/O points for the OPC DA Server:

- Subscriptions, which are implemented via scan groups.
- Read transactions, which are implemented via block reads.
- · Write transactions, which are implemented via block writes.

This Tech Note describes implementing Block Reads.

Note: This *Tech Note* assumes you have a basic understanding of Wonderware Application Server and DAServers. You should also be familiar with OPC communications. For information about OPC communications, see **Tech Note 761 OPC Connections at a Glance**.

Application Versions

· Wonderware Application Server 3.1 and later

Block Read Description

A block read is a set of user-defined attributes for which you want to retrieve values in a single transaction. Instead of the I/O points being on advise, as with a scan group, the attributes are updated once per transaction. A block read must be initiated from a user or script via the BlockRead.TransactionTrigger attribute.

For this Tech Note, we create a basic configuration that performs a block read from a local MBTCP DAServer via the OPCClient object.

Set Up a Basic Galaxy

Follow the steps below to create a basic galaxy and objects required for this technote:

- 1. Open the IDE and create a Galaxy.
- 2. Create an instance of the \$WinPlatform object called **Platform**.
- 3. Create an instance of the \$AppEngine object called Engine.
- 4. Create an instance of the \$OPCClient object called OPC.
- 5. Create an instance of the \$Area object called Area.
- 6. Create an instance of the \$UserDefined object called Tags.
- 7. In **Deployment** view, arrange the object instances as follows:

Implementing Block Reads Using the OPCClient Object

Co Deployment	-		×
Base_Application_Server Gene Gene			~
Trans (1		1920	0
<u><</u>		2	
Model 🖸 Deployment 💊 Deriv	vation		
FIGURE 1: OBJECT ARRANGEMENT) EP	

Configuring DASMBTCP

Note: This Tech Note assumes you have a basic understanding of a DAServer and can configure the DAServer to communicate to a PLC.

To configure DASMBTCP

- 1. Open the SMC.
- 2. Navigate to DASMBTCP in the DAServer manager section.
- 3. Create a port object called Port.
- 4. Create a PLC object called PLC.
- 5. Configure the PLC object to communicate to a PLC.

🖉 SMC - [ArchestrA System Management Console			
<u>File A</u> ction <u>V</u> iew <u>H</u> elp			
ArchestrA System Management Console Galaxy Database Manager DAServer Manager Default Group Coal Coal Configuration Port Configuration	Node Type: ModbusPLC Delim PLC Parameters Device Groups Device Items Network address: 10.2.82.112 Reply timeout (sec): 3	iter: . Port number: 502 Maximum outstanding messages: 4	
 ■ Log Viewer ■ Platform Manager 	 ✓ Use Concept data structures (Longs) ✓ Support multiple coil write Close Ethernet connection when no activity. Bit order format: B1 B2 B16 ▼ String variable style ✓ Full length C style Pascal style Block I/O size Discrete input/coil read: 1976 Register read: 122 	 ✓ Use Concept data structures (Reals) ✓ Support multiple register write ✓ Swap string bytes Register size (digits): 6 ✓ Register type ✓ Binary ✓ BCD Coil write: 800 Register write:	

FIGURE 2: PLC OBJECT COMMUNICATION SETTINGS

Configure the OPCClient object

- 1. Open the OPCClient object called OPC.
- 2. Under the General tab, select the Server name.
- 3. For this example, the Server name is ArchestrA.DASMBTCP.2. Leave all other default attribute settings.

1 OPC *				Ca ? 🖬	,
General Scan Group Block Read Bl	ock Write 📔 Object Information	n Scripts UDAs	Extensions Graphics		
Server node:	[£	9		
Server name:	ArchestrA.DASMBTCP.	2 🖌 🗗	Q		
Run server out-of-proc		പ്			
🗌 Use scan group name as access p	ath	പ്			
Restart attempts:	3	പ്	U		
Restart period:	30000 ms	പ്	W		
🔲 Detect restart alarm		പ്			
Priority:		63			
Connection heartbeat period:	10000 ms	£	W		
Detect connection alarm		£			
Priority:		63			
Restart reset security:			Q		

FIGURE 3: GENERAL TAB SERVER NAME AND DEFAULTS

4. Click the Block Read tab.

This is where we enter the block of PLC addresses that we want to read in a single transaction.

For this example, we will read addresses 400001, 400002, 400003, 400006, 400008, and 400200. We will also assign an attribute to each register.

Block Read	Transaction	Timeout (ms)	Access Path	
ReadPLC	0			
ssociated attributes for Read	PLC:	[£ 🖪	
Attribute		Item Reference		
Registeri Degisteri		Port PLC 40000	12	
Register3		Port PLC 40000	13	
Register6		Port.PLC.40000	 16	
Register8		Port.PLC.40000	18	
Register200		Port.PLC.40020	0	

FIGURE 4: BLOCK READ ADDRESSES

Note: Although a Scan Group is not required for Block Reads, go to the "Scan Group" tab and enter a scan group. This will allow the saving of the OPC object without any warnings.

Configure the UserDefined Object

- 1. Open the Tags UserDefined Object.
- 2. Create six analog Field attributes (Figure 5 below).
- 3. Provide each attribute with an input source that points to the six block read attributes as shown in the following table.

Field Attribute Name	Input Source
Tag1	OPC.ReadPLC.Register1
Tag2	OPC.ReadPLC.Register2
Tag3	OPC.ReadPLC.Register3
Tag6	OPC.ReadPLC.Register6
Tag8	OPC.ReadPLC.Register8
Tag200	OPC.ReadPLC.Register200

X ArchestrA IDE			
Galaxy Edit View Object Window Help			
B B Q Q A 0 Q A 0 Q B Q	🛛 🗙 🚉 🎭 😵 🗠 ኈ 💊 😂 🧭 🔕		
💌 Template Toolbox 🛛 👻 🔻 🗙	Tags *		🕞 ? 🗟 🗙
🕀 🛷 Base_Application_Server	Field Attributes Object Information Scripts U	DAs Extensions Graphics	
			<u> </u>
		Name: Tag1 Attribute type: Analog	
	Field attributes:	Access mode: InputOutput 🗾 Data type: Integer 🗾	
	Name	Category: User writeable	
	Tagi	Description:	
		(m)	
	Tag3	Value	
	Tag6	0 🗗 🥥 Value deadband: 0.0	63
	Tag8	🗖 Generate event upon change 🖆 🕔 Engineering units:	6
		- I/O — 🎦 🧊	
-		Input source: OPC.ReadPLC.Register1	63
		Output destination differs from input source	6
Umplate Toolbox V Graphic Toolbox		Output destination:	63
Deployment 🗸 🕈 🗙	Telessited field attain test	Enable I/O scaling	۲
Generation_Server	Name	Enable history	۲
Platform		🔽 Enable limit alarms	۲
Engine		Enable rate of change alarms	*
		Enable target deviation alarms	۲
		Enable bad value alarm	*
		Enable statistics	*

FIGURE 5: ANALOG FIELD ATTRIBUTE AND INPUT SOURCE

Test the Configuration

- 1. Deploy the Platform, Engine, Area, Tags and OPC objects. After all objects are deployed, open Object Viewer.
- 2. Add the following attributes to the watch list:
- Tags.Tag1
- Tags.Tag2
- Tags.Tag3
- Tags.Tag6
- Tags.Tag8
- Tags.Tag200

Implementing Block Reads Using the OPCClient Object

- OPC.ConnectionStatus (Displays the connection status between the OPC object and the OPC Server)
- OPC.ReadPLC.TransactionTrigger (Write a "True" to trigger a block read)
- OPC.ReadPLC.TransactionStart (Displays the time and date when the block read starts)
- OPC.ReadPLC.TransactionEnd (Displays the time and date when the block read ends)

The Watch List window should look like Figure 6 (below).

💋 Object Viewer					
<u>File Edit View Options H</u> elp					
🛛 🗞 🌃 🛤 💕 🛃 🗍 Attribute Reference:	Tags.Tag1.Input.InputSource.value	Go			
E Sase_Application_Server	Attribute Name 🔺	Value		Timestamp	
 ☐ - ⊕ ☐ - ⊕ ☐ - ⊕ Area [Area] ☐ - ⊕ ☐ Tags [Tags] ☐ - ⊕ ☐ OPC [OPC] 	ReadPLC.ItemErrorCntAlarm.AlarmModeCmd ReadPLC.ItemErrorCntAlarm.Category ReadPLC.ItemErrorCntAlarm.DescAttrName ReadPLC.ItemErrorCntAlarm.InAlarm ReadPLC.ItemErrorCntAlarm.Priority ReadPLC.ItemErrorCntAlarm.TimeAlarmAcked ReadPLC.ItemErrorCntAlarm.TimeAlarmOff ReadPLC.ItemErrorCntAlarm.TimeAlarmOn ReadPLC.ItemErrorCntAlarm.TimeAlarmOn	Enable Discrete me.Short false 500 0	:Desc	3/31/2011	12:4
	ReadPLC.ItemErrorCntLastEnd ReadPLC.ItemErrorCntLastStart ReadPLC.ItemErrorCntStart ReadPLC.Register1 ReadPLC.Register2 ReadPLC.Register200 ReadPLC.Register3 ReadPLC.Register6	0 0 0 0 0	1	3/31/2011 3/31/2011 3/31/2011 3/31/2011 3/31/2011	12:4 12:4 12:4 12:4 12:4
AttributeReference	Value	Timestamp	Quality	Status	
Tags.Tag1	0	3/31/2011 12:47:00.875 PM	20:Initializing	Ok	
Tags.Tag2	0	3/31/2011 12:47:00.875 PM	20:Initializing	Ok	
Tags.Tag3	0	3/31/2011 12:47:00.875 PM	20:Initializing	Ok	
Tags.Tag6	0	3/31/2011 12:47:00.875 PM	20:Initializing	Ok	
Tags.Tag8	0	3/31/2011 12:47:00.875 PM	20:Initializing	Ok	
Tags.Tag200	0	3/31/2011 12:47:00.875 PM	20:Initializing	Ok	
OPC.ConnectionStatus	Connected	3/31/2011 12:46:34.632 PM	C0:Good	Ok	
OPC.ReadPLC.TransactionTrigger	false	3/31/2011 12:46:34.632 PM	C0:Good	Ok	
OPC.ReadPLC.TransactionStart		3/31/2011 12:46:34.632 PM 3/31/2011 12:46:34 632 PM	CO:Good	Ok Ok	
		Spergeorr renois nose Ph		- SK	
1 Match List 1					•
		FILE: User: DefaultUs	er	Mode: User	_

FIGURE 6: WATCH LIST WINDOW WITH OPC ATTRIBUTES

3. Trigger a block read by writing a True value to OPC.ReadPLC.TransactionTrigger.

Implementing Block Reads Using the OPCClient Object

Modify Boolean ¥alue		
Reference: OPC.ReadPLC.Tran	nsactionTrigger	
True	C False	
Apply	Ok	Cancel

FIGURE 7: TRIGGER BLOCK READ

If everything is configured correctly, the values are read from the PLC every time the OPC.ReadPLC.TransactionTrigger is set to True.

OPC.ReadPLC.TransactionStart and OPC.ReadPLC.TransactionEnd will display the transaction start and end date and times (Figure 8 below).

💋 Object Viewer				_ [
<u>File E</u> dit <u>V</u> iew <u>O</u> ptions <u>H</u> elp					
🛛 🗞 🌃 🏘 💕 🛃 🗍 Attribute Reference:	Tags.Tag1.Input.InputSource.value	Go			
E generation_Server	Attribute Name 🔺	Value		Timestamp	•
 □ □	ReadPLC.ItemErrorCntAlarm.AlarmModeCmd ReadPLC.ItemErrorCntAlarm.Category ReadPLC.ItemErrorCntAlarm.InAlarm ReadPLC.ItemErrorCntAlarm.InAlarm ReadPLC.ItemErrorCntAlarm.Priority ReadPLC.ItemErrorCntAlarm.TimeAlarmAcked ReadPLC.ItemErrorCntAlarm.TimeAlarmOff ReadPLC.ItemErrorCntAlarm.TimeAlarmOn ReadPLC.ItemErrorCntLast ReadPLC.ItemErrorCntLast ReadPLC.ItemErrorCntLast ReadPLC.ItemErrorCntLastEnd ReadPLC.ItemErrorCntLastStart ReadPLC.ItemErrorCntStart ReadPLC.Register1 ReadPLC.Register2 ReadPLC.Register200	Enable Discrete me.Shor false 500 0 0 0 0	tDesc	3/31/2011 12 3/31/2011 12 3/31/2011 12 3/31/2011 12 3/31/2011 12	2:4 2:4 2:4 2:4
	ReadPLC.Register3 ReadPLC.Register6	0. 0		3/31/2011 12 3/31/2011 12	2;4 2:4 ▼ ▶
AttributeReference	Value	Timestamp	Quality	Status	
Tags.Tag1	54	3/31/2011 1:05:09.507 PM	C0:Good	Ok	
Tags.Tag2	566	3/31/2011 1:05:09.507 PM	C0:Good	Ok	
Tags.Tag3	54	3/31/2011 1:05:09.510 PM	C0:Good	Ok	
Tags.Tag6	566	3/31/2011 1:05:09.510 PM	C0:Good	Ok	
Tags.Tag8	566	3/31/2011 1:05:09.510 PM	C0:Good	Ok	
Tags.Tag200	54	3/31/2011 1:05:09.508 PM	C0:Good	Ok	
OPC.ConnectionStatus	Connected	3/31/2011 1:04:48.217 PM	CO:Good	Ok	
OPC.ReadPLC. Transaction Trigger		3/31/2011 1:04:48.217 PM	CU:Good	OK	
OPC.ReadPLC.TransactionStart	3/31/2011 1:05:09.197 PM 3/31/2011 1:05:10 010 PM	3/31/2011 1:04:48.217 PM 3/31/2011 1:04:48 217 PM	CO:Good	Ok	
		Sportcorr non ioner/ Ph	2014004		
1 Match List 1					•
		EILE: Liser: Default is	er	Mode: Liser	

FIGURE 8: BLOCK READ WITH TIMESTAMPS

G. Alldredge

software products at Wonderware Technical Support.

For technical support questions, send an e-mail to support@wonderware.com.

Back to top

©2011 Invensys Systems, Inc. All rights reserved. No part of the material protected by this copyright may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, broadcasting, or by anyinformation storage and retrieval system, without permission in writing from Invensys Systems, Inc. Terms of Use.