How to trace OPC writes in the ABB 800xA system

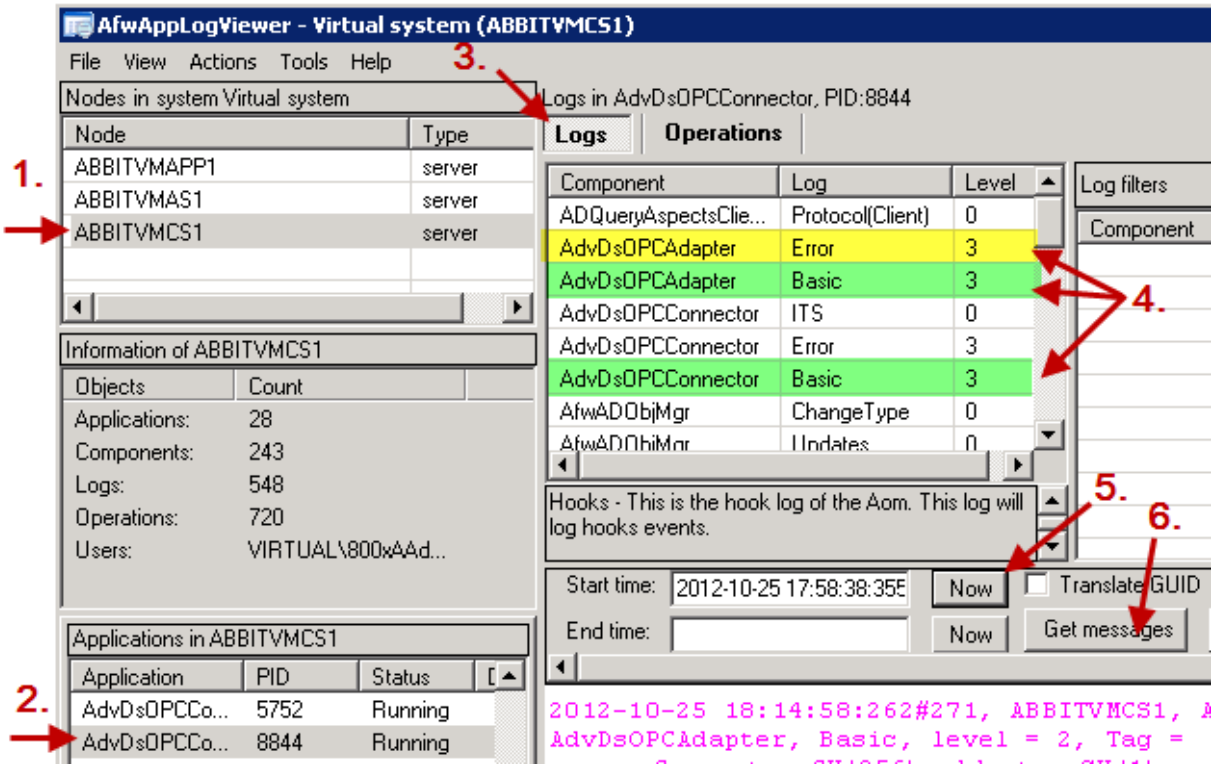
Stefan Strömqvist, ABB Automation Technologies

Login to any node in the system using an account member of the IndustrialITAdmin User Group in Windows.

Start the AppLog tool using Start → Run… `afwapplogviewer`

Click **OK** in the two following dialog boxes (accepting default values).

1. In the (upper left) list of nodes select the Connectivity Server node in which OPC writes are to be traced.
2. In the (lower left) list of application (processes) select the AdvDsOPCConnector process corresponding to the connectivity/OPC server being probed. In some cases multiple Connector processes exist in the same server node. In such cases, use the Plant Explorer and Service Structure to identify the PID of the process that is of interest.
3. Select **Logs**
4. Enable the following logs to level 3 (the highest log level)
    - **AdvDsOPCAdapter Basic**
    - **AdvDsOPCAdapter Error** *(once ItemIDs have been recorded, this log is enough to enable)*
    - **AdvDsOPCConnector Basic**
5. Create a time range filter (often it is desired to see new messages ignoring old, then just press the upper **Now** button to set current time as Start time.
6. Press **Get messages** button



To be able to properly identify ItemIDs, the logs must be running before the writing OPC client is started.

On the following pages I have captured log output from the Property Transfer service having one single item written every 10th second (some log output has been removed for better clarity).

2012-10-25 17:59:27:269#171, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 8032
AdvDsOPCConnector, Basic, level = 1, Tag =
CConnectorClient::Init

2012-10-25 17:59:27:269#172, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 8032
AdvDsOPCConnector, Basic, level = 1, Tag =
CConnectorClient::ConnectionEstablished
→ A new OPC client has connected to the OPC Connector service we're logging in!

2012-10-25 17:59:27:281#175, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 4056
AdvDsOPCAdapter, Basic, level = 1, Tag =
CAdvDsOPCServerAdapter::AddItems

2012-10-25 17:59:27:336#178, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 4056
AdvDsOPCAdapter, Basic, level = 2, Tag =
AddGroup: group(5,1) updateRate(10000 ms) environment {F6590D3D-7BB4-4430-A21E-62C1F2A1518A}
→ The client has added an OPC group.
The OPC Adapter group ID is 5,1 which actually corresponds to two different figures, a Client Handle (5) and Server Handle (1).

2012-10-25 17:59:27:343#179, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 4056
AdvDsOPCAdapter, Basic, level = 2, Tag =
AddItems: 1 items, group (5,1)

2012-10-25 17:59:27:343#180, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 4056
AdvDsOPCAdapter, Basic, level = 3, Tag =
Applications.Application_1.VALUE(VT_I4): active(0), item(3,1), blob(0/0)
→ The client has added an OPC item to the newly created group and some important IDs are passed back and forth:

ItemID:	**Applications.Application_1.VALUE**
OPC Adapter Client Handle	**3**
OPC Adapter Server Handle	**1**

2012-10-25 17:59:27:350#181, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 8032
AdvDsOPCConnector, Basic, level = 1, Tag =
CConnectorClient::WriteRequestHandler reqId(3)

2012-10-25 17:59:27:350#182, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 4056
AdvDsOPCAdapter, Basic, level = 1, Tag =
CAdvDsOPCServerAdapter::AsyncWrite

2012-10-25 17:59:27:350#183, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 4056
AdvDsOPCAdapter, Basic, level = 3, Tag =
CAdvDsOPCSAAsyncHandlerOPC20::Write
→ The client is requesting an Asynchronous OPC DA version 2.0 write

2012-10-25 17:59:27:351#184, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 4056
AdvDsOPCAdapter, Basic, level = 2, Tag =
Write: 1 items, group(5,1), tID(0)

→ OPC Adapter group ID is **5,1**. Write Transaction ID is **0**.

2012-10-25 17:59:27:351#185, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 4056
AdvDsOPCAdapter, Basic, level = 3, Tag =
item hS(1), value(VT_I4) = -1

→ OPC Adapter Server Handle = **1** (with this information you can now identify the actual ItemID).
The value to write is "-1".

2012-10-25 17:59:27:427#186, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 9864
AdvDsOPCAdapter, Basic, level = 2, Tag =
CAdvDsOPCSAAsyncHandlerBase::OnWriteComplete

2012-10-25 17:59:27:427#187, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 9864
AdvDsOPCAdapter, Basic, level = 1, Tag =
CAdvDsOPCServerAdapter::OnWriteComplete - transactionId = 0, count = 1

→The OPC Server has returned a "OnWriteComplete" for Write Transaction ID **0**.

2012-10-25 17:59:27:427#188, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 9864
AdvDsOPCAdapter, Basic, level = 2, Tag =
write: item(3), hr(0)

→ The OPC Adapter reports that the write was completed for OPC Adapter Client Handle **= 3** with HRESULT =**0** (S_OK) which is a success code.

2012-10-25 17:59:27:427#189, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 8032
AdvDsOPCConnector, Basic, level = 1, Tag =
CConnector::WriteCompleted, transactionID = 18, count = 1

2012-10-25 17:59:27:428#190, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 8032
AdvDsOPCConnector, Basic, level = 2, Tag =
CConnector::WriteCompleted - Item: hC(256), hS(513), hr(0)

→ The OPC Connector is reporting the same as the OPC Adapter just did above, but it is using another set of Client/Server Handles (that so far has not appeared in this trace). But we need to remember them to be able to perform a full decode later on.
OPC Connector Client Handle **256**
OPC Connector Server Handle **513**

2012-10-25 17:59:37:353#204, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 4056
AdvDsOPCAdapter, Basic, level = 1, Tag =
CAdvDsOPCServerAdapter::AsyncWrite

2012-10-25 17:59:37:353#205, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 4056
AdvDsOPCAdapter, Basic, level = 3, Tag =
CAdvDsOPCSAAsyncHandlerOPC20::Write

→ Ten (10) seconds have passed and the Property Transfer is triggered by a new value from the source OPC server, a new write is on its way down to the destination OPC server…

2012-10-25 17:59:37:356#206, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 4056
AdvDsOPCAdapter, Basic, level = 2, Tag =
Write: 1 items, group(5,1), tID(1)
→Same as last write, but the Write Transaction ID has been incremented by one.

2012-10-25 17:59:37:356#207, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 4056
AdvDsOPCAdapter, Basic, level = 3, Tag =
item hS(1), value(VT_I4) = 75
→Value to write to OPC Adapter Server Handle **1** is now "75".

2012-10-25 17:59:37:362#208, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 9864'
AdvDsOPCAdapter, Basic, level = 2, Tag =
CAdvDsOPCSAAsyncHandlerBase::OnWriteComplete

2012-10-25 17:59:37:362#209, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 9864
AdvDsOPCAdapter, Basic, level = 1, Tag =
CAdvDsOPCServerAdapter::OnWriteComplete - transactionId = 1, count = 1
→The OPC Server has returned a "OnWriteComplete" for Write Transaction ID **1**.

2012-10-25 17:59:37:362#210, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 9864
AdvDsOPCAdapter, Basic, level = 2, Tag =
write: item(3), hr(c0040007)
→ This time the OPC Adapter reports a write failure for OPC Adapter Client handle **3**
(I happened to disconnect the controller from the network… ☺)
c0040007 = OPC_E_UNKNOWNITEMID "The item is no longer available in the server address space."

2012-10-25 17:59:37:362#211, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 9864
AdvDsOPCAdapter, Error, level = 1, Tag =
failed to write item hC(513).: error = 0xc0040007
→ This is the so far only output from the OPC Adapter Error log we've seen. We can talk a lot about confusing IDs, etc. but this time the OPC Adapter is reporting the OPC Connector Server Handle **513**.

2012-10-25 17:59:37:363#212, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 8032
AdvDsOPCConnector, Basic, level = 1, Tag =
CConnector::WriteCompleted, transactionID = 19, count = 1

2012-10-25 17:59:37:363#213, ABBITVMCS1, AdvDsOPCConnector, PID: 4364, Thread: 8032
AdvDsOPCConnector, Basic, level = 2, Tag =
CConnector::WriteCompleted - Item: hC(256), hS(513), hr(c0040007)
→ Again, the same info but seen from the OPC Connector's point of view and IDs.

What do we learn from this???

- To be able to properly decode the logs we must have a record of when the OPC client added the groups and items (i.e. the very first moments of its "lifetime")

- With all IDs known, the OPC Adapter Error log (in red above) alone can tell us what writes goes wrong (and suppress all successful writes)