Tech Note 1004

Accessing InBatch Materials Inside of System Platform using Material Automation Server (Object Hosted to Remote Node)

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#: 002835
Created: January 2014

Introduction

This Tech Note demonstrates the syntax used for the Material Object creation/release in a .NET FrameWork environment.

The key is to register the COM object and also specify where material automation server running. In this scenario, object containing script for accessing InBatch Materials Data hosted to remote platform and InBatch Server is running on a separate node. The other functionality on reading/writing to InBatch Material Database is documented in InBatch COM Guide.

Application Versions

- InBatch 9.x & 10.x
- Wonderware Application Server 2012 R2 (for prior versions, you have to manually built the object)

Prerequisites

- InBatch development client software installed on a remote computer.
- Platform deployed to the InBatch Development Client node where the Object is hosted to access InBatch Material Data.

Procedure

1. From the Archestra IDE menu, click **Galaxy\Import\Script Function Library** and import the **MaterialSrv.exe** file. Browse to **C:\Program files(x86)\Wonderware\inBatch\Bin** on the InBatch Server Node.
2. Download and import the **InBatchMaterialAccess2** template **aaPkg** file to IDE.
3. Double-click the **$InBatchMaterialAccess2** object in the IDE.
4. Click the **Scripts** tab in the Object editor and set **me.BatchServerHostName** to the name of the Batch Server. In Figure 1 (below) it is **"SMInBatch10"**.
5. Create an instance of this object and host it to the Platform\AppEngine\Area that is running on the remote node where Batch Development Client Software is installed.

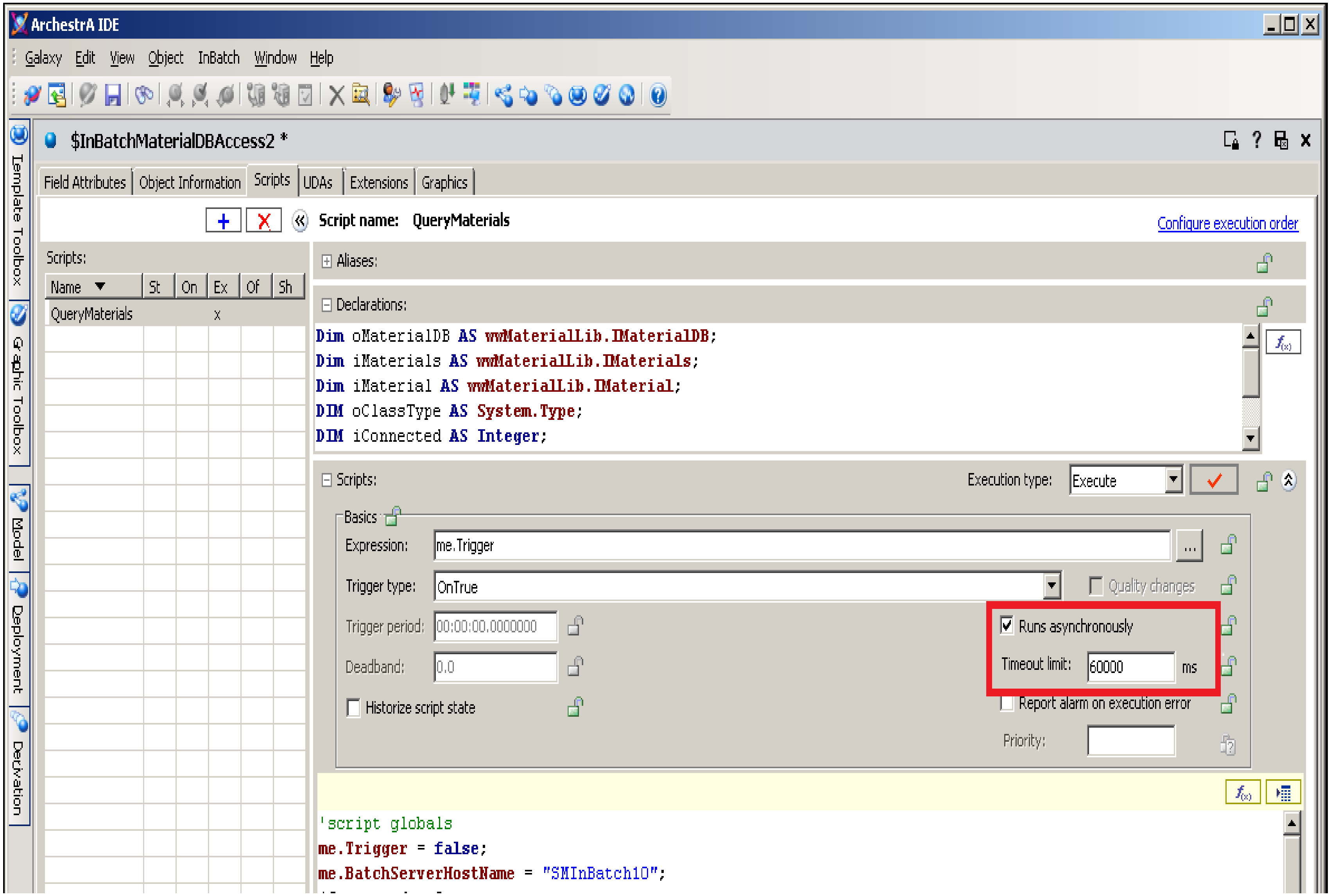
6. Set DCOM configuration settings pointing to InBatch Server Node per InBatch COM users guide, Chapter 4.

Note: If you are using a 64-bit operating system on the remote node, you have to launch DCOMCnfg window using following command. This command sets for 32-bit: **C:\Windows\SysWOW64>mmc comexp.msc /32**

5. Deploy the newly-created instance object and set the trigger attribute (me.Trigger) in ObjectViewer for this object.

6. Review the messages in the Archestra Logger for successful connectivity, query of materials and release of the material object.

Figure 1 (below) shows the screen snapshots of the template object editor, text in the script body and Archestra Logger Messages. In addition, Note2 talks about setting InBatch Server Host Name accordingly in a batch redundancy environment.



Script name: QueryMaterials

QueryMaterials x

Declarations:

```
Dim oMaterialDB AS wwMaterialLib.IMaterialDB;
Dim iMaterials AS wwMaterialLib.IMaterials;
Dim iMaterial AS wwMaterialLib.IMaterial;
DIM oClassType AS System.Type;
DIM iConnected AS Integer;
```

Execution type: Execute

Expression: me.Trigger

Trigger type: OnTrue

☐ Quality changes

Trigger period: 00:00:00.0000000

Deadband: 0.0

☐ Historize script state

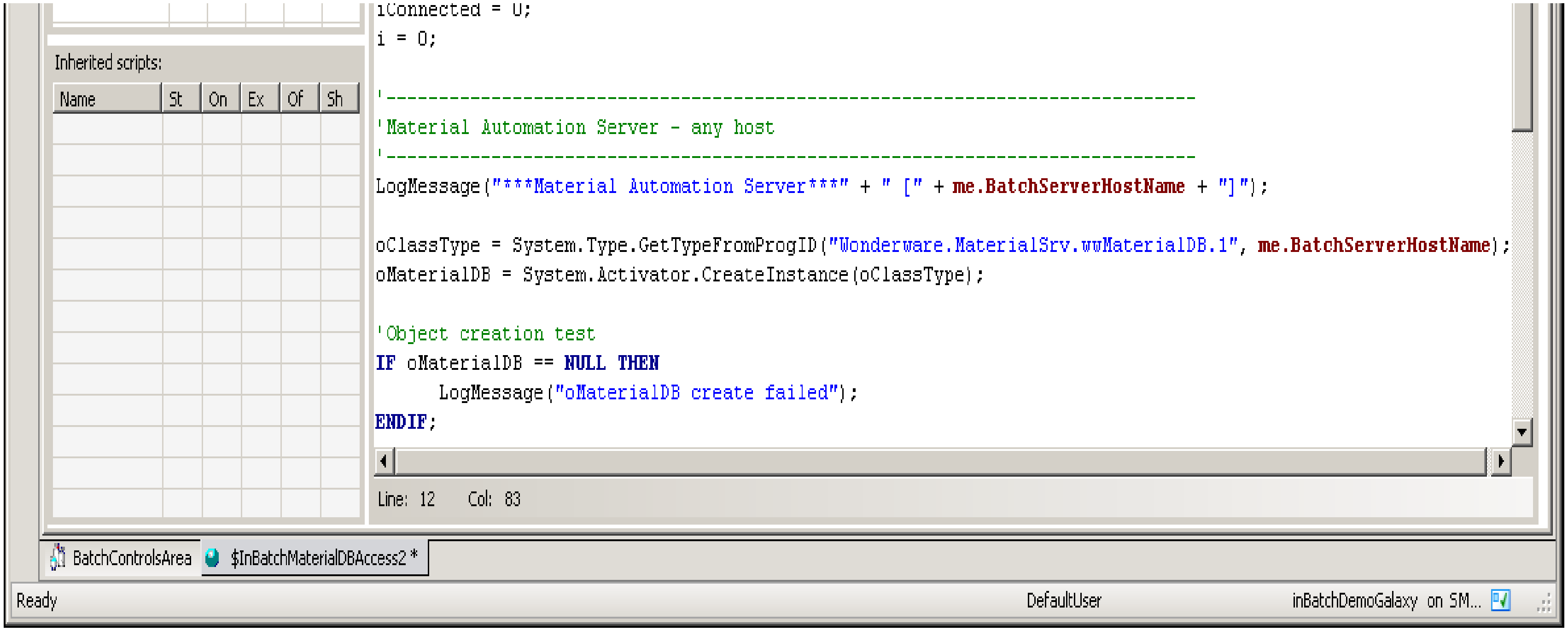
☑ Runs asynchronously

Timeout limit: 60000 ms

☐ Report alarm on execution error

Priority:

```
'script globals
me.Trigger = false;
me.BatchServerHostName = "SMInBatch10";
```



```
iConnected = 0;
i = 0;

'-----------------------------------------------------------------------------
'Material Automation Server - any host
'-----------------------------------------------------------------------------
LogMessage("***Material Automation Server***" + " [" + me.BatchServerHostName + "]");

oClassType = System.Type.GetTypeFromProgID("Wonderware.MaterialSrv.wwMaterialDB.1", me.BatchServerHostName);
oMaterialDB = System.Activator.CreateInstance(oClassType);

'Object creation test
IF oMaterialDB == NULL THEN
      LogMessage("oMaterialDB create failed");
ENDIF;
```

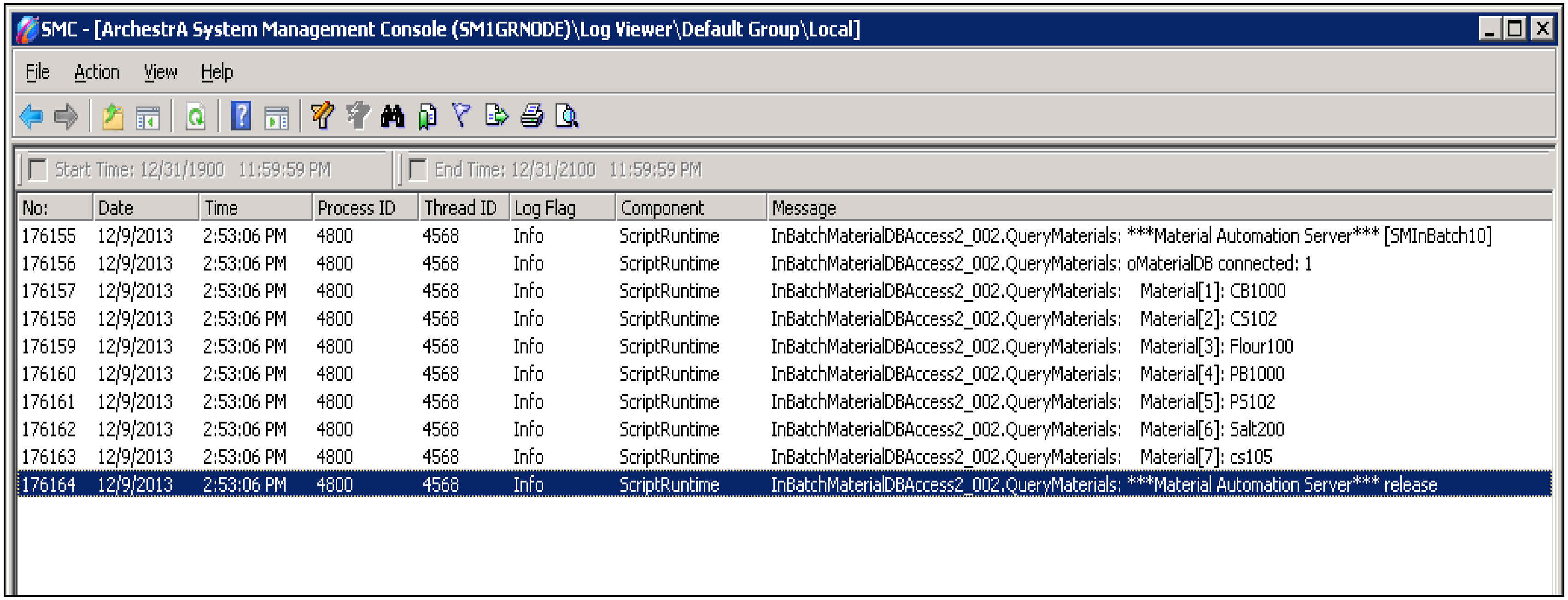
Figure 1: Ensure the Script Runs Asynchronously

Text in script body (if you using prior version of System Platform 2012 R2, copy the following text and paste it to your object, declare variables in declaration section as shown in Figure 1.

```
'script globals
me.Trigger = false;
me.BatchServerHostName = "BatchServerNodeName";
iConnected = 0;
i = 0;
'-----------------------------------------------------------------------------------------
'Material Automation Server - any host
'-----------------------------------------------------------------------------------------
LogMessage("***Material Automation Server***" + " [" + me.BatchServerHostName + "]");
oClassType = System.Type.GetTypeFromProgID("Wonderware.MaterialSrv.wwMaterialDB.1", me.BatchServerHostName);
oMaterialDB = System.Activator.CreateInstance(oClassType);

'Object creation test
IF oMaterialDB == NULL THEN
LogMessage("oMaterialDB create failed");
ENDIF;

'Connection test
iConnected = oMaterialDB.TestConnection();
IF iConnected <> 1 Then
LogMessage("oMaterialDB not connected: " + iConnected);
Else LogMessage("oMaterialDB connected: " + iConnected);
ENDIF;

'Sample script on Query Materials Ids
oMaterialDB.QueryMaterials();
iMaterials = oMaterialDB.Materials;
FOR EACH iMaterial in iMaterials
i = i + 1;
LogMessage(" Material[" + i + "]: " + iMaterial.Id);
NEXT;

LogMessage("***Material Automation Server*** release");
'note - despite release, material server does not shutdown but it is reused
if oMaterialDB <> NULL Then
System.Runtime.InteropServices.Marshal.ReleaseComObject( oMaterialDB );
```

```
        oMaterialDB = NULL;
        ENDIF;
```

Archestra Logger Messages


SMC - [ArchestrA System Management Console (SM1GRNODE)\Log Viewer\Default Group\Local]
File
Action
View
Help
Start Time: 12/31/1900 11:59:59 PM
End Time: 12/31/2100 11:59:59 PM

```
No:     Date       Time        Process ID  Thread ID  Log Flag  Component      Message
176155  12/9/2013  2:53:06 PM  4800        4568       Info      ScriptRuntime  InBatchMaterialDBAccess2_002.QueryMaterials: ***Material Automation Server*** [SMInBatch10]
176156  12/9/2013  2:53:06 PM  4800        4568       Info      ScriptRuntime  InBatchMaterialDBAccess2_002.QueryMaterials: oMaterialDB connected: 1
176157  12/9/2013  2:53:06 PM  4800        4568       Info      ScriptRuntime  InBatchMaterialDBAccess2_002.QueryMaterials:   Material[1]: CB1000
176158  12/9/2013  2:53:06 PM  4800        4568       Info      ScriptRuntime  InBatchMaterialDBAccess2_002.QueryMaterials:   Material[2]: CS102
176159  12/9/2013  2:53:06 PM  4800        4568       Info      ScriptRuntime  InBatchMaterialDBAccess2_002.QueryMaterials:   Material[3]: Flour100
176160  12/9/2013  2:53:06 PM  4800        4568       Info      ScriptRuntime  InBatchMaterialDBAccess2_002.QueryMaterials:   Material[4]: PB1000
176161  12/9/2013  2:53:06 PM  4800        4568       Info      ScriptRuntime  InBatchMaterialDBAccess2_002.QueryMaterials:   Material[5]: PS102
176162  12/9/2013  2:53:06 PM  4800        4568       Info      ScriptRuntime  InBatchMaterialDBAccess2_002.QueryMaterials:   Material[6]: Salt200
176163  12/9/2013  2:53:06 PM  4800        4568       Info      ScriptRuntime  InBatchMaterialDBAccess2_002.QueryMaterials:   Material[7]: cs105
176164  12/9/2013  2:53:06 PM  4800        4568       Info      ScriptRuntime  InBatchMaterialDBAccess2_002.QueryMaterials: ***Material Automation Server*** release
```

FIGURE 2: ARCHESTRA LOGGER MESSAGES

Note: If you are using InBatch Redundancy, you can set the attribute **me.BatchServerHostName** accordingly by monitoring the status of a Boolean Tag **IBSERV_RED_MASTER**. This tag is provided by the InBatch Server Application.

The value of IBSERV_RED_MASTER is **True** on the Master Node and **False** on the Backup Node. You can configure two instances of **DDESuiteLink** objects and configure them as shown below and deploy these two instances to the node where the Platform is deployed, and where the InBatch Client Software is installed.

If you monitor the value **DDESuiteLink1.IB_TAGS.IBSERV_RED_MASTER** and **DDESuiteLink2.IB_TAGS.IBSERV_RED_MASTER** in ObjectViewer, you will see the values as **True** and **False**. You can use these two attributes in the same script as shown above in script body or in the template object and set the Batchserver HOSTNAME accordingly to point to InBatch Master node.

This scenario is included here to monitor the InBatch Redundancy Status and was not tested for this *Tech Note*.

	ServerName	TopicName	Item	NodeName
DDESuiteLink1	IBSERV	IB_TAGS	IBSERV_RED_MASTER	InBatchMaster
DDESuiteLink2	IBSERV	IB_TAGS	IBSERV_RED_MASTER	InBatchBackup

S. Mariyala, M. Mummert

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