Tech Note 962
Automating Object Configuration Tasks in the GRAccess Toolkit: Configuring UDA Array

Topic#: 002782
Created: April 2013

Introduction

The GRAccess Toolkit enables automating activities that engineers normally perform manually using the Industrial Application Server Integrated Development Environment (IDE).

This *Tech Note* describes configuring UDA Array and assigning values to UDA Array programmatically using a C# application.

For this *Tech Note,* a template called **$UDArrayTest** has been created. It is derived from the **$UserDefined** template.

The Following tasks are done programmatically in the newly created **$UDArrayTest** template object.

1. **Configure the UDA Array**
2. **Modify the Value of Only the 3rd Element of the Array**

Application Versions

To execute the GRAccess code snippet that is described in this document, you will need the following :

- Visual Studio 2005, C#
- Industrial Application Server 3.0 and later
- GRAccess 3.0

Before executing the code snippet in item#1, the ArchestrA IDE configuration looks like following:

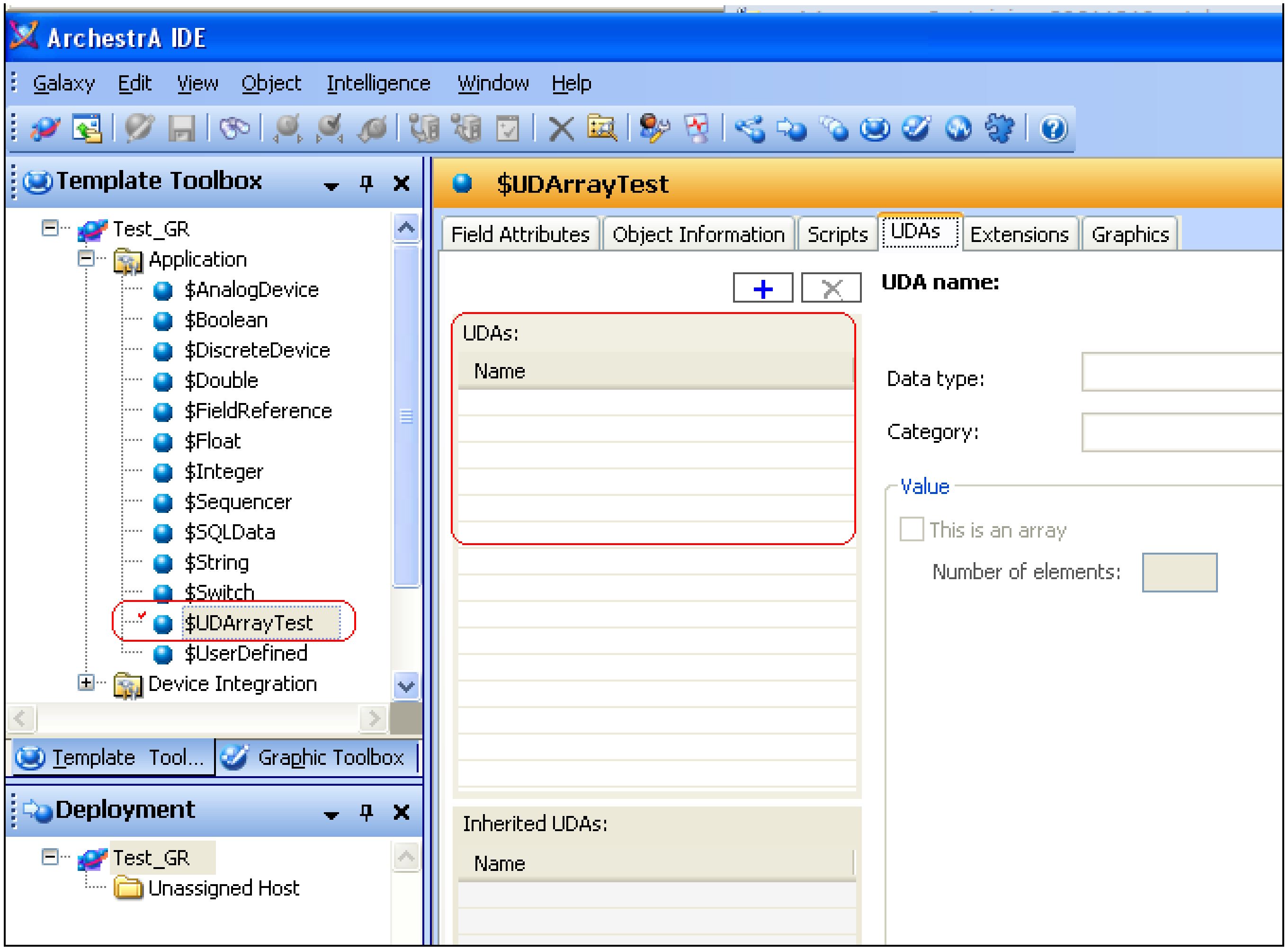


Figure 1: UDA for $UDArrayTest Object

Configure the UDA Array

Configure the UDAarray with 5 integers, and assign values to each element of the array.

Code Snippet in C#

Declarations:

```
ICommandResult CR;
ITemplate UDATest;
MxValue MXVal = new MxValueClass();
MxValue MXValArray = new MxValueClass();
```

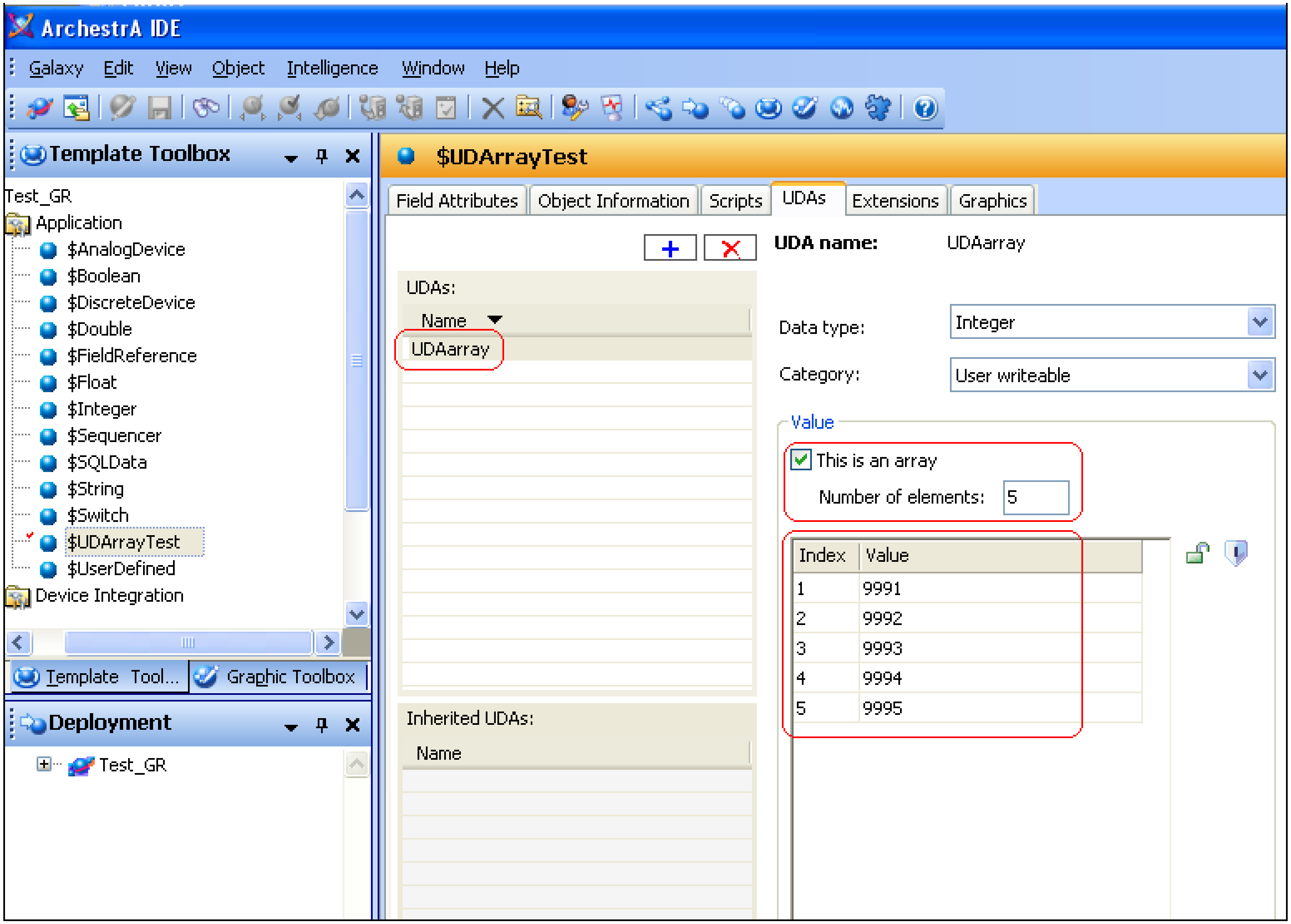
Code Snippet:

```
//Checkout the Template Object
UDATest.CheckOut();
UDATest.AddUDA("UDAarray", MxDataType.MxInteger,
   MxAttributeCategory.MxCategoryWriteable_USC_Lockable,
   MxSecurityClassification.MxSecurityFreeAccess, true, 5);
CR = UDATest.CommandResult;
if (!CR.Successful)
{
   MessageBox.Show("Adding UDA UDAarray to $UDArrayTest failed:"
      + CR.Text + " : "
      + CR.CustomMessage);
   UDATest.CheckIn("");

}
else
{
   UDATest.Save();
   //Assign Values to all 5 elements of Array
   MXVal.PutInteger(9991);
   MXValArray.PutElement(1,MXVal);
   MXVal.PutInteger(9992);
   MXValArray.PutElement(2,MXVal);
   MXVal.PutInteger(9993);
   MXValArray.PutElement(3,MXVal);
   MXVal.PutInteger(9994);
   MXValArray.PutElement(4,MXVal);
   MXVal.PutInteger(9995);
   MXValArray.PutElement(5,MXVal);
UDATest.ConfigurableAttributes["UDAarray"].SetValue(MXValArray);
   UDATest.Save();
   UDATest.CheckIn("");
}
```

Notice that there is no datatype set specifically for the array in the above snippet. The array gets the datatype of the very first element set. In this example, **MXVal.PutInteger(9991);** determines that the array will be of **Integer** type.

Figure 2 (below) shows the ArchestrA IDE after executing the above code snippet.



UDA name: UDAarray

Data type: Integer

Category: User writeable

☑ This is an array

Number of elements: 5

Index	Value
1	9991
2	9992
3	9993
4	9994
5	9995

Figure 2: UDAarray Modifications

Modify the Value of Only the 3rd Element of the Array

Code snippet in C#

Declarations:

```
ICommandResult CR;
ITemplate UDATest;
MxValue MXVal = new MxValueClass();
MxValue MXValArray = new MxValueClass();
```

Code Snippet:

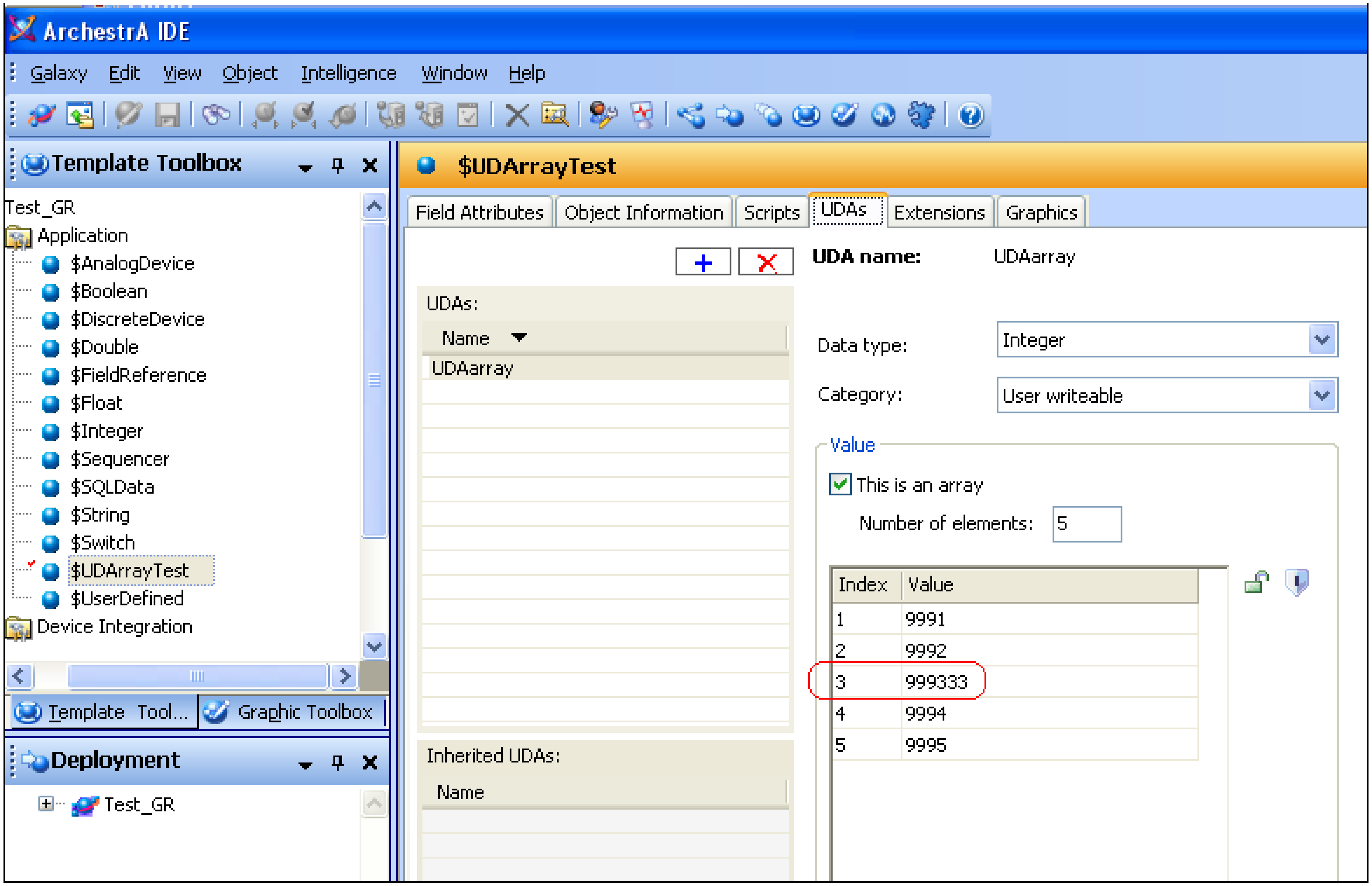
```
//Change value of only element#3 in the array
UDATest.UpdateUDA("UDAarray", MxDataType.MxInteger,
   MxAttributeCategory.MxCategoryWriteable_USC_Lockable,
   MxSecurityClassification.MxSecurityFreeAccess,true,10);
CR = UDATest.CommandResult;
if (!CR.Successful)
{
   MessageBox.Show("Adding UDA UDAarray to $UDArrayTest failed:"
      + CR.Text + " : " +
      CR.CustomMessage);
   UDATest.CheckIn("");

}
else
{
   UDATest.Save();
   MXVal.PutInteger(999333);
   MXValArray.PutElement(3, MXVal);

UDATest.ConfigurableAttributes["UDAarray"].SetValue(MXValArray);
   UDATest.Save();
   UDATest.CheckIn("");

}
```

Figure 3 (below) shows the ArchestrA IDE after executing the above code snippet.



UDA name: UDAarray

Data type: Integer

Category: User writeable

☑ This is an array

Number of elements: 5

Index	Value
1	9991
2	9992
3	999333
4	9994
5	9995

Figure 3: IDE After Running the Code

P. Kulkarni

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 **Wonderware Technical Support.**

For technical support questions, send an e-mail to **wwsupport@invensys.com**.

Back to top