

Tech Note 526

Automating Object Configuration Tasks Using GRAccess: Configuring the OPCClient DIOject

All Tech Notes and KBCD documents and software are provided "as is" without warranty of any kind. See the [Terms of Use](#) for more information.

Topic#: 002262
Created: March 2008

Introduction

The GRAccess Toolkit enables developers to automate activities that users normally perform manually using the Industrial Application Server Integrated Development Environment (IDE).

This *Tech Note* describes how to create and configure a derived template of the OPCClient DIOject using a C# console application.

Application Versions

To execute the GRAccess sample application that is described in this document, you will need the following prerequisites:

- Visual Studio 2005
- Industrial Application Server 2.1 or later
- DASABTCP DAServer or any other OPC Server

Using the IDE to Configure the OPCClient DIOject

Before describing using GRAccess to manipulate the OPCClient object, we'll describe the final goal by demonstrating first the manual steps using the Archestra IDE.

1. First create a derived template of the OPCClient DIOject.

2. Give the new derived template the name **\$OPCClient_02** and open the template editor.

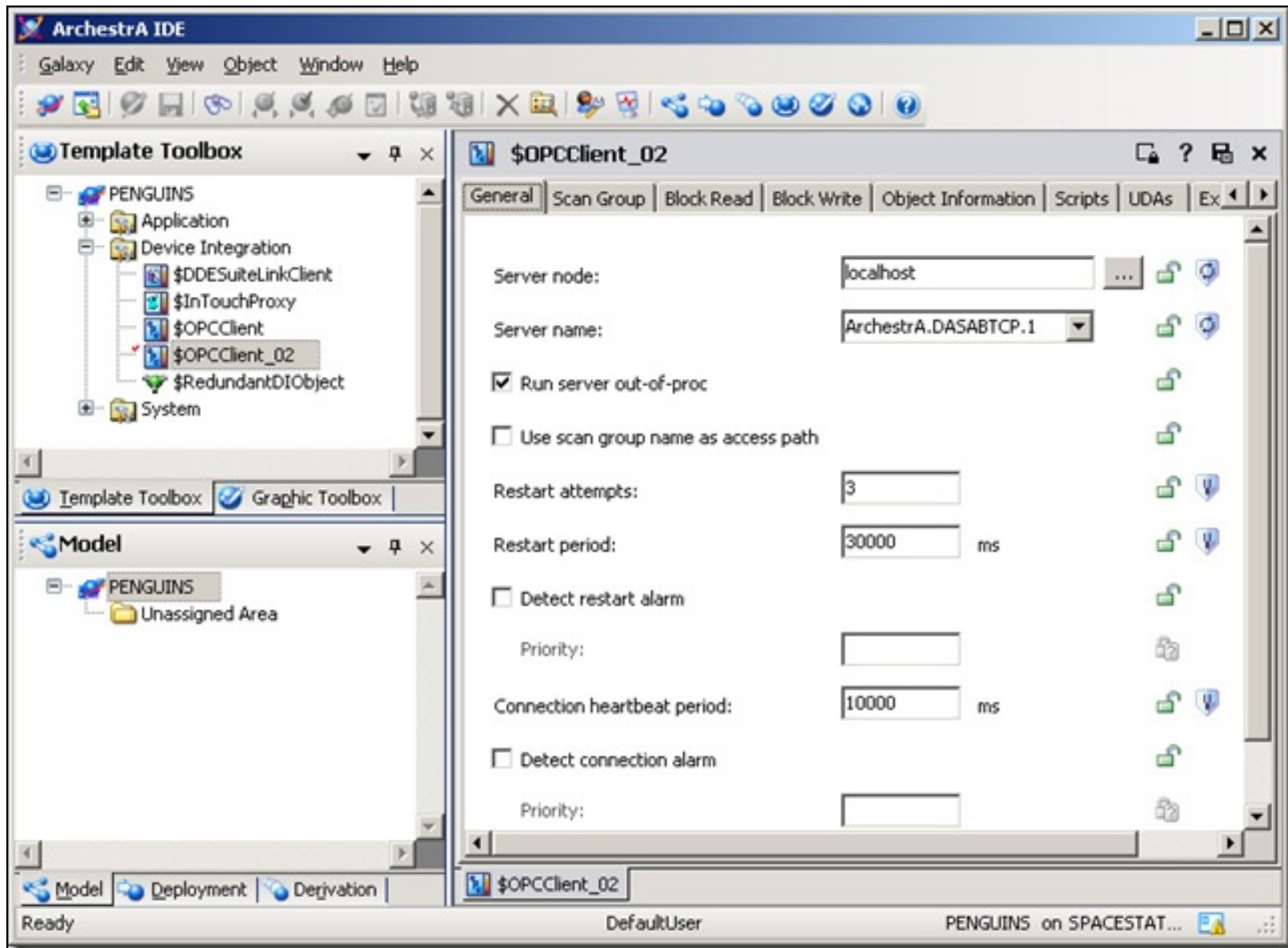


Figure 1: Editing the Derived Template

3. Click the **General** tab and set the Server node to **localhost**. Then select a registered OPC Server from the **Server name** list.
4. Click the **Scan Group** editor tab.

5. In the Scan Group tab add a new Scan Group called **Group1**. Then select **Group1**.
6. Add two Attributes and name them **Counter_1** and **Counter_2**. Assign the Item Reference **N100:1** to **Counter_1** and **N200:1** to **Counter_2**.

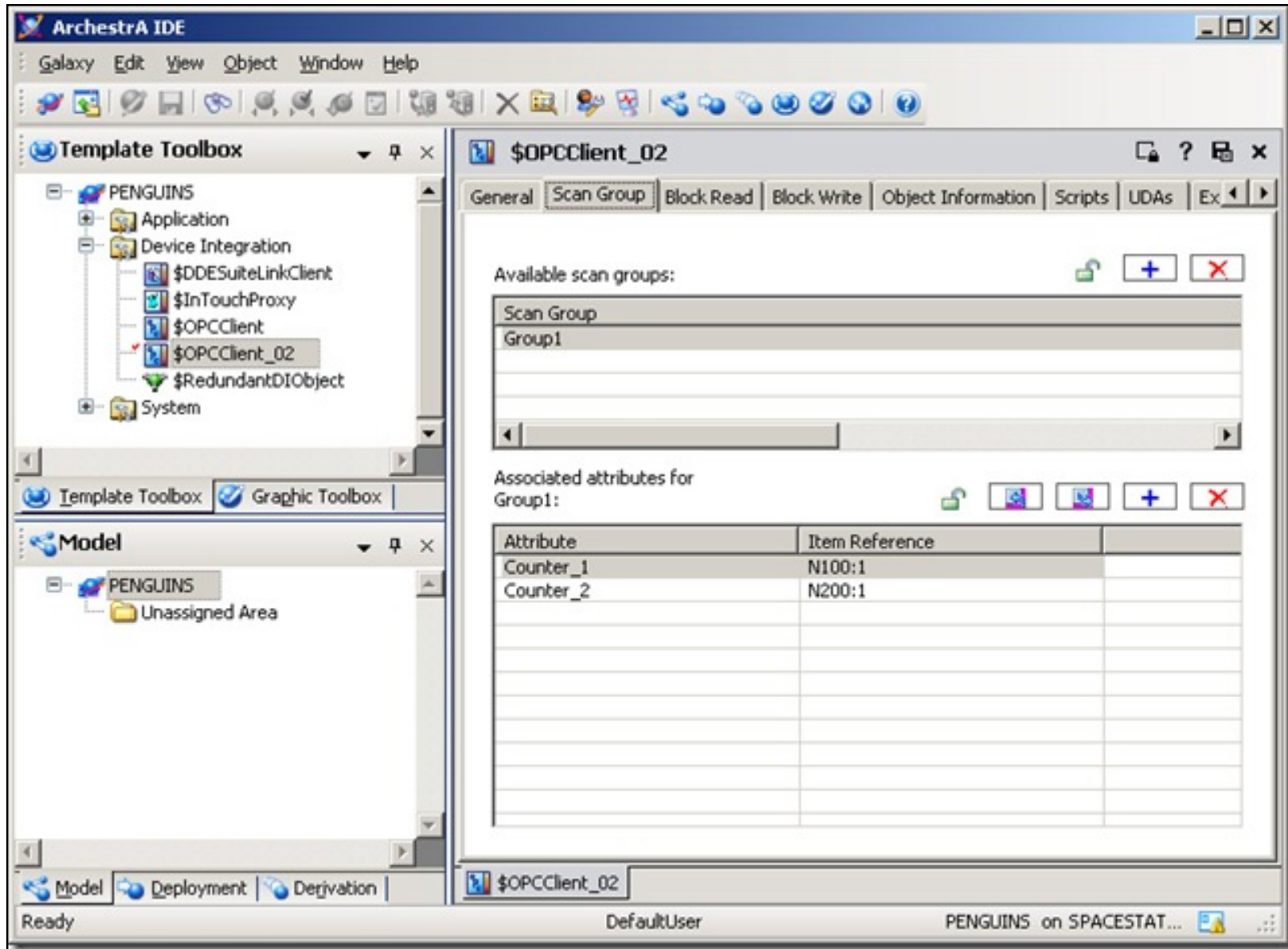


Figure 2: Configure Scan Groups

7. Save and close the object.

What Happens at the API Level

This section describes in GRAccess API terms what happened under the hood during the configuration of the OPCClient DIOject:

Configure the Common Attributes

Query the Galaxies and Login

```
Galaxies = GR.QueryGalaxies(GRMachineName);
G = Galaxies[GalaxyName];
G.Login(UserName, Password);
```

Get the \$OPCClient Template and Derive a New Template

```
string[] Names = { "$OPCClient" };
Objects = G.QueryObjectsByName(EgObjectIsTemplateOrInstance.gObjectIsTemplate, ref Names);
T = (ITemplate)Objects[1];
Parent = T.CreateTemplate(OPCClientTemplateName, true);
```

Checkout Template

```
Parent.CheckOut();
```

Configure ServerNode and ServerName Attributes

```
MxValue v = new MxValueClass();
//Configure computer name of OPCServer
v.PutString(OPCServerComputer);
Parent.Attributes["ServerNode"].SetValue(v);
//Configure OPC server name
v.PutString(OPCServerName);
Parent.Attributes["ServerName"].SetValue(v);
```

Add ScanGroup and an Attribute Array of Items

```
//Configure OPCGroup name
```

```

v.PutString(OPCGroupName);
Parent.Attributes["_ScanGroupAdd"].SetValue(v);
//Add list of alias names to Group1
a.Empty();
v.PutString(OPCItemAttributeName1);
a.PutElement(1, v);
v.PutString(OPCItemAttributeName2);
a.PutElement(2, v);
Parent.Attributes[OPCGroupName + "_ItemAdd"].SetValue(a);

```

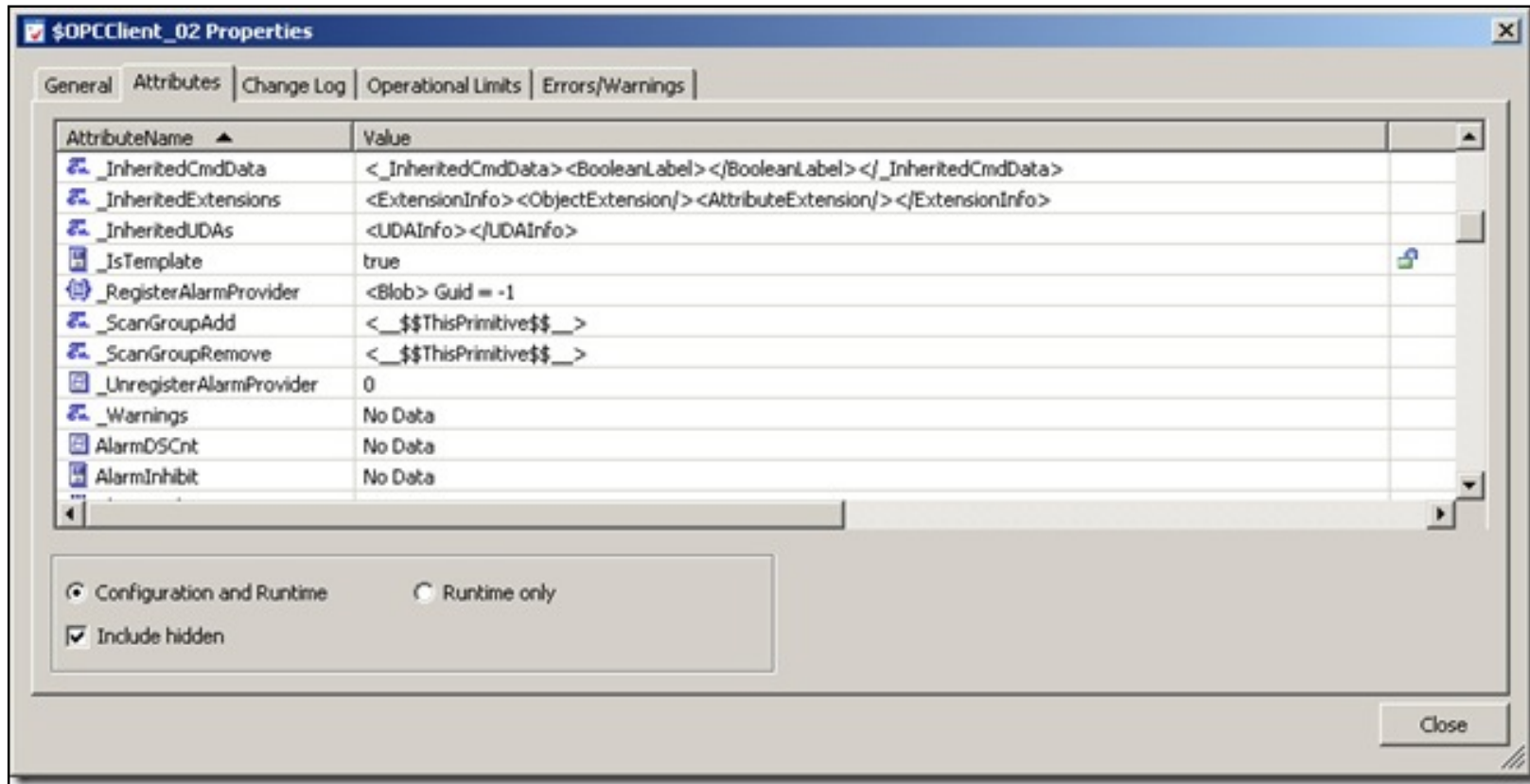


Figure 3: Object Properties

Save and CheckIn

```

Parent.Save();
Parent.CheckIn("");

```

The configuration of the item alias mapping table is the tricky part of this procedure, because the attribute name-to-item association is stored as XML string in an element called [ScanGroupName].AliasDatabase.

The Arcestra item attributes are XML elements in this XML file and the alias names are XML attribute of the item elements. To modify the mapping, the XML string needs to be retrieved from the attribute, the XML nodes need to be modified and then the modified XML string copied back into the original Arcestra attribute.

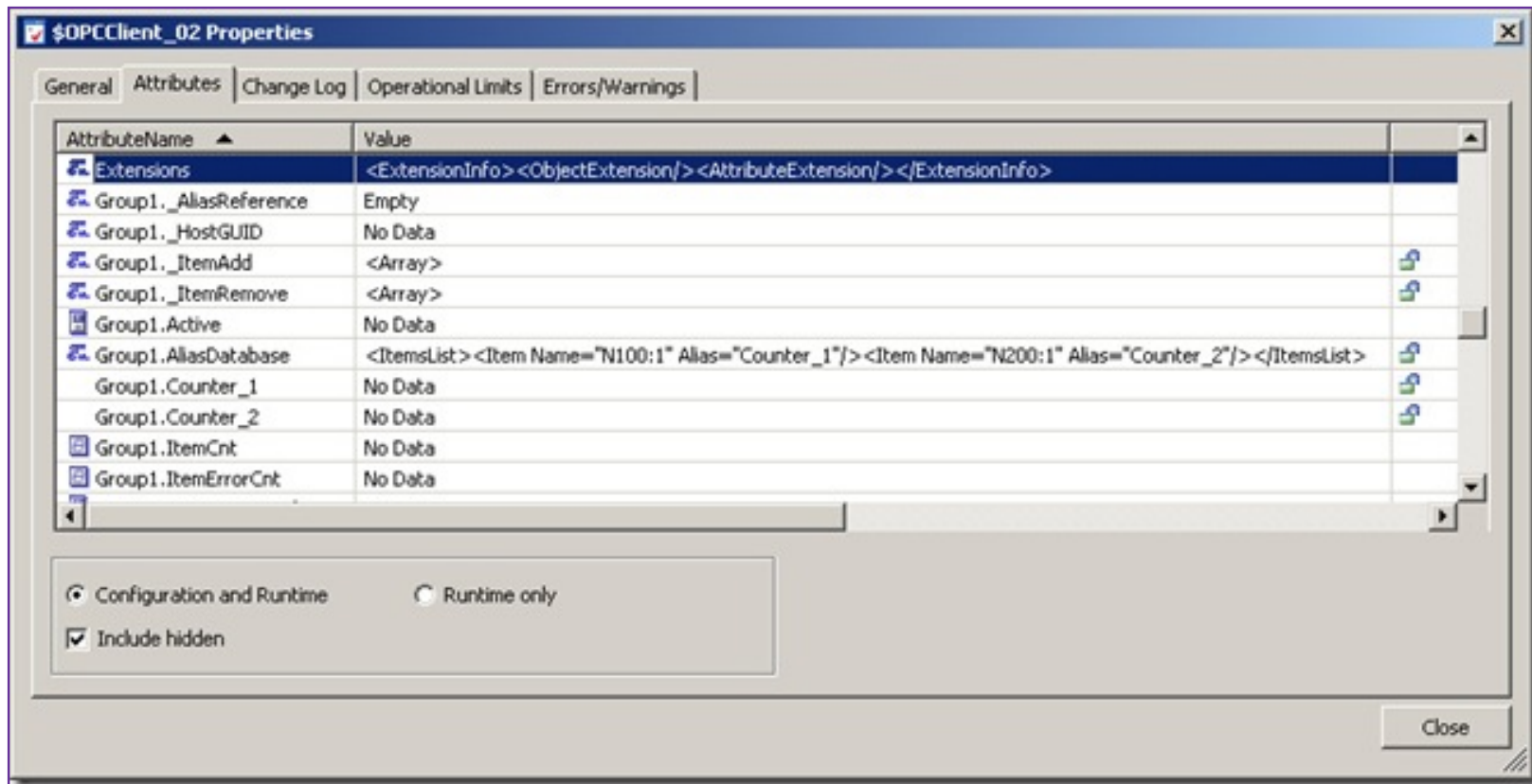


Figure 4: XML Modifications

Get Group1.AliasDatabase XML

```
v = Parent.Attributes[OPCGroupName + ".AliasDatabase"].value;  
string ItemDBXMLString = v.GetString();  
//Create an XML document instance, and load XML data from AliasDatabase attribute  
XmlDocument doc = new XmlDocument();  
System.IO.StringReader SR = new System.IO.StringReader(ItemDBXMLString);  
System.Xml.XmlTextReader TR = new System.Xml.XmlTextReader(SR);
```

```
doc.Load(TR);
```

Use XPath to Modify Item Names in XML

```
//XPath query to the actual items
XmlNodeList nodeList = doc.SelectNodes("/ItemsList/Item");
//Modify items list to point to the actual item names
XmlNode Item1 = nodeList[0];
Item1.Attributes["Name"].Value = OPCItemName1;
XmlNode Item2 = nodeList[1];
Item2.Attributes["Name"].Value = OPCItemName2;
string NewItemDBXMLString = doc.OuterXml;
```

Checkout Template

```
Parent.CheckOut();
```

Change Group1.AliasDatabase attribute with new XML

```
v.Empty();
v.PutString(NewItemDBXMLString);
Parent.Attributes[OPCGroupName + ".AliasDatabase"].SetValue(v);
```

Save and CheckIn

```
Parent.Save();
Parent.CheckIn("");
```

The Complete Source File

[Download the complete program.cs file.](#)

Compilation

To compile the program.cs file run the following command in the Visual Studio 2005 command prompt:

```
csc /out:CreateOPCCClient.exe program.cs /r:"C:\Program Files\Common Files\ArchestrA\ArchestrA.GRAccess.
```

```
dll";
```

Test Run

To execute **CreateOPCClient.exe** run the following command. In the following command line PENGUINS is the Galaxy name. Change it to reflect your server name:

```
CreateOPCClient gr=localhost g=PENGUINS u=Administrator pw=ww on=$OPCClient_02 os=ArchestrA.DASABTCP.1  
oc=localhost og=Group1 oil=N100:1 oa1=Counter_1 oi2=N200:1 oa2=Counter_2
```

Summary

The key to configuring an OPCClient DIObject using the GRAccess toolkit is the knowledge that the alias item names are stored in a single attribute as XML string. Modifying this string will create the mappings as seen in the OPCClient object's Scangroup editor.

K. Graefensteiner

Tech Notes are published occasionally by Wonderware Technical Support. Publisher: Invensys Systems, Inc., 26561 Rancho Parkway South, Lake Forest, CA 92630. There is also technical information on our software products at www.wonderware.com/support/mmi

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



[Back to top](#)

©2008 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 any information storage and retrieval system, without permission in writing from Invensys Systems, Inc. [Terms of Use](#).