

Invensys



Foxboro I/A Series OPC Solutions

Installation, Configuration, Troubleshooting

Maintained by Ron.Deen@Invensys.com and Sven.Audenaert@Invensys.com

Revision 1.7

April 15, 2005

This document is © Copyright 2002-2004 Invensys.

Foxboro, I/A Series are trademarks of Invensys, its subsidiaries and affiliates.

All other brand names may be trademarks of their respective owners.

Contents

1	Introduction.....	7
1.1	Disclaimer.....	7
1.2	Conventions.....	7
1.3	Revision history.....	8
1.4	Related documents.....	8
1.5	Introducing OPC.....	9
1.6	A Two Box Approach.....	9
2	The Foxboro OPC I/O Gate client (to Foxboro AIMOPC Server).....	11
2.1	Requirements.....	11
2.1.1	General hardware and system software.....	12
2.1.2	OPC I/O Gate licensing.....	12
2.2	Installation.....	12
2.2.1	Installing the OPC I/O Gate.....	12
2.2.2	DCOM and Server side configuration.....	13
2.2.3	Registering the OPC Server.....	14
2.2.3.1	On the AW70 machine: Registering the Server.....	14
2.2.3.2	Loading the OPC Server reg key on the OPC client host.....	16
2.2.4	Creating the Foxboro account on the Server.....	16
2.2.5	DCOM Settings.....	17
2.2.5.1	On the OPC Server side.....	17
2.2.5.2	On the OPC Client side.....	19
2.2.6	Alias files.....	19
2.2.7	Start OPC I/O Gate at system startup.....	21
2.2.8	OPC I/O Gate Log files.....	21
2.3	Creating a Control Database.....	22
2.3.1	The Server ECB (ECB99).....	23
2.3.2	Creating the Group ECB (ECB100).....	24
3	The Foxboro AIM*OPC Server on Windows NT.....	31
3.1	Requirements.....	31
3.1.1	AIM*OPC Server licensing.....	31
3.2	Installation instructions.....	31
3.2.1	Installing the AIM*AT Utilities.....	31
3.3	Initial API Admin configuration.....	32
3.3.1	New AIMREG.DAT file.....	32
4	The Foxboro AIM*OPC Server 3.2 on Windows XP.....	37
4.1	Requirements.....	37
4.2	Installation instructions.....	37
4.2.1	Installing the AIM*AT Utilities.....	37
4.3	Initial API Admin configuration.....	38
4.4	Package licensing.....	39
5	AIM*Historian and the AIM OPC I/O Gate client.....	43
5.1	Setting up the AIM*Historian 3.2.....	43

6	Foxboro OPC I/O Gate (to Emerson PROVOX OPC Server).....	45
6.1	Introduction.....	45
6.2	Prerequisites.....	45
6.2.1	Actions on the Emerson OPC CHIPNT Server:.....	45
6.3	Installing the Emerson OPC Server on the client machine.....	46
7	Honeywell/Measurex OPC Server (to Foxboro OPC I/O Gate).....	51
7.1	Configuration steps on the HMX RaeServer OPC Server.....	51
7.2	Configuration steps on the OPC Client (AW70).....	55
7.3	Introducing the HMX RaeServer on the AW70.....	55
8	ABB rtdb OPC Client (to Foxboro AIM*OPC Server).....	59
8.1	Introduction.....	59
8.2	Requirements.....	59
8.3	To do's on the Server.....	59
8.3.1	Create the user account.....	59
8.3.2	DCOM Settings:.....	59
8.4	To do's on the Client.....	59
9	The Foxboro FDSI OPC Driver for FBM232 and FBM233.....	61
9.1	What you should know.....	61
9.2	Related documents.....	61
9.3	General information about the OPC FDSI FBM.....	61
9.3.1	Requirements.....	61
9.3.2	Difference between the Single and Redundant FBM.....	62
9.3.3	Difference with other 200-series FBM's.....	62
9.3.4	Limitations.....	62
9.4	Step by Step procedure: Redundant OPC FBM to Matrikon Simulation Server.....	63
9.4.1	The FDSI Configuration tool.....	63
9.4.1.1	Purpose of the FDSI Configuration tool.....	63
9.4.1.2	Installing the FDSI Configuration tool.....	64
9.4.2	Installing the FDSI OPC OPC Client driver.....	67
9.4.3	Configuration needed on the server side.....	68
9.4.4	The Port Configuration File.....	69
9.4.4.1	What is the port configuration?.....	69
9.4.4.2	Creating a Port Configuration file.....	69
9.4.5	The Device Configuration File(s).....	73
9.4.5.1	What is the Device Configuration?.....	73
9.4.5.2	Creating a Device Configuration file.....	74
9.4.6	Creating a Control Database.....	78
9.4.6.1	The Port ECB (ECB200 or ECB202).....	78
9.4.6.2	The Device ECB (ECB201).....	80
9.4.6.3	The DCI I/O blocks.....	82
9.4.7	Loading the driver in the FBM.....	83
9.4.8	Downloading the Port configuration file.....	85
9.4.9	Downloading the Device Configuration File.....	86
9.4.10	Starting communication.....	88

9.4.11	Backup FBM in redundant configuration.....	89
9.4.12	Master/Tracker Role switching in redundant configuration.....	91
9.5	OPC Diagnostic Client.....	91
9.5.1	Connecting the client to the FBM.....	91
9.5.2	Diagnostic view.....	92
9.5.3	Log Files.....	92
9.5.4	OPC Browser functionality.....	92
10	Troubleshooting / FAQ.....	94
10.1	General tips for troubleshooting OPC.....	94
10.2	Local Clients connect but remote clients can't.....	95
10.2.1	Unable to add groups.....	95
10.2.2	Tags in OPC clients show bad status, even though the blocks on the server are healthy in I/A:.....	95
10.2.3	While attempting connection: RPC server not found.....	96
10.3	Foxboro OPC I/O Gate.....	96
10.3.1	The I/O Gate windows stays empty.....	96
10.3.2	Get error message: "Interface not registered".....	96
10.3.3	Registering the OPCProxy.dll.....	97
10.3.4	Can't get ECB's on-line.....	97
10.3.5	ECB's in System Management repeatedly switch red/white.....	97
10.3.6	None of the configured items work.....	97
10.3.7	OPC Items do not work and group scan rate shows "0 msec".....	97
10.3.8	OPC I/O Gate crashes after installing QF1001908.....	97
10.3.9	Blocks in the OPC I/O Gate appear, but slowly.....	98
10.3.10	Only read of OPC items possible, No writes!.....	98
10.3.11	When writing OPC items, all ECB's go OFF-Line.....	98
10.4	AIM*OPC Server.....	98
10.4.1	Can't enter a license key because OPCDAO or OPCHDA are not in list.....	98
10.4.2	No packages at all in the list when trying to enter the license.....	98
10.4.3	OPC Server does not start (does not show in systray).....	99
10.4.4	Error "PSTCFG: Interface Not Supported".....	99
10.4.5	Error: Access Violation after re-installing OPC Server 3.3.....	99
10.5	APIAdmin tool.....	100
10.5.1	No Servers found when invoking APIAdmin.....	100
10.6	Only read of OPC items possible, No writes!.....	100
10.7	Network connection setting not available after installing OPC.....	100
10.7.1	Not all items become active.....	100
10.7.2	Analog values shown in the OPC I/O Gate show ".0000".....	100
10.7.3	Cannot start the OPC Server Configurator.....	101
10.7.4	Initial connection fails, retry always works.....	101
10.8	FDSI FBM & OPC Driver (Work In Progress).....	101
10.8.1	Hey... It doesn't work.....	101
10.8.1.1	Check the port configuration.....	101
10.8.1.2	Are the correct files downloaded?.....	102

10.8.2 FBM Not responding to commands, Rebooting the FBM.....	103
10.8.3 Diagnostic Blocks and parameters.....	104
10.8.3.1 ECB Parameters.....	104
10.8.3.2 Diagnostic DCI Blocks.....	104

1 Introduction.

Based on a true story this piece of documentation was intended to help setting up a **Foxboro OPC Server** and/or **Foxboro OPC I/O Gate Client** in combination with each other or a third party OPC client or server.

Please note that this is NOT an official Foxboro Document, neither is it meant to be a replacement of the official manuals that come with any of the products described in here. This is a “quick ‘n dirty” manual. Based on the author’s personal experience with the products in this document. The official documentation overrules the information contained here unless explicitly stated otherwise;-)

I did not attempt to invent anything more than the bare minimum myself which is why parts of this document are extracted from any other resource available for both the **AIM*OPC Server** and the **OPC I/O Gate** (The Foxboro OPC Client) product.

Feel free to share this document as much as you do see fit. Remarks, additions etc. are always welcomed.

1.1 Disclaimer.

Due to the inherently complex nature of computer software and the fact that third party software is involved, the author does not warrant that the software described in this document or this documentation is completely error free, will operate without interruption, is compatible with all equipment and software configurations, or will otherwise meet your needs. Accordingly, this documentation is provided as-is, and you assume all risks associated with its use. The author makes no warranties expressed or implied, with respect to this document. In no event will the author be liable for indirect, incidental or consequential damages, including, without limitation, loss of income, use, or information.

1.2 Conventions.

The conventions used in this document are listed below.

When you see this:	It means this:
[Filename]	This typeface indicates a filename of which the contents are printed starting on the next line. This line is NOT part of the file contents.
STATION# some command <cr>	Text printed like this is ASCII text as it appears in a file or on screen. User data that has to be entered is printed in bold . Also used to display a pathname or filename in normal text. When used to indicate a command, type the bold printed command until <CR> which means to hit the ENTER key.
Use this data	This typeface is used to show a list of data to be entered in a location indicated in the text. Also used to describe a procedure.

When you see this:	It means this:
<Alt_F4>	Text displayed like this means: press the keys mentioned between brackets <u>simultaneously</u> . In this case press the “ALT” key on your keyboard <u>together</u> with the function-key “F4”
Start/Run...	Indicates a menu sequence. Here it reads Press the Start button and on the next menu select Run...

1.3 Revision history

Revision number:	Description:
Version 0.2	<ul style="list-style-type: none"> • Preliminary
Revision 0.3	<ul style="list-style-type: none"> • learned more and wrote it down.
Revision 0.4	<ul style="list-style-type: none"> • Upgraded to OpenOffice for doc maintenance. Lost my work just once too often with standard software. • Some additions to the manual
Revision 1.0	Initial 1.0 release (sort off)
Revision 1.1	<ul style="list-style-type: none"> • Adopted Invensys standards for Documents.
Revision 1.2	<ul style="list-style-type: none"> • Added feedback from readers, tips and hints from Roel Olieman and Bram Marsman, thanks guys! • Updates with Windows XP stuff
Revision 1.3	<ul style="list-style-type: none"> • Added AIM*Historian with OPC I/O Gate collectors. • Added AIM OPC on WindowsXP.
Revision 1.4	<ul style="list-style-type: none"> • Roel Olieman, Bram Marsman and Niels Peerdeman again provided much feedback about their OPC Experience. Thanks for that! • Small changes to the Emerson Provox OPC Server setup.
Revision 1.5	<ul style="list-style-type: none"> • Cleaned up some stuff. Many small edits. • Updated OPC Server on XP, Updated AIM*AT and OPC I/O Gate collectors on 2000/XP. (was a big mess) • Documented some trouble shooting tools like APITST and APISPY
Revision 1.6	<ul style="list-style-type: none"> • Updated the Honeywell/Measurex OPC configuration chapter. Need to enlarge the AIM*Historian and AIM OPC I/O Gate Client section.
Revision 1.7	<ul style="list-style-type: none"> • Added chapter about the FDSI OPC Driver for FBM232 and FBM233. Special thanks to Bram Marsman!

1.4 Related documents.

As far as the Foxboro products are involved the related documentation is found here:

The OPC I/O Gate:

After installation some info is found after installation on the AW70X under:

D:\opt\diw\iogates\reference_guide\index.htm and:

D:\opt\diw\iogates\opc\help\OPC Users Guide.htm

this in addition to the printed manual that comes with it.

The OPC Server:

A manual is provided with the product: **B0400DG** rev A or higher which is also available in PDF format. The **AIMOPC** Server version 3.2 is provided on the AIM*AT 3.2 CDROM accompanied by the documentation in PDF format.

Other brands:

For the other brand OPC products please refer to the manuals provided.

1.5 Introducing OPC.

OPC is originally short for *OLE for Process Control* and version 1.0 was released in 1996. During time the OLE was replaced by ActiveX and the current OPC version is 2.04 for DA and 1.0 for HDA. Both the Foxboro OPC I/O Gate and the AIM*OPC Server comply to these DA standards. The Foxboro OPC I/O Gate does NOT support HDA functionality.

1.6 A Two Box Approach.

This manual deals with (and I use Alex Johnson's words) a "**Two Box**" OPC approach only. This means that the OPC Client and OPC Server both run on different PC's.

I thought I had good reasons for that too. Although the support department will always suggest to take the **One Box** approach (This is when both OPC Client en OPC Server reside in the same box) this is not always possible.

The situations I have come across turned out to require that the OPC Server be in one box and the Client in another. My advise would be: If you can put them both in one machine, please do so. It will save you the trouble of introducing the Client and Server to each other and as a bonus you can skip the entire process related to Microsoft DCOM user accounts and the likes. This last bit can be a real pain. However: This manual deals with the **Two Box** implementations only. Things tend to be a little harder to setup, which gives me a reason to maintain this manual and gives me something to do.

2 The Foxboro OPC I/O Gate client (to Foxboro AIMOPC Server).

This Foxboro solution for the OPC implementation on I/A Series is based on the Foxboro I/A 70 Series Integrator¹ with an additional I/O Gate providing the OPC functionality. This I/A 70 Series integrator AW can, being part of the I/A Series system, provides the OPC functionality to the rest of the I/A Series Windows/UNIX platform. Once the OPC data is available on the AW70 Server it is available in the entire I/A Series system.

Currently **Invensys** is developing a hardware solution for the OPC Client. This solution will be based on a fieldbus module which runs Windows CE in ROM. This new OPC Client is coming available in I/A Series Version 8.0.

2.1 Requirements.

The basic requirements as described in the documentation supplied with the OPC I/O Gate:

- An I/A AW70 Integrator processor version 6.2, 7.0 or higher (NOT version 8.0). The Integrator function is currently not available on I/A Series Version 8.0.
- A Foxboro OPC I/O Gate (Q0301HT).
- An OPC Server to connect to would be nice.

This chapter is written and verified using:

Product: **OPC I/O Gate**
Partnumber: **Q0301HT**
Version: **3.0.1**
Date (US): **08/02/2002**

For this example we used an AW OPC Client with letterbug AWNT01 connecting to an AW OPC Server with letterbug 81AW04. The HOSTS table reflects this:

¹ The Integrator is an AW70 with Control.

```

*****
# Start of I/A hosts
# created Wed Jan 30 14:28:01 2002
*****
#
# The following host entries were created by the I/A
# Software Install sub-system. Any additional entries
# should be placed outside the Start and End delimiters.
#
151.128.8.65    AWNT01  loghost localhost
#
*****
# End of I/A hosts
*****
10.31.19.227  AWNT01
10.31.19.226  81AW04
10.31.19.225  81AW01

```

2.1.1 General hardware and system software.

This product requires a Foxboro *I/A 70 Series integrator* to function. (this is the AW70X and AW70CP in SysDef depending on the I/A Series version). Although it is not very well defined in any of the documentation, the Integrator function *is* a requirement. I/A Series software version must be 6.2 or higher.

Internet Explorer 5 or higher is recommended to view the online documentation.

2.1.2 OPC I/O Gate licensing.

The **OPC I/O Gate** requires a license which is not part of the shipment. The license key must be obtained separately. For a license instructions or form you may have a look at:

D:\opt\diw\iogates\opc\help\OPC License.htm on your AW70 after the I/O Gate was installed.

You will need at least the following information:

- The US order number (also placed on the documentation supplied with the software).
- The End user identification (Customer/site name).
- The AW station name (letterbug).
- Whether or not the OPC Client station is the CSA Server.

Forward this information according the instructions in the form or e-mail directly to dmartini@foxboro.com.

You will receive precise instructions about installing the license on your system.

2.2 Installation.

Some installation notes regarding the OPC I/O Gate installation.

2.2.1 Installing the OPC I/O Gate.

The I/O Gate is supplied on a single CD-ROM containing the installation software. After loading the CD in the CD-ROM drive you will get the screen as shown below:



Figure 1 OPC I/O Gate installShield wizard .

To install the gate click **Next >** .The installation will take a few seconds and will inform the user of a successful installation:



Figure 2 I/O Gate is installed.

The I/O Gate is now installed on your I/A Station and can be started via the **Start** menu with: **Start/Programs/IOGates/OPC/OPCIOGate**. The Gate requires a license to operate. If you haven't done so, obtain a license as this would be a great moment to get it in place. Follow the instructions that come with the license key that you receive from Foxboro.

2.2.2 DCOM and Server side configuration.

This is the tricky part but let's give it a try anyway....

OPC is based on Microsoft DCOM and ActiveX technology. DCOM layer does not start working all by itself and thus needs some configuration.

These are basic steps we must go through to have at least a little chance things will start to work:

1. Register the OPC Server on the AW70 platform (Section: 2.2.3).
2. Create the “**Fox**” user account on the Server.(Section: 2.2.4).
3. Set the DCOM security levels.(Section: 2.2.5).
4. Optionally configure *alias* files if we run into configuration constraints in the I/O Gate side (Section: 2.2.6).
5. Setup automatic startup of the I/O Gate after system startup (Section: 2.2.7).

Let’s start with the first one:

2.2.3 Registering the OPC Server.

OPC is a client/server based solution. For the Client to be able to connect to a given Server, this Server must be known to that Client. This process is known as “registering the Server”. (*Remarks like “Is the server registered yet in my client” make you stand out in a crowd. You seem to be very “on top of things” and people will listen to you better;-)*).

There are several levels of doing this and they range from “*Piece of Cake*” to “*Where Is Einstein When You Need Him?*”. The procedures are very dependent on the manufacturer of the server/client. In this little example we take the Foxboro AIM OPC server. The procedure is somewhere between “*Cake*” and “*Einstein*” so it will be fun. Since the I/O Gate is now in place and we are able to start it, its time for a “*Server to be registered*”.

2.2.3.1 On the AW70 machine: Registering the Server.

This procedure depends on the instructions provided with the OPC Server you are working with. In this example we chose the **Foxboro AIM*OPC Server** which is installed on another host (AW70 or OFF-Platform machine) (It’s no fun when it is on the same machine as the Client as registering is done automatically in that case).

For the **Foxboro AIM*OPC Server** to Register on the Client side we need the “*registry key*” from this server as it is found in the Server’s Windows registry.

You should perform this procedure on the AIM OPC Server machine.

For the Foxboro OPC Server this “*registry key*” contains the letters “OPC” (but this is not always true for other brands though) and the required key can be obtained using the **regedit** program. Invoke the **regedit** program from **Start/Run...** and type `regedit` in the dialog:

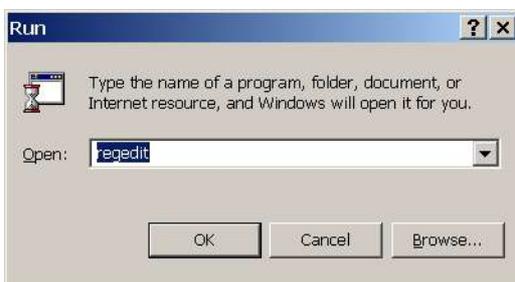


Figure 3envoking regedit.

The resulting screen shows the **Registry Editor**. Open the (highlighted) “**HKEY_CLASSES_ROOT**” tree:

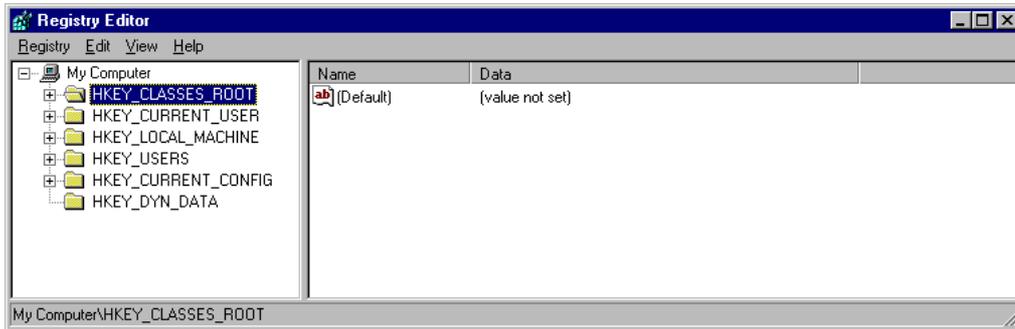


Figure 4The desired key is in HKEY_CLASSES_ROOT tree.

And scroll down until you locate the **AIM.OPC** key (shown expanded here...)

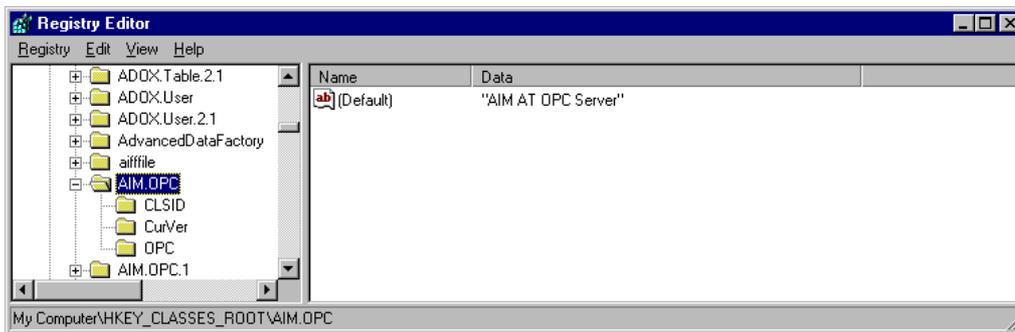


Figure 5Located the AIM.OPC key.

From the Menu choose **Registry/Export Registry file...** and save the selected key:

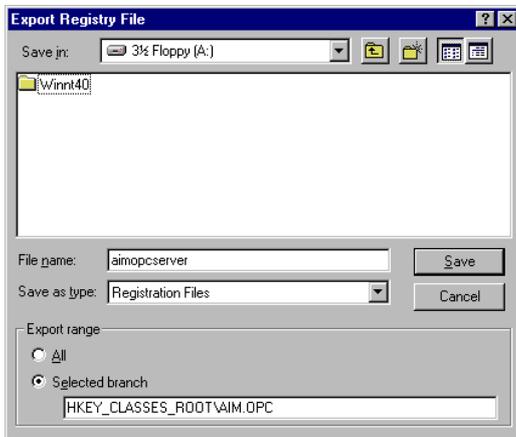


Figure 6Export key to floppy.

The name given in the screenshot is just an example. On the floppy you now have a registry key (aimopcserver.reg) which must be loaded on the OPC client machine. Close the **regedit** application and remove the diskette from the drive.

Note:

Sometimes copying the registry key as described above does not work. In that case it may be an idea to install the OPC Server software on the machine where the OPC Client is installed.

Sometimes additional DLL and other files are required on that platform. Installing the OPC Server

will solve those dependencies automatically.

2.2.3.2 Loading the OPC Server reg key on the OPC client host.

Next step is to take the diskette with the registration key that was just created on the OPC Server platform and take that to the OPC Client machine and enter the registration key there. This is done by inserting the diskette in the drive of the AW70 and run `a:\aimopcserver.reg` in the run dialog as shown here:

Start/Run...



Figure 7 Loading the registry file

After entering the data you will be informed of this:



Figure 8 Registry data entered successfully

At this point the OPC Server is *Registered* on the AW70 OPC Client PC.

Note:

When you are working with an OPC Server from any brand and this does not solve the communication problems, choose the easy way and install the OPC Server on the client machine.

This will make sure that all the required DLL and executables required to establish a link between the two are available on the OPC Client machine.

2.2.4 Creating the Foxboro account on the Server.

For DCOM to function properly the Foxboro standard user (*Fox*) must be available on the Server side so this account for the user "**Fox**" must be added on the remote Server using the **NT User manager. (Start/Programs/Administrative Tasks (Common)/User Manager).**

Choose **User/New User** from the menu bar and enter the data as in the example:

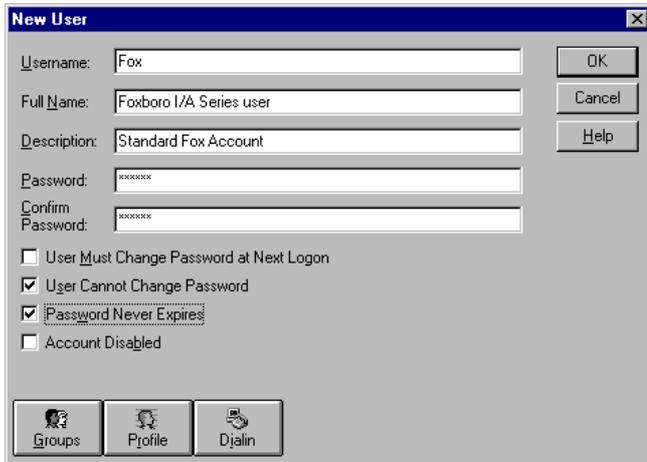


Figure 9 Setting up the Fox account.

Full name and description can differ from this but the Username “**Fox**” must be typed as in the example and the password must be “**gnomes**” (without the quotes). The password must never expire.

2.2.5 DCOM Settings.

DCOM is the underlying protocol used by OPC to exchange the process data between the different systems. Since it requires interaction between these systems things like authentication and privileges must be configured accordingly. The tool to perform this task is the “**DCOM configurator**”. DCOM must be configured on both Client and Server side.

2.2.5.1 On the OPC Server side.

Start the DCOM configurator from **Start/Run...** and enter **dcomcnfg** in the dialog:
When selecting **OK** this is the screen you get:

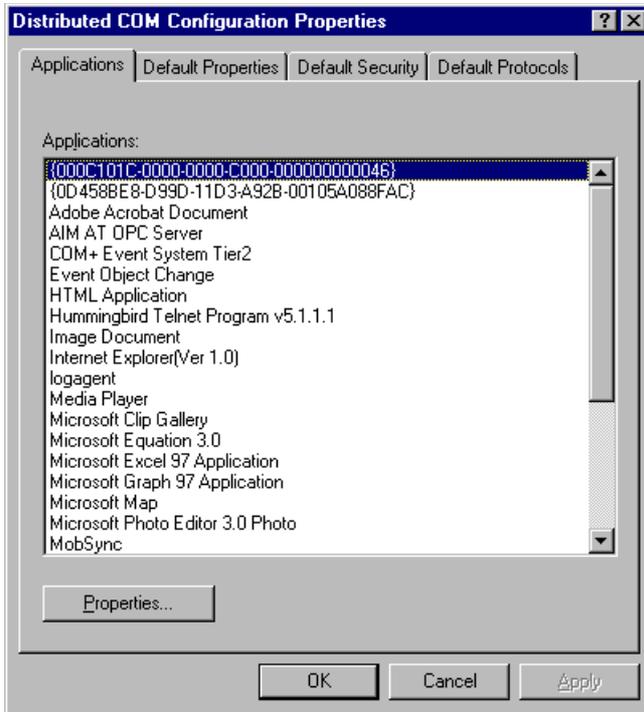


Figure 10 DCOM initial screen: Applications tab.

If the default settings for the selected server do not result in a working setup try matching the DCOM settings as indicated below.

Select the **Default Properties** TAB and make the modifications as indicated:

- The **Enable Distributed COM on this computer** box must be checked.
- **Default Authentication Level** to **(none)** and
- **Default Impersonation Level** to **Anonymous**

Note for Windows 2000 Family (and possibly XP hosts):

Setting the Default Impersonation to Anonymous may cause network and Dialup connections previously defined are no longer visible.

Windows Explorer uses the Default Impersonation level settings to access these objects.

Anonymous access does not give the correct privileges to display these. See also Microsoft knowledge base Article 273461.

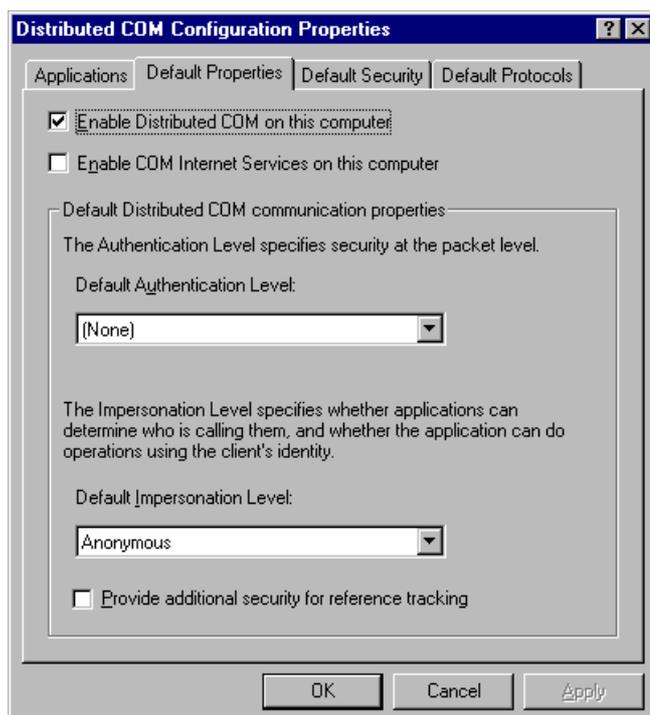


Figure 11DCOM: The Default Properties tab.

Select the **Default Security** tab and make the modifications as indicated:

- For the **Default Access Permissions** section click the **Edit Default** button; When the dialog box pops up click the **Add** button another dialog will pop up. Add the user "Fox" to the list.
- Perform the above steps (add user "Fox") to the **Default Launch Permissions** section of the **Default Security** tab.

These steps should be sufficient to enable the Foxboro OPC Server to accept the Foxboro OPC Client.

2.2.5.2 On the OPC Client side.

The DCOM settings for the OPC I/O Gate are found in the documentation and come down to making a few changes to the **Default Properties** tab.

- **Default Properties** tab; Select the box: **Enable Distributed COM on this computer**
- Also in the same tab the **Default Authentication Level** should be set to **None**.
- And finally the **Default Impersonation Level** should be set to **Anonymous**.

2.2.6 Alias files.

Since the I/O Gate acts as an interface between the "real world" and the (much older) "Foxboro world" we can run into some constraints that happen to be a part of our environment. You should think of these limitations when configuring the I/O Control database:

- *Port ECB (ECB98 / 99) parameters:*
 - SVRAPP parameter (OPC ProgID string) is limited to 19 chars.

STADDR (server IP address or machine name string) limited to 19 characters.

- *Device ECB (ECB100) parameters:*
IOM_ID This name should be the same name as the ECB Name.
DEVADR parameter (OPC Group string) is limited to 19 characters.
- *I/O Control Blocks:*
Item field string must 30 characters or less and ALL CAPS.
- All parameters in the database are stored/presented as ALL CAPS!

To alleviate this problem the OPC I/O Gate provides the option to use *alias* files that will allow avoiding such problems should they arise. These *alias files* are always consulted during the retrieval of any OPC related communication and the files are located in the

D:\opt\diw\iogates\opc\cfg\ directory.

Two files in here are of concern: `aliasnames.ini` and `opciogate.ini`

Where `opciogate.ini` holds configuration info on the OPC version that will be used to translate any info stored in there between Foxboro I/A and the outside world.

`aliasnames.ini` holds information related to the specific OPC data such as:

- The Server aliases,
- The Group aliases and last
- The OPC item aliases.

The files have a typical layout like this:

The `opciogate.ini` file example where the OPC server from ICONICS is forced as a Version 1.0 compatible OPC Server:

```
[ICONICS]
VersionCompatibility = 1

[OLD_V1OPC_SERVER]
VersionCompatibility = 1

[LOGINFO]
LOGTYPE=0
```

Figure 12An example `opciogate.ini` file

The file will give you the option to force the I/O Gate to assume a version 1.0 compatible server when connecting to it. The default behavior is to assume version 2.0 and if that fails fall back to version 1.0. Sometimes this approach fails and communication breaks. In that case enter the server in this configuration file.

The `aliasnames.ini` file:

```
[SERVER]
RSLINX = "RSLinx OPC Remote Server"
HMXOPC = "Hmx.RaeServer_0"

[GROUP]

[ITEM]
RSLITEM1 = "[Liberty1].T1.B1"
RSLITEM2 = "[Liberty1].T1.B2"
```

Figure 13 An example `aliasnames.ini` file

This file serves as a translator between *items* in the file on the left which are the Foxboro ICC compatible types and the *items* on the right that are the “Real world names”. This is to overcome the incompatibilities between the two as indicated in the start of this chapter.

The example above allows to use the OPC Server references RSLINX or HMXOPC in your OPC Client Control Database (in the ECB99 where you define this) instead of the long “real” name.

2.2.7 Start OPC I/O Gate at system startup.

The OPC I/O Gate can be started from the Start menu found in Windows but it would be better if it would be started in due time after the AW70 is fully on-line. A Windows “cmd” script is provided for this purpose. For the procedure to enable automatic startup for the OPC I/O Gate:

An item must be added to the startup group. Follow these steps to make the modification:

1. Select the **Start** menu button in the Windows Task Bar. Then select **Settings** and **Taskbar**
2. In the **Taskbar Properties** dialog box, select **Start Menu Programs** then select the **Add** button.
3. In the **Create Shortcut** dialog box, select the **Browse** button.
4. In the **Browse** dialog box, locate the "D:\opt\diw\IOGates\OPC\Bin\OPCstart.cmd" file. Select the `OPCstart.cmd` file so that its name appears in the **File name:** field then select the **Open** button.
5. Select **Next** in the **Create Shortcut** dialog box.
6. In the **Select Program Folder** dialog box, locate and select the **Startup** folder.
7. Select **Next** then **Finish**.

After this the OPC I/O Gate will be started automatically when the AW70 is running.

(This will not work when the NT shell was disabled at boot time).

2.2.8 OPC I/O Gate Log files.

The communication can be hard to set up. Some help may be found in the log files found in the system.

One place to look may be the Event Log as maintained by Windows itself. Since OPC uses DCOM, errors on this level are logged in the Event log. Errors can be for instance related to insufficient privileges reported by the server. Another place is the log files maintained by the I/O Gate. These logs are found in "D:\opt\diw\IOGates\OPC\log". Here you will find the days of the week and (if logging was enabled) the log info. Please note that these logs grow until the disk is full and the logs can only be toggled by stopping and starting the I/O Gate.

2.3 Creating a Control Database.

At this point a control database can be made. The steps include creating the ECB blocks for the OPC Server connection and after that the I/O (F-)blocks that will do the talking. Before we continue it is nice to have the OPC I/O Gate up and running so it must be started (from the Windows Start menu) the screen as below will appear (unless you did not have a license yet):

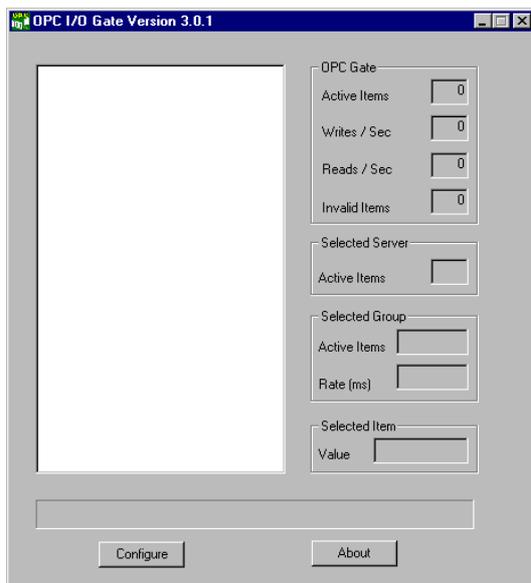


Figure 14 OPC I/O Gate up and running

The Server ECB is used to define the OPC Server on the client side. The ECB for this is the ECB99/98. ECB99 is used for network setup and ECB98 is used for serial communication. This manual deals with ECB99 only.

A simple database configuration consists of three steps:

1. Configuring a port (ECB 99 /98) and get it “on-line” and White in System Management.
2. Configuring devices (ECB 100) and get it “on-line” and White in System Management.
3. Configuring data items (F-blocks in the ICC database).

A diagram showing the logical relation between the three:

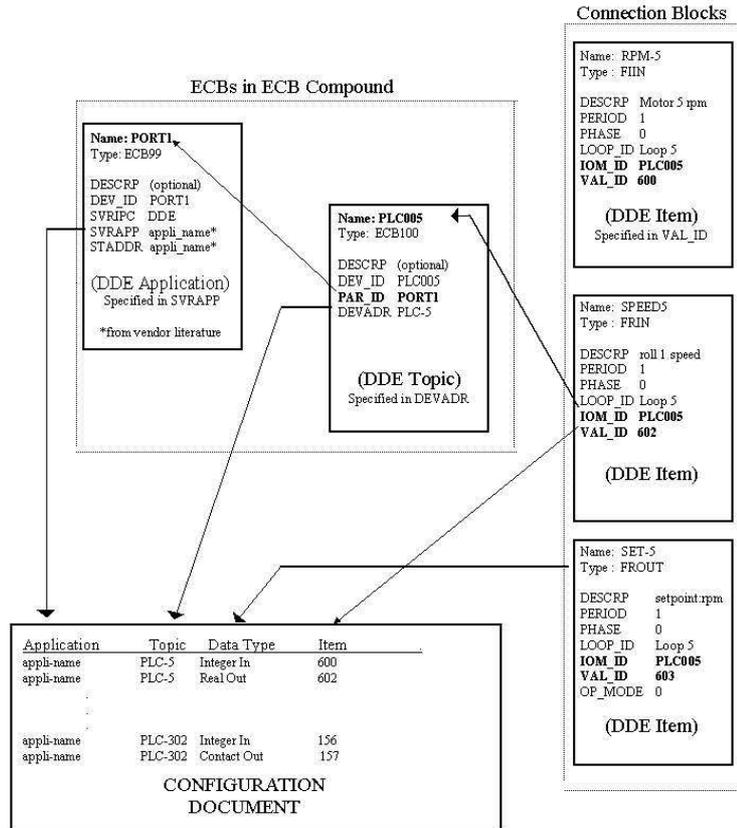


Figure 15 Internal affairs

So the first step to take is create the ECB's for the job.

2.3.1 The Server ECB (ECB99).

The first ECB that must be created is the ECB99 which defines the OPC Server specifics like the Server application name, the Server hostname or IP address. It defines the (remote) OPC Server station to our local OPC Client AW70.

In this ECB the most important parameters are:

- **SVRIPC:** usually this field takes OPC or OPCNR²
- **STADDR:** This is the Server's IP address or name as it is in the "hosts" table.
- **SVRAPP:** The OPC application name as found in the Registry key or alias³ file.

2 Entering OPCNR requires QF1001908 or Windows XP to be installed on the OPC I/O Gate integrator AW.

3 Aliases for are set for the I/O Gate is set in the appropriate configuration file.

D:\opt\diw\iogates\opc\cfg\aliasnames.ini.

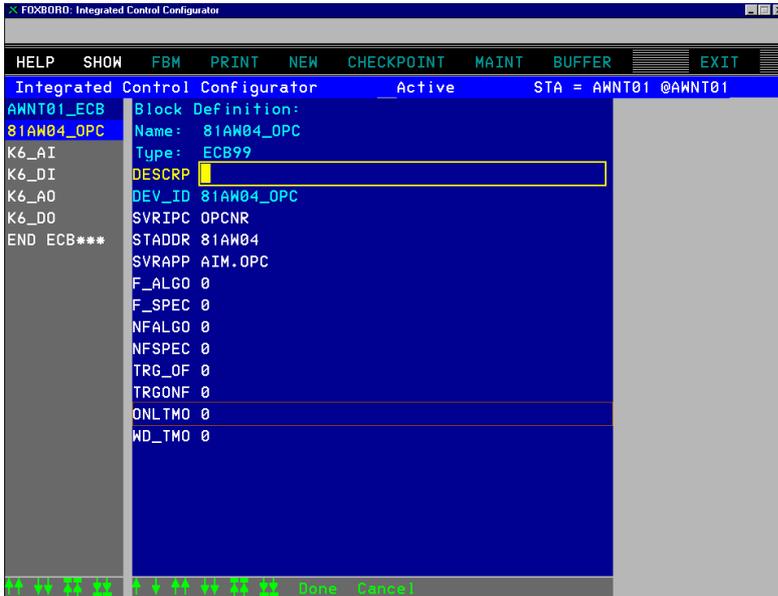


Figure 16A An example of ECB99 completed

Force a checkpoint and after this we should be able to see the ECB99 showing up in System Management. If there is network connection to the OPC Server and the network is OK *and the OPC I/O Gate* is running, we must be able to get the ECB99 on-line and white in System Management.. If you can't get the ECB99 on-line there is a need to check the configuration once again. You must be able to perform this task otherwise the OPC connection will not be successful later.

2.3.2 Creating the Group ECB (ECB100).

When the ECB99 functions as desired we can define a “group” to be created on the server. The server will maintain this “group” with all items contained in it.

The “group ECB” is the ECB100. The most important parameters are:

- **Name/DEV_ID** A logical name for the ECB
- **PAR_ID** The Parent ECB (the ECB99 name)
- **DEV_ADR** The desired “OPC group” name (arbitrary)

Below a picture of a simple ECB100 with this data entered. For this example a group containing all Analog inputs will be created and the name chosen in this example is **K6_AI**:

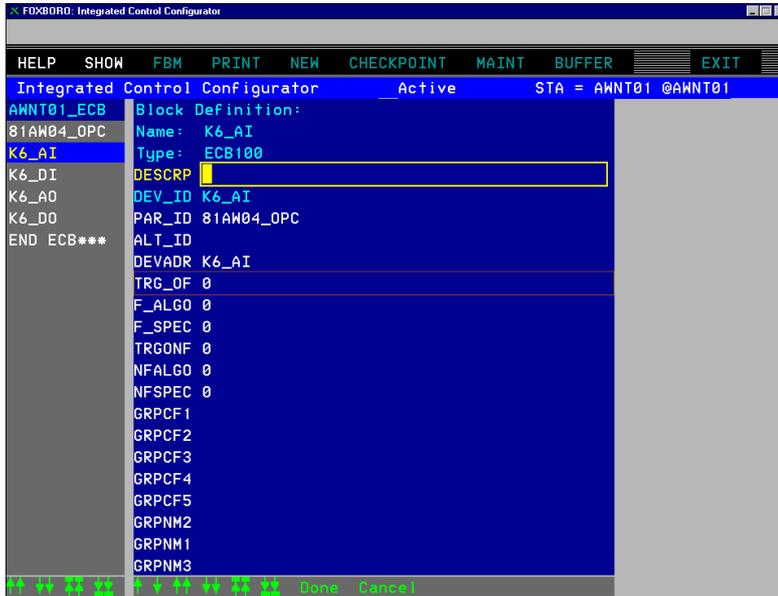


Figure 17 An ECB100 example

Following this logic we can create ECB100 for the DI, DO and AO signals too. System Management will show all these ECB's in a screen similar to the one below (remember to checkpoint first).

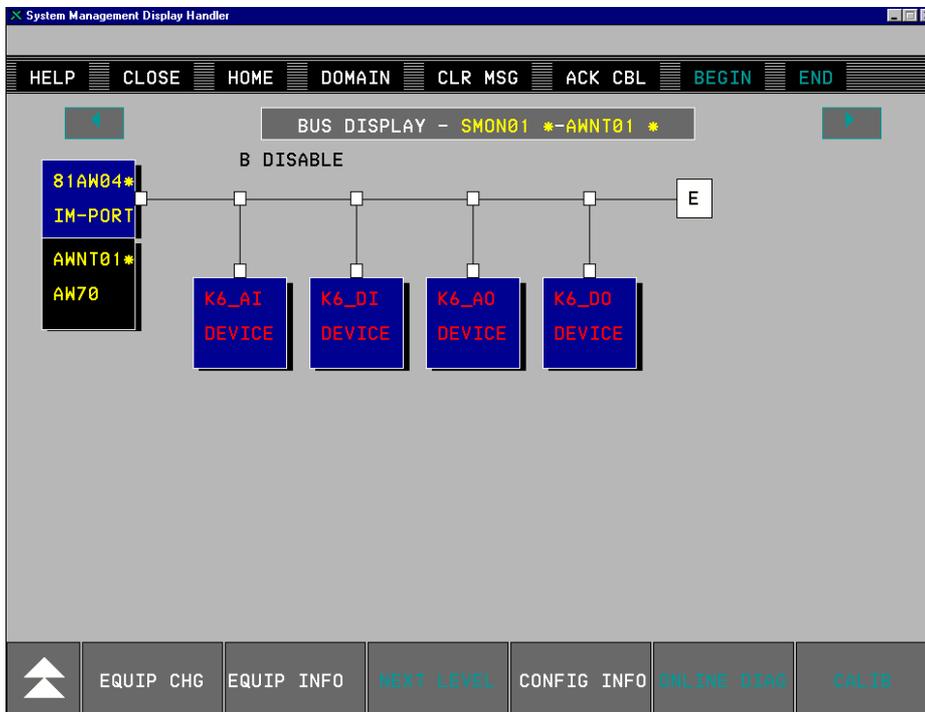


Figure 18 All required ECB100's are created

When the OPC I/O Gate process is up and running, we must be able to get ALL ECB's configured until now in the On-Line (White) state. Below you can find an example of an ECB100 which was assigned to the Digital outputs for our setup.

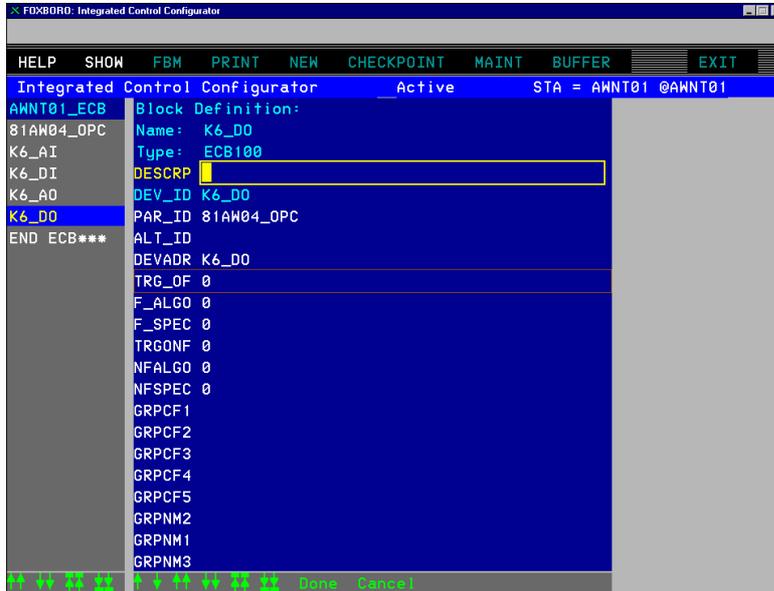


Figure 19 ECB 100 for Digital outputs group

Similar blocks can be built for the four types of I/O we were dealing with. After we have done this, it is time to start the OPC I/O Gate application I (if it is not already running that is). The picture below will show you the initial screen of the OPC I/O Gate in action.

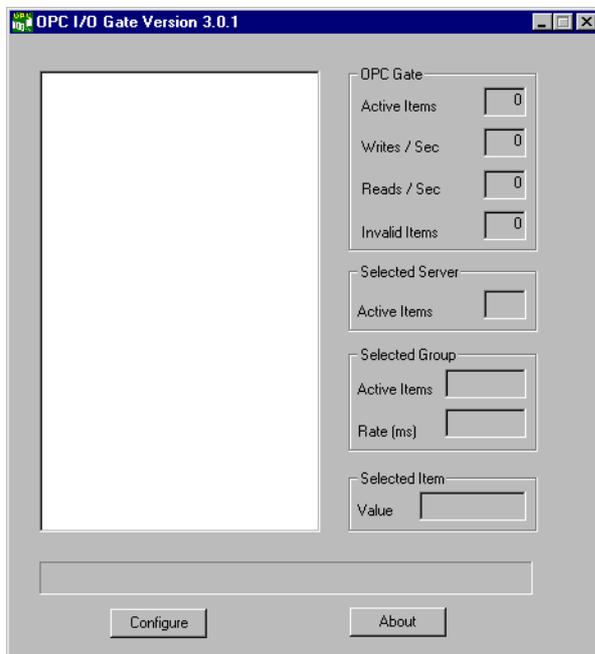


Figure 20 OPC gate window initial state, no Servers found yet.

After some time (and this can be up to one minute, so be patient), the OPC Server will show in

the tree when it was found.

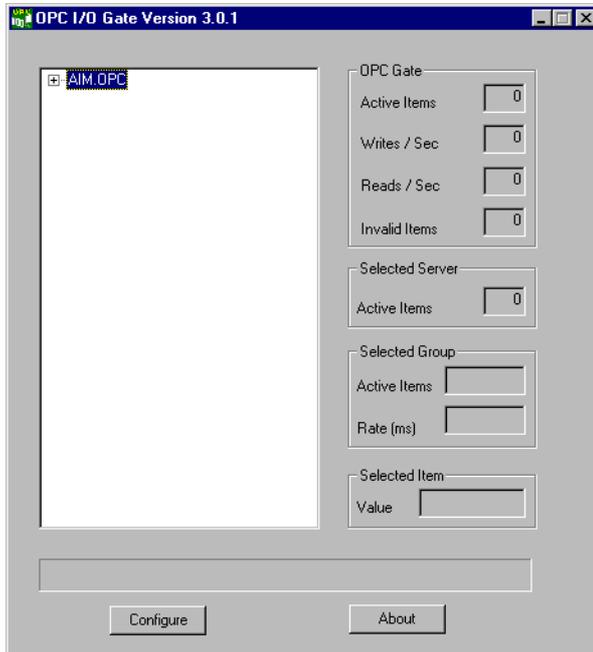


Figure 21 AIM.OPC Server was contacted successfully

Please note that for this to happen, there must be a path (quite physical actually) from the OPC Client to the OPC Server machine.

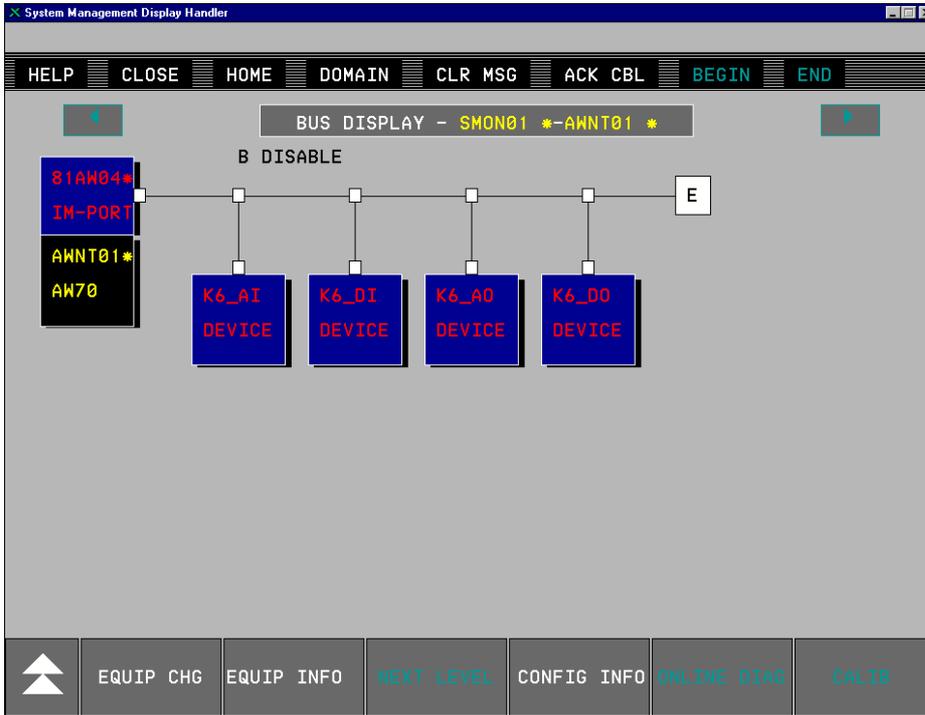


Figure 22 All ECB's 99 and 100 added but off-line

When you put the ECB 100 blocks in the ONLINE state, they should also (apart from being white in the System Monitor) show up in the tree on the OPC I/O Gate window.

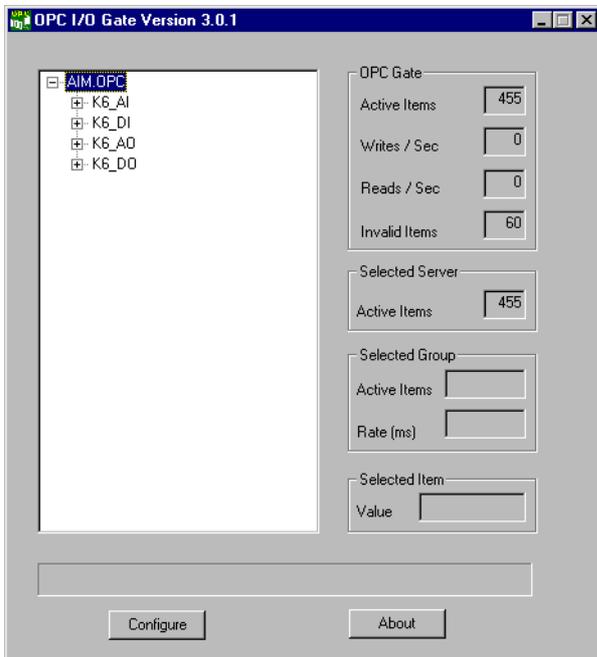


Figure 23 When ECB's are configured/on-line they show up here

You are now ready to proceed with your Control Database. These F-blocks are explained in detail

in the on-line manuals after installing the OPC I/O Gate application. Refer to the beginning of this chapter on where to find it.

3 The Foxboro AIM*OPC Server on Windows NT.

The **AIM*OPC Server** is the application that provides the Foxboro OPC Server implementation. The name may give the impression that AIM*AT must be present on the system but this is not the case. Some tools as found on the AIM*AT installation CD are however required to allow setup and implementation of this OPC Server. The AIM*OPC Server is fully OPC DA 1.0a and 2.0.4 compliant and fully compliant with the OPC HDA version 1.0 standard.

3.1 Requirements.

The hard- and software requirements for the OPC Server as indicated in the accompanying documents:

- A Pentium Class processor, Recommended 500MHz or higher.
- Windows NT 4.0 SP5 or higher, Windows 2000 SP2 for the "off-platform" server.
- 128Mb of RAM
- 25Mb disk space.

[Additional requirement NOT in the manual:](#)

- Internet Explorer 5.0 or higher for "XML" support required by OPC Server.

Reference Documents:

- **The I/A Series AIM*AT OPC Server User's Guide – B0400DG.**
- **AIM*API User's Guide – B0193YN.**
- **AIM*AT Installation Guide – B0193YM.**

3.1.1 AIM*OPC Server licensing.

The OPC Server is a product that requires licensing from Invensys. The license is NOT shipped with the product but must be acquired later. Licensing involves the system's "*hostid*" which can be generated on the actual system after the I/A or API admin utility installation. This is the main reason we want some of the AIM*AT utilities installed on the system where the OPC Server will reside. This applies for both ON- and OFF platform configurations.

Data required for licensing:

- The **Hostid** of the station where AIM*OPC is installed.
- The Type of OPC Server desired:
 - DA** for **Data Access** only
 - HDA** version which supports **Historical Data Access** in addition to DA.

3.2 Installation instructions.

To obtain the *hostid* for this station you are setting up it is best to install the AIM*AT utilities from the AIM*AT cd -rom. For this example I have chosen version 3.1 of this CD.

3.2.1 Installing the AIM*AT Utilities.

In order to allow convenient configuration of the AIM*OPC Server it is best to install the AIM*AT

Utilities and AIM*Historian from the AIM*AT installation CD-ROM. To install these utilities insert the CD-ROM in the drive and run the setup program on the CD. It may be best to install the AIM*AT suite completely not to miss anything. The utilities are the only thing we will be using but they may require more than the obvious to function.

3.3 Initial API Admin configuration.

If this is the first time you installed an AIM* product on your system, some initial configuration work needs to be done. This consists of executing the steps outlined next.

Start the **APIAdmin** tool from **Start/Programs/AIM*AT/API Admin**.

You will be prompted for a Server (in a drop down box) and if no server is listed you will have to configure one. For that click on the button the reads **Edit Server Connection**.

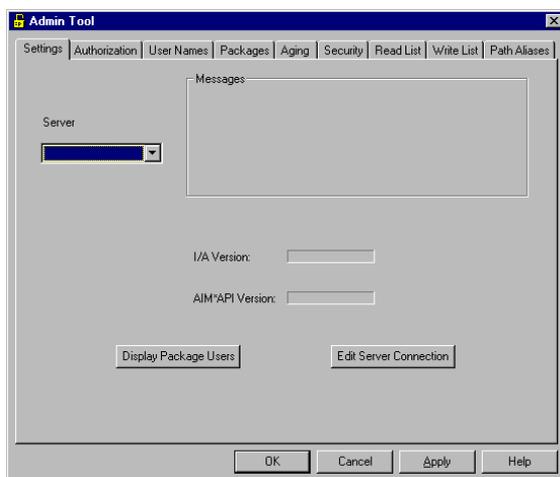


Figure 24Admin tool dialog

Create a Server alias (Edit Server Connection) if this is the first time logging in (note the values are just an example): The IP address you type in should of course match the Ethernet port that will be used to connect to that server. I.e: when you connect over the secondary Ethernet connection you type in the address of that Ethernet connection.

3.3.1 New AIMREG.DAT file.

This applies only when running the ApiAdmin utility on an AIM installation before version 3.1.2. To be able to authorize the OPC package a new `aimreg.dat` file must be copied to the `d:\opt\aim\bin` directory. This file contains the available software packages you can enter licenses for. The OPC server application was not available at the time AIM*AT version 3.1 was released and that explains why it was not in the `aimreg.dat` file. To resolve this, an updated `aimreg.dat` file is provided. The new version is supplied on the AIM*OPC installation CD. Make sure this file is *write enabled*. Since it usually is copied from CD to the harddisk it is wise to check the properties. After the modifications to the `aimreg.dat` file we can attempt a connection to the AIM API in order to license the package in the AW or “off-platform” host.

Start the **APIAdmin** tool from **Start/Programs/AIM*AT/API Admin**. You will be prompted for a Server (in a drop down box) and if no server is listed matching the one you are looking for a Server Connection must be added to the list.

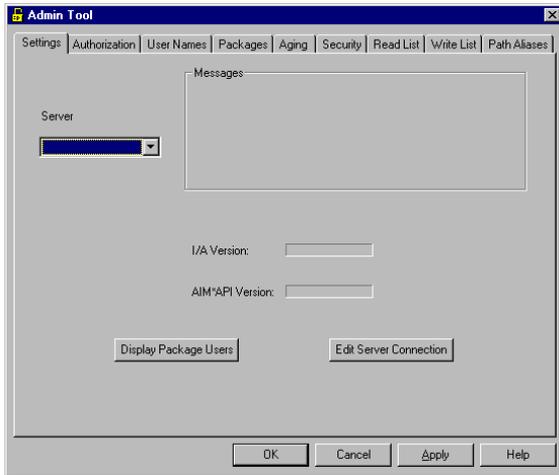


Figure 25 Admin tool dialog

Assuming a Server must be created we choose **Edit Server Connection**. Create a Server alias if this is the first time logging in (note the values are just an example):

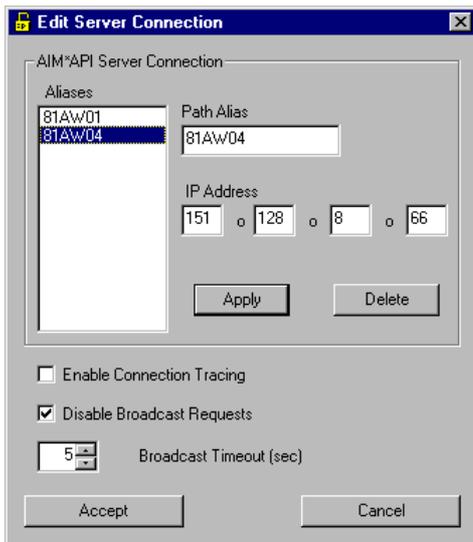


Figure 26 Create a Server connection

After completion of this step you must exit the **APIAdmin** program and re-open it. In the drop down box (**Server**) you now should have an entry for the AW that was just configured.



Figure 27 Login to the api admin tool

Enter the password for the API Server you are connecting to.

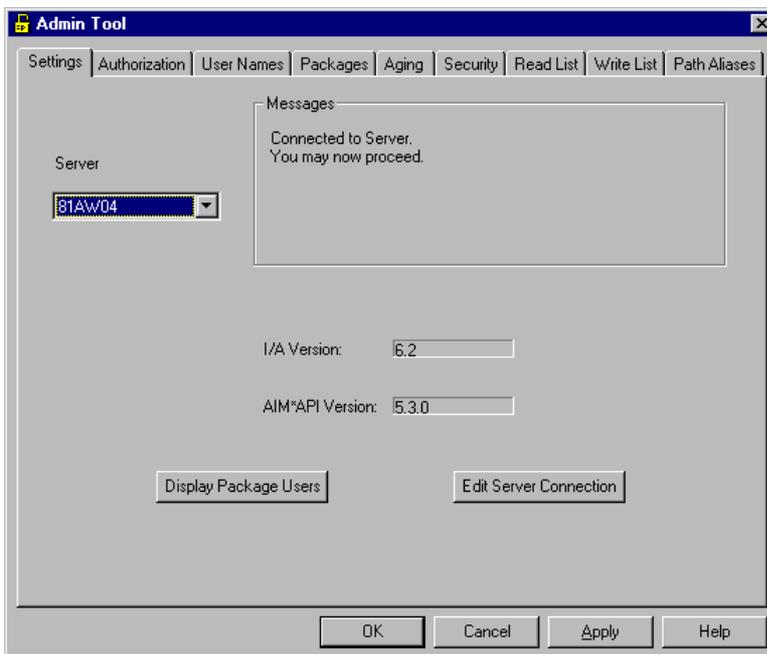


Figure 28A successful connection

In the example above an *Authorization* code was entered for the **OPCDAO** package. A user **[machine name]** must be added and this is shown here: *(The PC was named 81AW04 in this example).*

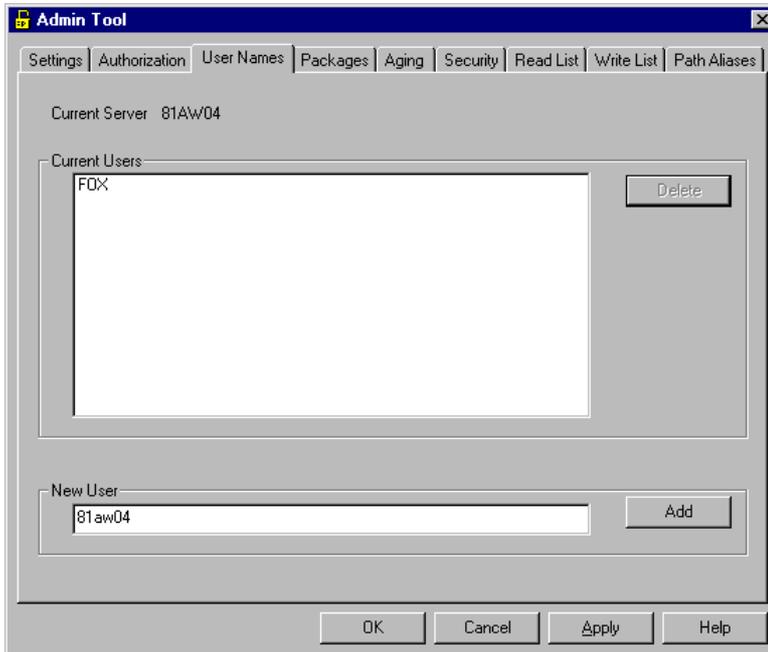


Figure 29 Adding the Machine name as user

Press **Add** when the name was completed which will bring you this result:

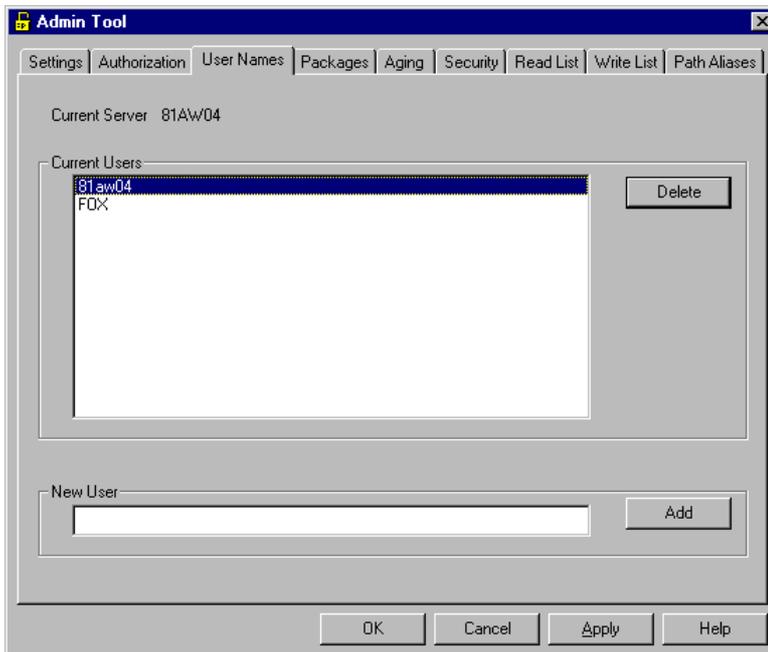


Figure 30 Machine 81aw04 is now added as user

A note for OFF Platform OPC Server installations:

- The Machine name of the *OPC Server* PC **MUST** be added as a *User name* on the *I/A* station that will accommodate the AIMAPI link into the *I/A* Series system. Note that only AP's and AW's are supported AIMAPI hosts within *I/A* Series systems.

- If you are using an **I/A 50 Series** machine to communicate with the OPC Server you must add the OPC Server's hostname as an AIMAPI user on the I/A 50 Series station using the **apiadmin** utility on that station. You must then get an OPC license for the UNIX machine as this is where the access to the I/A Series system is provided.
- Make sure that on the OPC Server side the AIMAPI configuration shows the path to the UNIX machine. This UNIX machine will hold the license information for your OPC Server application.

A typical example with two machines where:

- AWRD01 is a Windows NT PC with OPCHDA installed and
- AW51RD is an AW51B where the OPC Server is communicating with i.e. Where AIMAPI is configured.

Assigning the software:

To assign the software installed to the specific users select the **packages** tab, add the packages: **AIM*API** and **OPCDA** (or **OPCHDA**) as shown and assign them to the newly created user account (machine name):

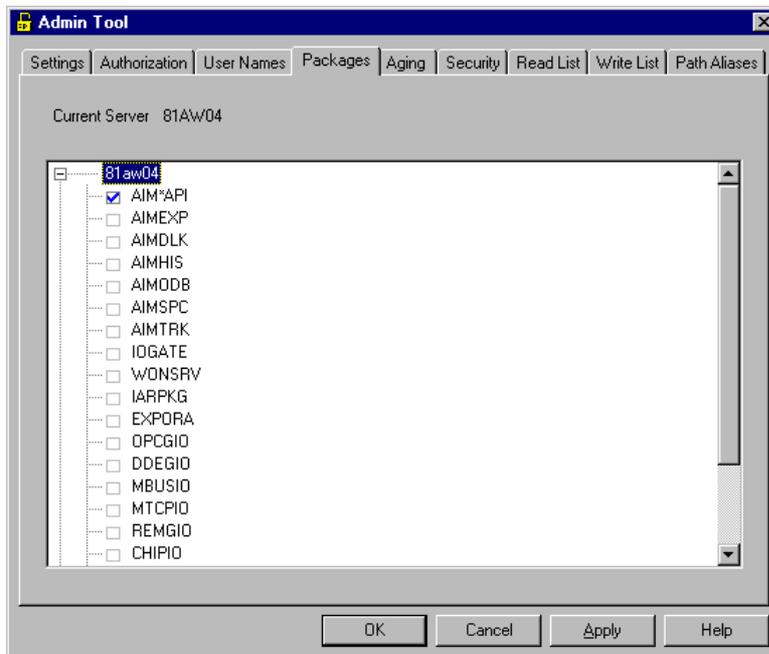


Figure 31 Assigned packages to this new user (machine name)

When you scroll down you must have a check mark next to the **OPCDA** or **OPCHDA** packages. Do not select them both, I have seen strange behavior when you do that.

4 The Foxboro AIM*OPC Server 3.2 on Windows XP.

The **AIM*OPC Server** is the application that provides the Foxboro OPC Server implementation. You do not need the AIM*AT Historian installed on your system. Some tools as found on the AIM*AT installation CD are however required to allow setup and implementation of this OPC Server. The AIM*OPC Server is fully OPC DA 1.0a and 2.0.4 compliant and fully compliant with the OPC HDA version 1.0 standard.

4.1 Requirements.

The hard- and software requirements for the OPC Server as indicated in the accompanying documents:

- A Pentium Class processor, Recommended 500MHz or higher.
- Windows XP SP1 or higher, Windows NT 4.0SP5 or higher, optionally Windows 2000 SP2 for the "off-platform" server.
- 128Mb of RAM
- 25Mb disk space.

The setup used in this chapter:

- A Pentium III 1GHz based processor.
- Windows XP SP1.
- AIM*AT Utilities version 3.2.
- AIM*OPC Server version 3.2.

The OPC Server is a product that requires licensing from **Invensys**. The license is NOT shipped with the product but must be acquired later.

Licensing involves the system's "*hostid*" which can be generated on the actual system after the I/A or API admin utility installation. This is the main reason we want some of the AIM*AT utilities installed on the system where the OPC Server will reside. This applies for both ON- and OFF platform configurations.

Data required for licensing:

- The **Hostid** of the station where AIM*OPC is installed.
- The Type of OPC Server desired:
 - **OPCDA** for **Data Access** only
 - **OPCHDA** version which supports **Historical Data Access** in addition to DA.

4.2 Installation instructions.

To obtain the *hostid* for this station you are setting up you will need to install the AIM*AT utilities from the AIM*AT cd -rom. For this example we chose version 3.2 of this CD.

4.2.1 Installing the AIM*AT Utilities.

In order to allow convenient configuration of the AIM*OPC Server it is best to install the AIM*AT Utilities and AIM*Historian from the AIM*AT installation CD-ROM.

To install these utilities insert the CD-ROM in the drive and run the setup program on de CD. It may be best to install the AIM*AT suite completely not to miss anything. The utilities are the only thing we will be using but they require more than the obvious.

4.3 Initial API Admin configuration.

If this is the first time you installed an AIM* product on your system, some initial configuration work needs to be done. This consists of executing the steps outlined next.

Start the **APIAdmin** tool from **Start/Programs/AIM*AT/API Admin**.

You will be prompted for a Server (in a drop down box) and if no server is listed you will have to configure one. For that click on the button the reads **Edit Server Connection**.

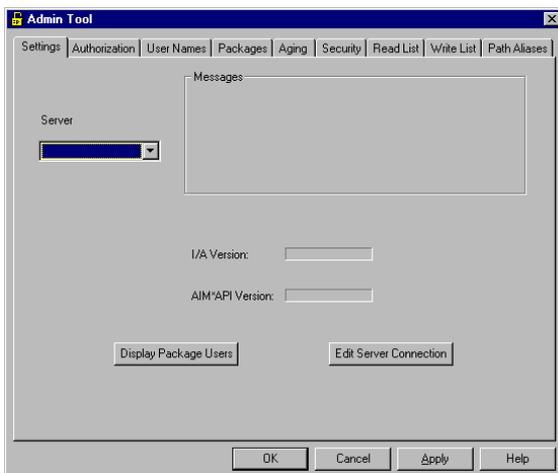


Figure 32Admin tool dialog

Create a Server alias (Edit Server Connection) if this is the first time logging in (note the values are just an example): The IP address you type in should of course match the Ethernet port that will be used to connect to that server. I.e: when you connect over the secondary Ethernet connection you type in the address of that Ethernet connection.

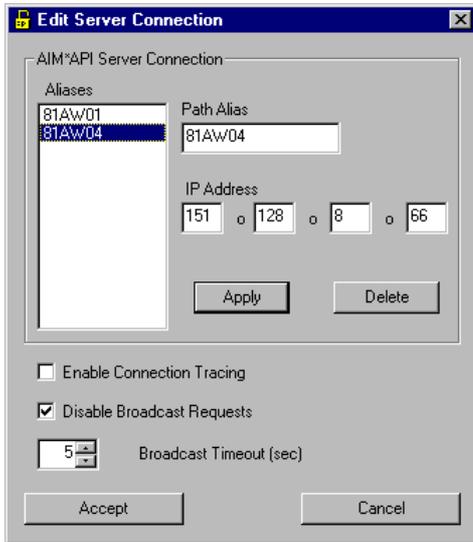


Figure 33 Create a Server connection

After completion of this step you must exit the **APIAdmin** program and re-open it. In the drop down box (**Server**) you now should have an entry for the AW that was just configured. When connection for the first time you will be prompted to set the password for this server. The next time you connect you will get a dialog as shown here:



Figure 34 Login to the api admin tool

Enter the password for the API Server you are connecting to.

4.4 Package licensing.

The **APIAdmin** tool is mainly used to enter the licenses for your applications in addition to setting the proper user accounts and privileges. There is some work for you here as well.

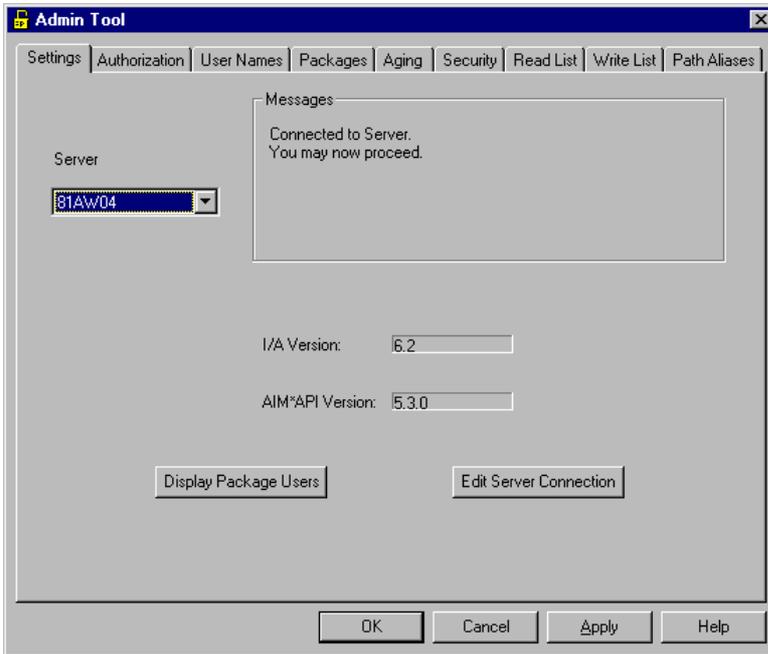


Figure 35A successful connection

In the example above an *Authorization* code was entered for the **OPCDAO** package. A user [**machine name**] must be added and this is shown here:(*The PC was named 81AW04 in this example*)

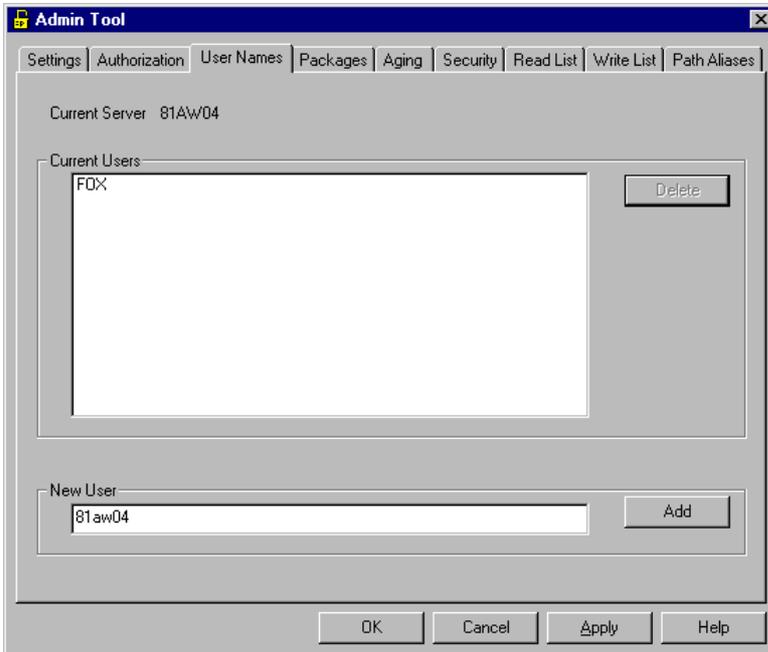


Figure 36 Adding the Machine name as user

Press **Add** when the name was completed which will bring you this result:

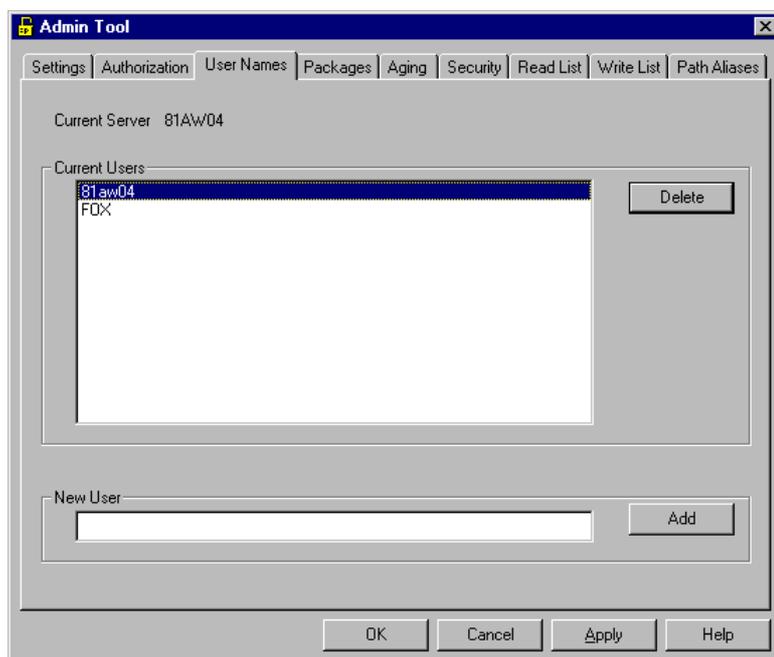


Figure 37 Machine 81aw04 is now added as user

A note for OFF Platform OPC Server installations:

- The Machine name of the *OPC Server* PC MUST be added as a *User name* on the I/A station that will accommodate the AIMAPI link into the I/A Series system. Note that only AP's and AW's are supported AIMAPI hosts within I/A Series systems.
- If you are using an **I/A 50 Series** machine to communicate with the OPC Server you must add the OPC Server's hostname as an AIMAPI user on the I/A 50 Series station using the **apiadmin** utility on that station. You must then get an OPC license for the UNIX machine as this is where the access to the I/A Series system is provided.
- Make sure that on the OPC Server side the AIMAPI configuration shows the path to the UNIX machine. This UNIX machine will hold the license information for your OPC Server application.

A typical example with two machines where:

- **AWRD01** is the Windows XP PC with OPCHDA in stalled and
- **AW51RD** is an AW51B where the OPCServer is communicating with i.e. Where AIMAPI in configured.

Assigning the software:

To assign the software installed to the specific users select the **packages** tab, add the packages: **AIM*API** and **OPCDA** (or **OPCHDA**) as shown and assign them to the newly created user account (machine name):

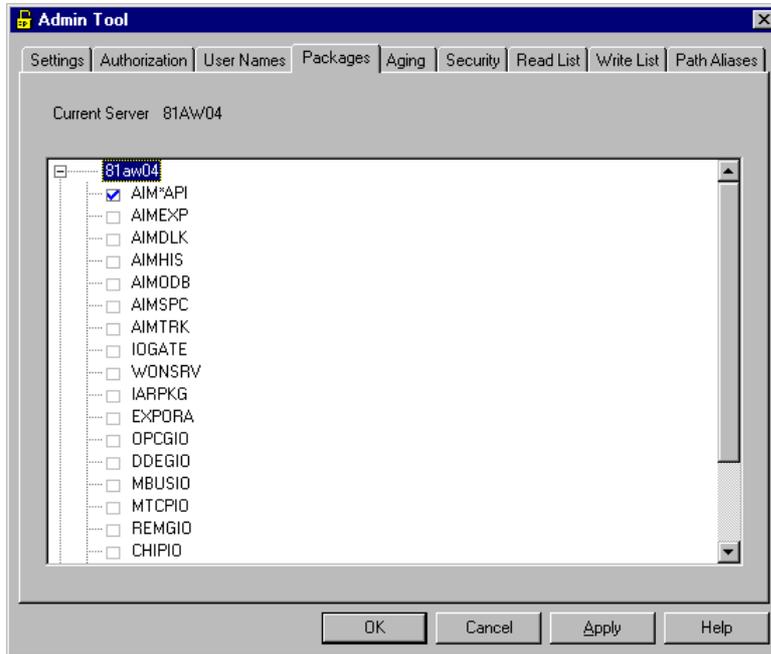


Figure 38 Assigned packages to this new user (machine name)

When you scroll down you must have a check mark next to the **OPCDA** or **OPCHDA** packages.

5 AIM*Historian and the AIM OPC I/O Gate client.

This sections explains (or at least tries to explain) how to setup the AIM*Historian with remote collectors and interfacing with OPC Servers from a third party through the OPC I/O Gate client.

NOTE: This chapter deals with an OFF PLATFORM AIM*Historian setup with Remote Collectors and the AIM*OPC I/O Gate.

In this setup we assume an Off-platform AIM* Historian although the setup on this side is no different from the standard. The AIM*OPC client will be installed on another PC platform where we will setup both the OPC I/O Gate and the remote collector (the mandatory installation). This chapter was written based on a specific customer site installation. The Windows platform was Windows 2000 Server for both the AIM*Historian and the remote collectors/OPC I/O Gate software. We start with the AIM Historian platform first

5.1 Setting up the AIM*Historian 3.2.

Here we follow the standard procedure of installing AIM*Historian.



Figure 39Run setup from the CD...

You will see the standard dialog to accept the license etc.

The package will be installed in the Program Files folder by default. Make other choices if you need to. In order to install AIM* Historian we need to install at least the Historian (duh) and the AIM*Utilities.

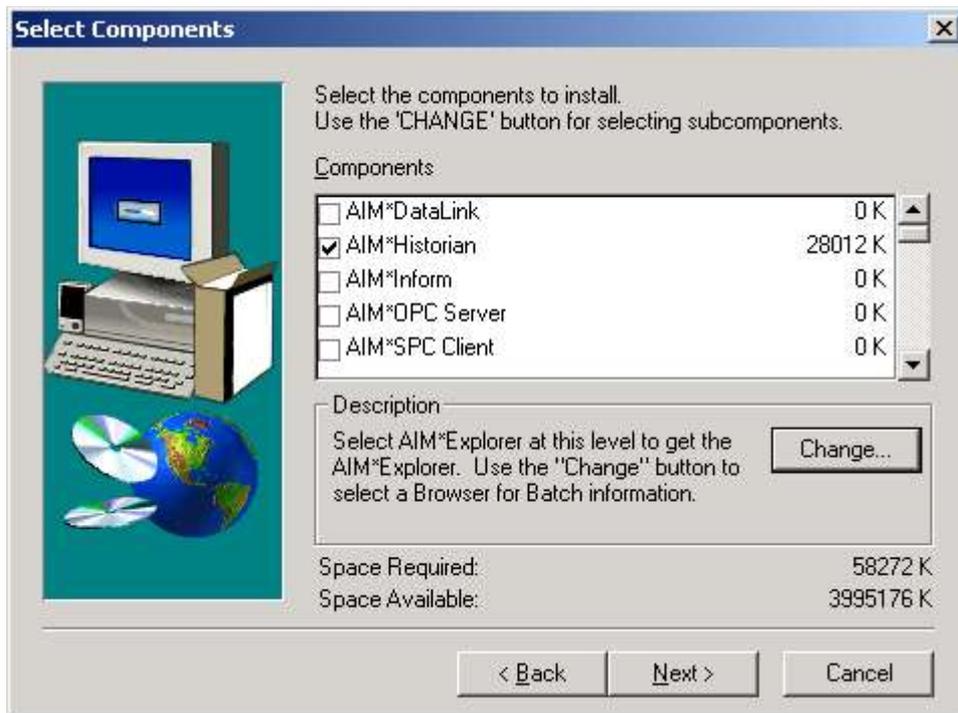


Figure 40 Select AIM*Historian and Utilities

6 Foxboro OPC I/O Gate (to Emerson PROVOX OPC Server).

For one of our customers we needed to interact with the Emerson OPC Server for the **Emerson Provox CHIPNT** platform. In this chapter you can find some of the things we came across to set this up.

6.1 Introduction.

Making the **Emerson CHIPNT OPC Server** work together with the **Foxboro OPC I/O Gate Client** is what this section is all about.

This is a little handout based on the following:

- A Foxboro AW70 OPC Client PC with I/A Version 6.2.1 and the OPC I/O Gate version 3.0.1 as part of a larger UNIX based I/A Series system.
The OPC station is AW7002 in this example is of course a newly introduced NT based station in that UNIX environment.
- A Emerson CHIPNT OPC Server on a remote location (Off-shore in this particular case).
The Emerson station is CHIPF15 (F15 is the Off-shore platform identification)

The customer provided us the CD-ROM that contained the OPC Server software that would enable us to install the OPC Server software (or parts of it) on the Foxboro machine. In this example I have taken the “simple” approach to run “setup” and see where that takes us.

6.2 Prerequisites.

Before you continue with this procedure the OPC client must be setup “as good as possible” which requires some actions to be taken. Taking a good look at the Foxboro OPC client documentation is a very good start. Make sure that all the steps as mentioned in the Foxboro OPC Client documentation are followed prior to following this procedure. In short this includes at least the following: On the Foxboro OPC client:

- Install the Emerson OPC PAS Server. This is handled in section Added the *Registry entry* from the OPC Server as mentioned in the documentation: [6.3 Installing the Emerson OPC Server on the client machine.](#)
- Loaded the **Matrikon OPC Browser** that comes with **AIM*OPC Server** from Foxboro as a second opinion and try to make connection this way. This is a very handy tool to see if you can browse a OPC Server without the need of I/A Series software.
- Created a little test DB with:
ECB99:
OPC, CHIPF15 and OPC.PROVOX as the required fields for this ECB.
OPC.PROVOX (in caps) is the exact name as it appears in the REG file entry.

6.2.1 Actions on the Emerson OPC CHIPNT Server:

- Added the user **Fox** with **gnomes** as the password (as shown in user documentation for the OPC I/O Gate)
- Added the **Fox** user to the users that would be allowed to access and launch the CHIP NT Server application. (just follow the documentation o this).

- Adjust the **DCOM** settings on the Server according the suggestion in the Foxboro doc and set *default security* to **anonymous** and *default Impersonation* to **none** decided to ADD the Fox account to the list of “allowed” users on the Emerson Server).

It is very important that the **OPC client** and server are able to “ping” each other for instance. Without this functionality there is not any chance you will get the connection for OPC up and running.

NOTE:

I’m not sure that all the steps are required to get the client to connect to the Emerson OPC Server but the time was not available to check all options. I fact I’m pretty sure these steps are NOT required and in the end it will prove that copying one or two DLL files will do the trick bu until then let us enjoy the show.

6.3 Installing the Emerson OPC Server on the client machine.

We are installing the OPC Server on the Client side. The Emerson OPC Server install program in time will suggest an either Server or client install.

For this example the entire “PAS” directory from the OPC Server installation CD was copied to a local disk on the AW7002 so the setup program could be run from the local harddisk on the Foxboro Client. To install the OPC software you must run the **setup** program found in the PAS directory on the Foxboro OPC client.

When running the **setup** program you will be prompted to have a look at the *readme* file before proceeding.

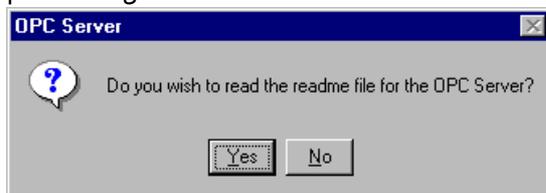


Figure 41 Read the readme file now?

Click **No** to go on if you know what you are doing. When you select **No** you proceed to the next screen.

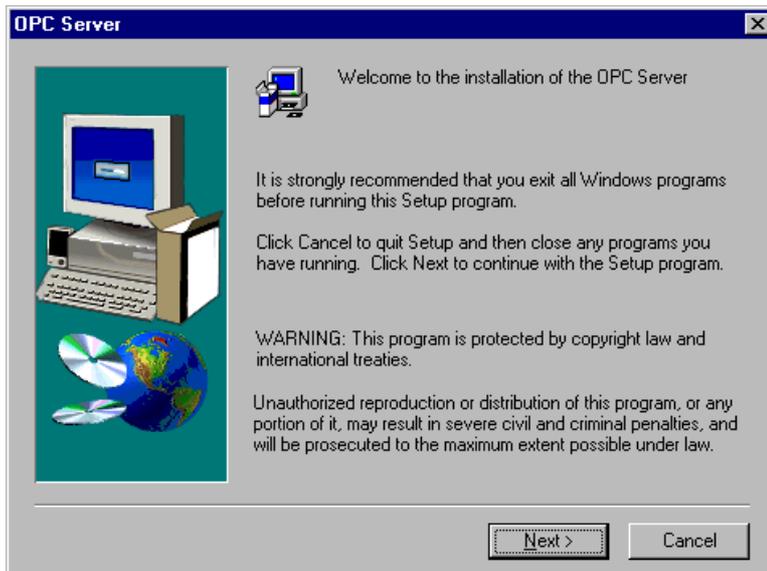


Figure 42 Start the installation...

Start the installation by selecting **N**ext.

Choose the defaults where applicable (see below):

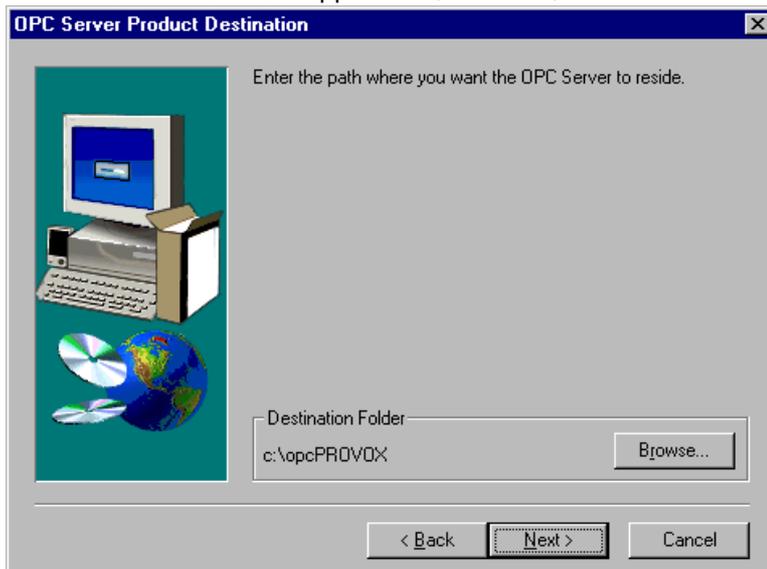


Figure 43 Choose the location for the OPC Server

Choose **N**ext to continue...

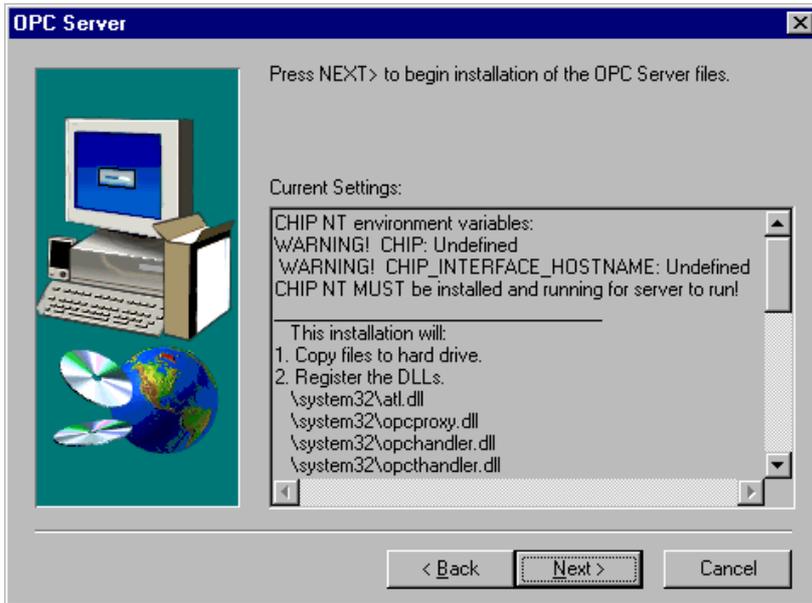


Figure 44 Review your selection and press the **Next>** button.

Continue with **Next** to resume the installation:

Next we will choose the option as shown in the dialog and we choose to install the client:

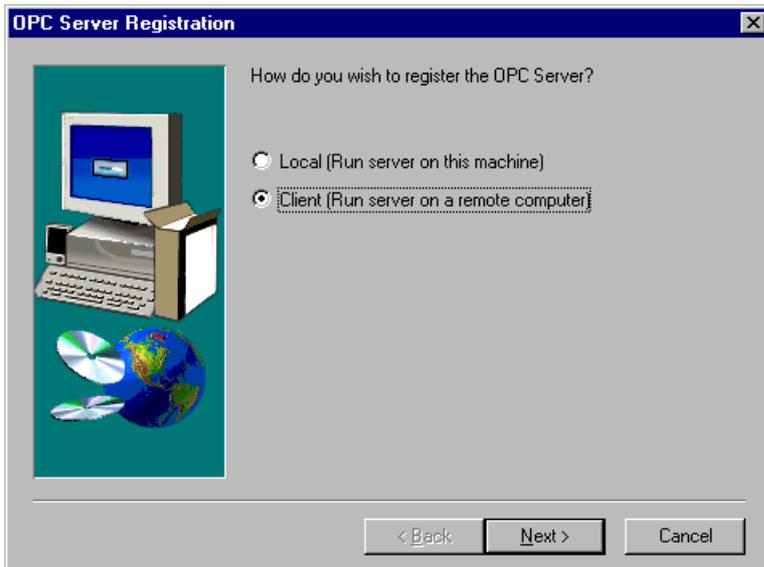


Figure 45 Select Client with Remote Server for this setup

You will be prompted a sexy looking DOS box where the OPC server to connect to must be specified. This is the name as found in the "hosts" table and this is the "network" name of the remote machine. In the example the remote OPC Server was "CHIPF15".

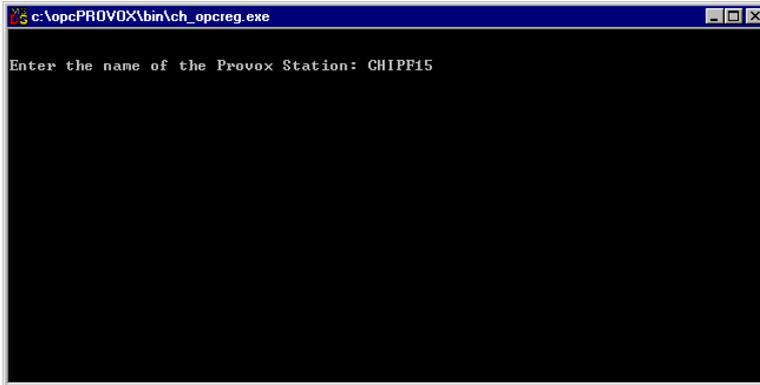


Figure 46 Enter the correct server name

After entering the correct name press ENTER to continue. You can connect to other similar PROVOX OPC Servers after entering the previous dialog and not only to the CHIPF15 as in the example.

When you need to connect to more than one OPC.PROVOX Server from one OPC I/O Gate client simultaneously, you need to have duplicate Registry keys (for example copy the OPC.PROVOX key to OPC.SRV1.PROVOX and OPC.SRV2.PROVOX. Use these names in the ECB99 in the **SVRAPP** field, to connect to the servers OPC.SRV1.PROVOX and OPC.SRV2.PROVOX. You should then have separate “trees” showing up in the OPC I/O Gate window listing the items for both servers.

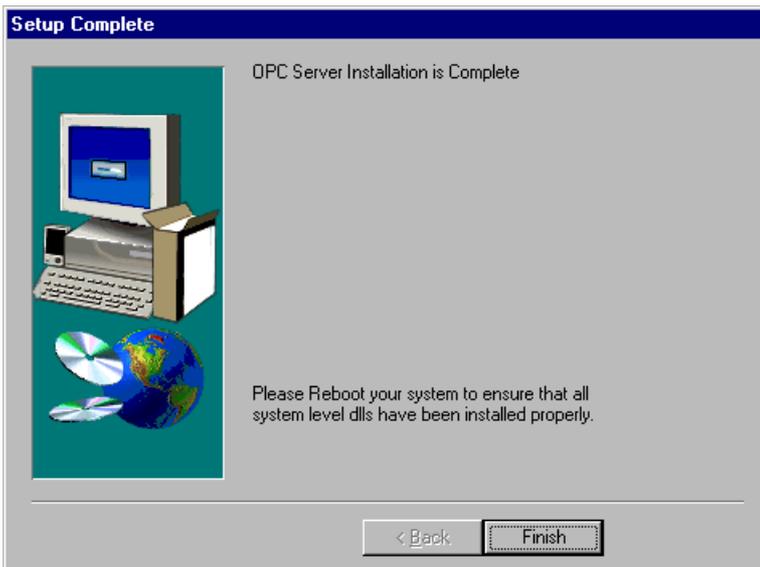


Figure 47 OPC Server installation completed.

This concludes the OPC Server/client setup for this AW and the Foxboro OPC Client requires a reboot now. After the reboot you should be able (make sure the I/O Gate is running) to get the configured ECB99 and ECB100 for this OPC Server on line and white.

7 Honeywell/Measurex OPC Server (to Foxboro OPC I/O Gate).

An OPC Server we encountered at a paper plant and a glasswool insulation plant. The OPC connection was used to establish a real time data collection link between **Foxboro I/A Series** and the **Honeywell/Measurex** system. The OPC Server found on both sites is the **HMX Rae Server**. A name that will encounter more than once in this chapter.

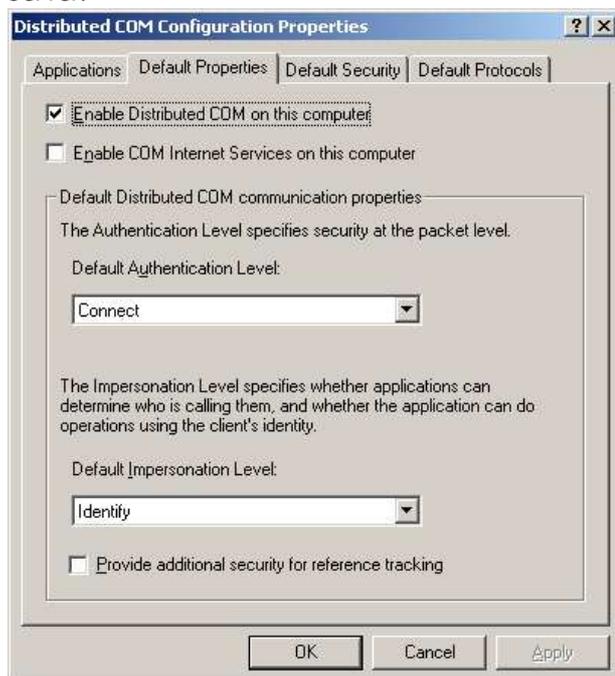
7.1 Configuration steps on the HMX RaeServer OPC Server.

This is some info how to allow the HMX Rae Server to interact with the Foxboro OPC I/O Gate client.

Some important steps as usual:

1. The I/A Series account should be introduced on the OPC Server: Fox/gnomes, not expiring yada yada.
2. We must go through the DCOM settings for the OPC Server. Some screenshots are provided that hopefully provide some additional information.

We have found that the **HMX RaeServer** does not use the default DCOM settings but instead uses it's own settings. Thus the default settings which are shown here are NOT used by the OPC Server:



Once again: The **“DEFAULT PROPERTIES”** are not used by the **HMX RAESERVER**.

We make sure the the TCP /IP protocol is in the list. It is OK to have multiple protocols here but since we are going to use TCP/IP this one is a requirement.

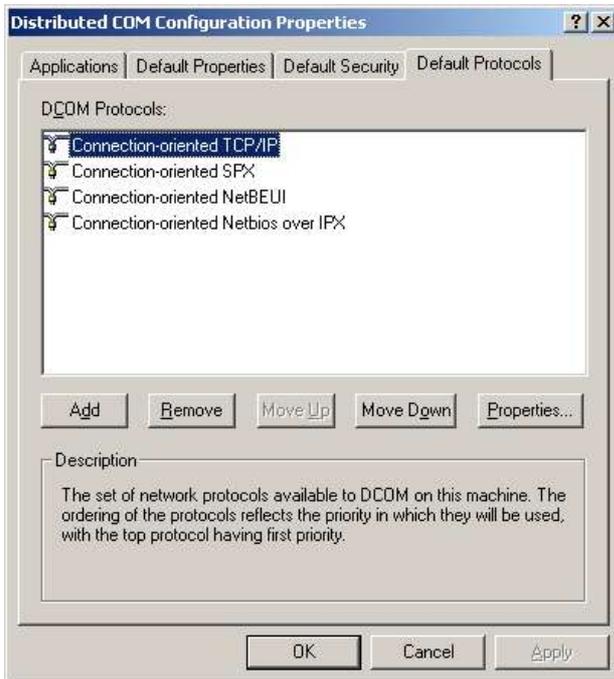


Figure 48 Available protocols for DCOM

To set the DCOM properties for the OPC Server we must select this application from the list:

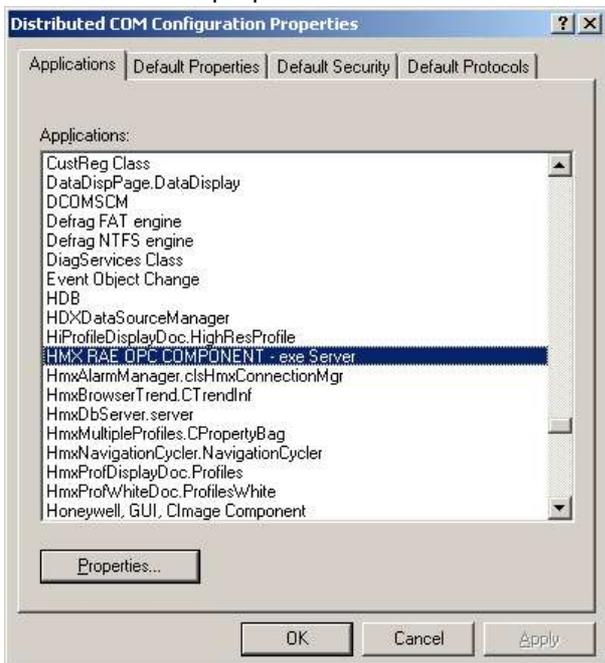


Figure 49 Select the HMX OPC Server to change DCOM settings

When you select “properties” here, you will get a screen as shown next:
For **Authentication Level** choose “**Connect**”:

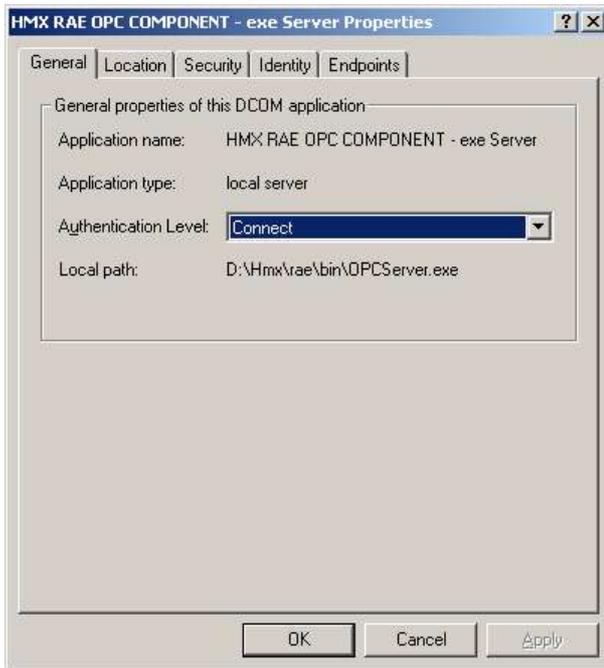


Figure 50 Select Connect as the Authentication Level

The next step is to define where the application is run. In the **Location** tab select to **Run the application on this computer**:

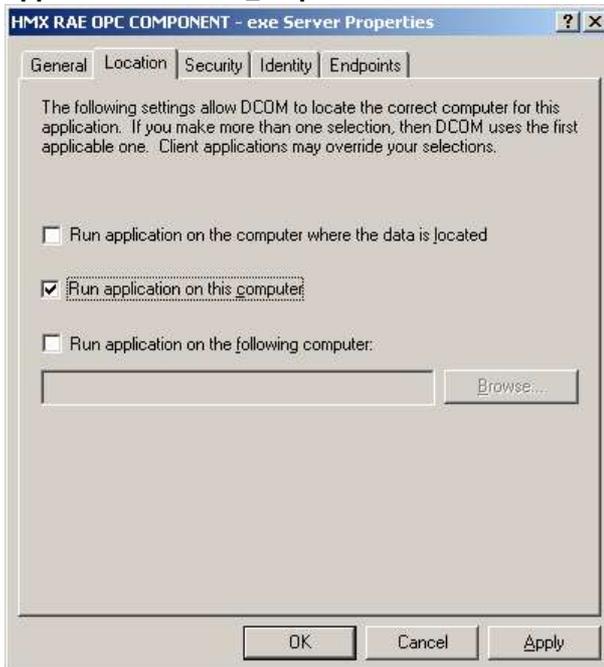


Figure 51 Run on this computer

Note: This differs from the default setting for this server!

The next tab is the **Security** tab where we must select the security settings. Make sure that the user **Fox** is added to the list of users that are allowed to **Access** and **Launch** the OPC Server application.

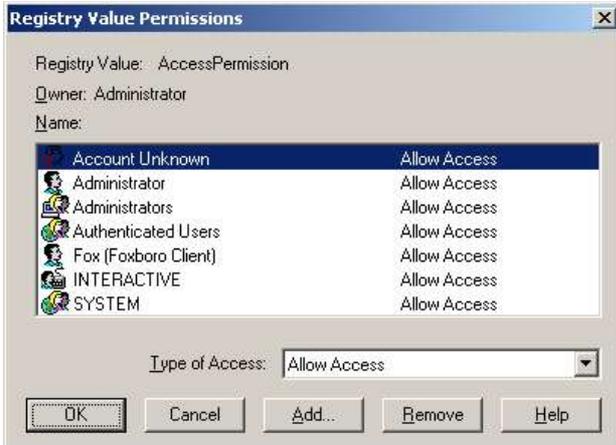


Figure 52 Add the user Fox to the Access Permission list

After adding the Fox account to the **AccessPermission** tab, the **LaunchPermissions** must have the same account: add the user **Fox** to the **Launching** users as shown below:

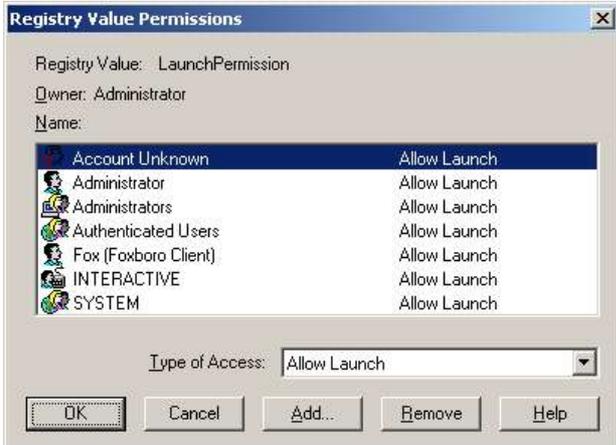


Figure 53 Add Fox to the Launching users



After all these modifications the OPC Server machine must be rebooted.

7.2 Configuration steps on the OPC Client (AW70)

After successful installation of the **OPC I/O Gate** package, it is time to configure the different layers in the system.

This includes:

1. Networking configuration.
2. DCOM settings.
3. OPC Server registration.
4. OPC Client setup options.

We have created an (in this case optional) *alias* for the OPC Server name since this name is rather long. These aliases are set/found in the `aliasnames.cfg` file as found in the directory :

d:\opt\diw\iogates\opc\cfg\.

The first field is the OPC Server name that you set in the **ECB99** on the Control database of this AW70, and the second field is the contents of the OPC Server registry value as it is found in the Windows Registry:

```
[aliasnames.cfg]
HMXOPC = "Hmx.RaeServer_0" < the letter "O" not the number zero "0"
```

As a result the **ECB99** contains the name HMXOPC as the SRVAPP name.

When you choose not to set an alias for the Server, you **MUST** use the full name (**HMX.RAESERVER_O**) in the ECB99 parameter value for SRVAPP.

7.3 Introducing the HMX RaeServer on the AW70.

These steps provide the necessary info to properly configure the OPC Server on our AW70 where the OPC Client runs. This makes sure that all DLL's and support files for the OPC connection to the (remote) server are correctly installed on the client side. On the install media (at least that I had) there are some files and notes that are specific for this install. A short list of the files that I received from **Honeywell/Measurex** (usually these files are provided on diskette with some instruction how to put everything in place):

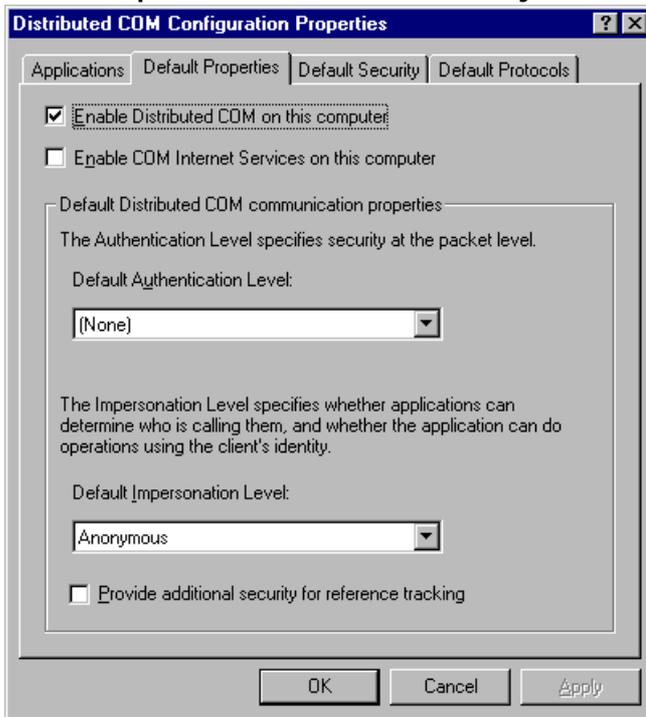
```
opccomn_ps.dll
opcenum.exe
opcproxy.dll
OPCServer.dll
OPCServer.exe (this one may not be on your set)
OpcServer.Reg
register.bat
```

The **Honeywell/Measurex** server takes these steps to properly configure the Server on our AW70 platform. (thanks Measurex for the assistance):

In this example we choose to install the required files on the D: drive in the system. The directory will be D:\hmx\rae\bin.

1. Create a directory D:\hmx\rae\bin on the AW70's D: drive.
2. Have all files listed above available on the AW70 and copy them to the location you have just created

3. From this location, run the “register.bat” program which will register some DLL’s and register the OPC server on the AW70.
4. Via the NT Control Panel add a variable `mxroot` with the value `d:\hmx`.
5. In DCOM (with **Start/Run.../dcomcnfg**) set the default properties to
Default Authentication Level “(None)”
Default Impersonation Level “Anonymous” as shown below:



6. Create an ECB99 with the correct settings:
 - SVRIPC** OPC or OPCNR
 - STADDR** The OPC Server's IP address or hostname from the Windows “hosts” file.
192.168.2.10 or MYREMOTESERVER for example.
 - SVRAPP** HMXOPC (if this is the *alias* that was created) or HMX.RAESERVER_O if you choose not to.
7. Checkpoint the database and open System Management.
8. When the Client and Server are connected you should be able to get the ECB99 on-line and white in System Management.
9. After this create your ECB100 (or more than one).
 - NAME** When you create the ECB use a sensible name.
 - PAR_ID** The parent should be the name of the ECB99 you defined before.
 - VAL_ID** Any arbitrary name you can come up with. Note that this is the OPC Group name that will be created for you in the server space.
10. Checkpoint the database and open System Management.
11. When the Client and Server are connected you should be able to get the ECB99 on-line and white in System Management.

After successful completion you can start to build your application database.

Foxboro I/A Series OPC Solutions

Installation, Configuration, Troubleshooting



8 ABB rtdb OPC Client (to Foxboro AIM*OPC Server).

Connecting the ABB OPC Client to the Foxboro AIMOPC server.

8.1 Introduction.

Setting up an OPC link between the ABB RTDB OPC Client and the Foxboro AIMOPC Server.

The OPC Server tested was version 1.2.0.7

The ABB RTDB OPC Client was version 2.1

8.2 Requirements.

This example assumes that the Foxboro OPC Server was installed and configured correctly with the default as set forth in chapter [3 The Foxboro AIM*OPC Server on Windows NT](#). Modifications to these default are noted here when applicable.

8.3 To do's on the Server.

Some things you must better always do on the OPC Server.

8.3.1 Create the user account.

The user account by which the ABB Client will connect must be added to the OPC Server. This is a normal Windows Administrative task and the account must be added with the Windows “user manager” application. We have found that this user must be a member of the Administrator group.

8.3.2 DCOM Settings:

Oh yes, the DCOM settings.

We can leave some security in place if we want to...

Open the **dcomcnfg** application and make these modifications:

Default Authentication: None

Default Impersonation Level: Identify

and make sure that the OPC Server uses these defaults!

Under the *Default Security tab* add the OPC enter the OPC Client user to the *Default Access Permissions* and *Default Launch Permission* dialoges.

You must select **Add/Show Users** to allow see the users configured.

8.4 To do's on the Client.

The OPC Client requires only the AIMOPC registry key to find the Server correctly.

There are no known other issues on the ABB Client side.

9 The Foxboro FDSI OPC Driver for FBM232 and FBM233.

With I/A version 8.0 Foxboro has developed a new concept of integrating devices into the I/A series system. This new way of integration is referred to as “FDSI”, short for “Foreign Device System Integrator”. Instead of using a module on the CP-level as integrator (Gateways and FD-Integrators), a set of four 200-series FBM's have been introduced (FBM230/231/232/233) that can be used in four ways of connecting to foreign devices: Serial Single, Serial Redundant, Ethernet Single and Ethernet Redundant. The FBM's run Windows CE as operating system and can be loaded with any protocol-driver required. Foxboro has released a set of basic drivers at introduction of I/A V8.0 (OPC, Modbus and AB-Contrologix), but for any other protocol a driver could be developed to be used in a FDSI-FBM, and for some projects around the world custom drivers have already been build.

This chapter will cover the use of the FBM232 and FBM233 in combination with the OPC-Client driver. It contains a lot of information about the FBM, but not everything. Be sure to also read the related documents!!

9.1 What you should know.

The reader of this chapter should be generally familiar with the I/A Series system, preferably with I/A 8.0, although this is not critical. Terms like Compound, Block, (F)CP, ECB, SMDH (System Management) 200-series FBM and the use of the things these terms stand for should not hold any major secrets. It is also recommended to read the chapters on OPC in this document.

9.2 Related documents.

1. FoxDoc 8.0 B0700BH: FDSI Driver – OPC Client's user guide
2. FoxDoc 8.0 B0800AH: Field Device System Integrators (FBM230/231/232/233) Users Guide

9.3 General information about the OPC FDSI FBM.

9.3.1 Requirements.

The basic requirements to get the FDSI-OPC FBM working:

- An I/A 8.0 (or higher) system with at least 1 [FZ]CP270 and at least 1 FBM232 or 1 or 2 FBM233
- The OPC I/O Driver (FDSI Drivers are not installed on a default I/A installation)
- The FDSI Configurator Software.
- An OPC Server with OPC Items to connect to.
- Network equipment to connect the FBM(s) and server(s), like UTP-cables and Hubs/Switches.

Also recommended is:

- The FDSI OPC Diagnostic Client tool.

9.3.2 Difference between the Single and Redundant FBM.

The FBM and the drivers have been developed in such a way that using a single or redundant FBM does not require any more or less configuration effort or different software. The only important difference is the ECB-block needed for the FBM. In case of the single configuration (FBM232) an ECB200 is used, and for the redundant (FBM233) configuration an ECB202 must be configured. This difference will show up in the functions in SMDH (System Management) and of course the available information on the ECB-blocks.

The two modules in a redundant configuration should be seated on the baseplate directly next to each other, starting on an uneven position. (1+2, 3+4, 5+6 etc). The 2 FBM's need only 1 ECB block, and are regarded as 1 logical FBM. Since the letterbug naming convention of 200-series FBM's dictates that the last 2 characters determine the position of the FBM on the baseplates, this is slightly different for a redundant FBM since it occupies 2 slots. For a redundant FBM, the first 2 slots on a baseplate are called "A", the slots 3+4 are called "B" etc. The left FBM is regarded to as the "**Main**" FBM, the right one is the "**Backup**" FBM. Although both FBM's communicate with the foreign device, only one of these two FBM's transmits the I/O to the CP, and this means that each FBM acts in certain "Role". The FBM communicating with the CP is called the "**Master**" FBM and the other, "standing by" is the "**Tracker**". So, depending on its position on the baseplate, an FBM is the Main or Backup, and depending on the state of the other FBM or the foreign device, both their roles can be Master or Tracker. (But not the same role at the same time of course).

9.3.3 Difference with other 200-series FBM's

The design and implementation of the FDSI FBM is significantly different from the Hardwired-I/O FBM's, different enough to mention it here in order to avoid wrong expectations and more important, wrong assumptions and conclusions during operation.

The FDSI runs Windows CE as operating system and contains several layers of software to bring the foreign data into the I/A C:B-environment. Because of the flexibility of the drivers and the "PC"-like nature of the FBM, it is more complex than Hardwired-I/O FBM's and needs more configuration than just an ECB and a "Go Online". It is better to compare the FDSI-OPC-FBM to the OPC-IOgate on a WinXP - PC that is squeezed into an FBM, but without a graphics card and monitor to interface to it. This FBM has flash-memory where user-files are stored (downloaded) and where also the driver resides. Startup of this FBM is also significantly slower than a Hardwired-I/O FBM, because it needs to boot like a PDA does, start the OPC Client, connect to a server and start requesting OPC-Items. In general, more patience is in order, and more diagnostic is needed than with a conventional FBM.

9.3.4 Limitations

- The OPC Client Driver does not support OPC 1.0 Servers
- In contrast to the OPC I/O Gate, a single FBM OPC Client can connect to only 1 server at a time. The two FBM's in a redundant setup can connect to two different IP-adresses, but the server-software must be the same as well as all points in that server.
- The FBM can handle a maximum of 2000 OPC items.
- OPC Array data types are not supported

9.4 Step by Step procedure: Redundant OPC FBM to Matrikon Simulation Server

This chapter will go through the steps to get a redundant FBM-pair connected to a Matrikon Simulation Server. As mentioned before, the configuration between a single FBM or a redundant pair is not very different and configuring a FBM232 by using this guide should be no problem. Where necessary, the differences will be explained.

It is assumed that an I/A system is running with at least 1 CP270 and that this CP is connected to a baseplate with a FBM233 pair (or single FBM232). It is also assumed that a PC is available with the Matrikon Simulation Server installed. (Free download from the Matrikon Website, www.matrikon.com.) Connecting the FBM('s) to the server using FTP cables and a hub/switch or a cross-UTP-cable should also pose no problems and is not discussed in this document.

The CP's letterbug in this example is OP6200, the FBM is OP620A. (A single FBM would be called OP6201)

The FBM('s) should be plugged into their baseplate and both the red and green leds should light up after a while. (Offline)

9.4.1 The FDSI Configuration tool.

9.4.1.1 Purpose of the FDSI Configuration tool.

Normally, the properties of a FBM that are configurable by the user can be set on the block that is associated with the FBM, being the ECB-block. With the FDSI-FBM (as well as some other), there is extra configuration needed that cannot be filled in at block-parameters, and therefore separate files must be created that are loaded into the FBM.

The FDSI configuration tool is used to create these files. They are XML-files, which are normal text-files and can be read and created by any text-editor. The XML-format is automatically created by this tool, and is therefore at least convenient to create the initial file-structure. After that, the files can easily be modified by any text editor, but off course also with the FDSI-Configurator. Because of the easy text-format, on large projects, these files can even be automatically created using database tools like MS Excel or Access. The FDSI tool has another functionality: It checks for problems in both the Port Configuration and the Device Configuration. If you create the XML-files other means than the FDSI-configurator, you can open those files with the configurator to check for mistakes and consistency.

9.4.1.2 Installing the FDSI Configuration tool.

The tool can be installed on any windows-XP machine, it is not necessary to do this on a (the) AW, however this is sometimes convenient because this saves you from having to copy the created files to the AW.

The tool comes on CD and can be installed using the normal windows setup procedure, using setup.exe. The configurator needs the Microsoft .NET framework and will install this if not already present. (The installation of .NET can take a while)

The following screens will show up:

The tool can be installed on any windows-XP machine, it is not necessary to do this on a (the) AW, however this is sometimes convenient because this saves you from having to copy the created files to the AW.

The tool comes on CD and can be installed using the normal windows setup procedure, using setup.exe. The configurator needs the Microsoft .NET framework and will install this if not already present. (The installation of .NET can take a while)

The following screens will show up:



Figure 54 Installing .NET

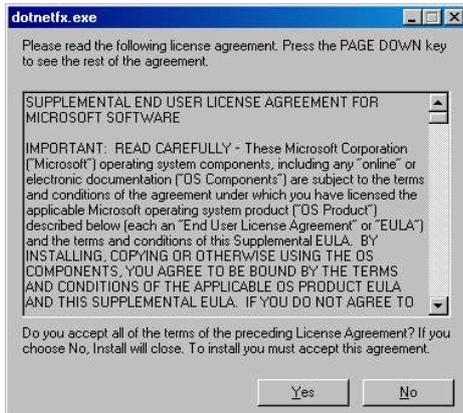


Figure 55 Always agree with MS

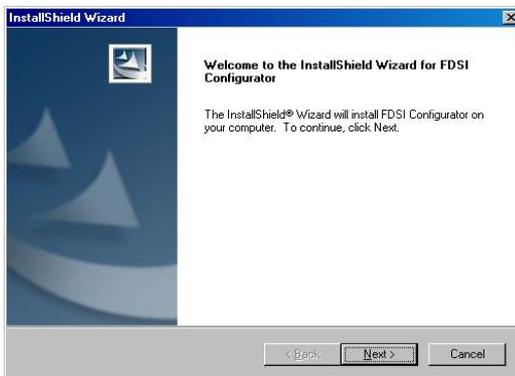


Figure 56 Installing the FDSI configurator itself

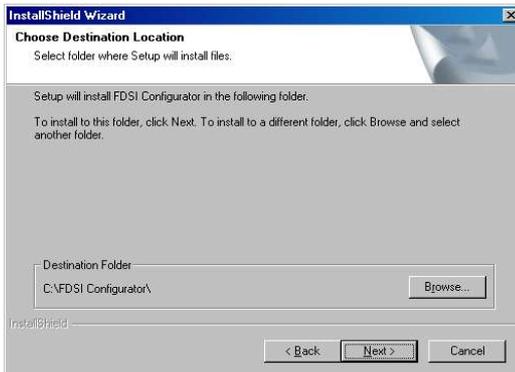


Figure 57 Select a installation directory. You might want to change the default!!

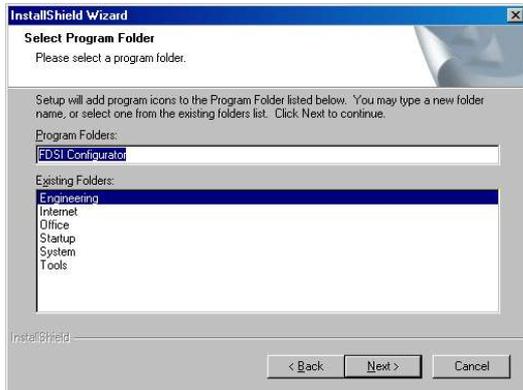


Figure 58 Start menu folder selection

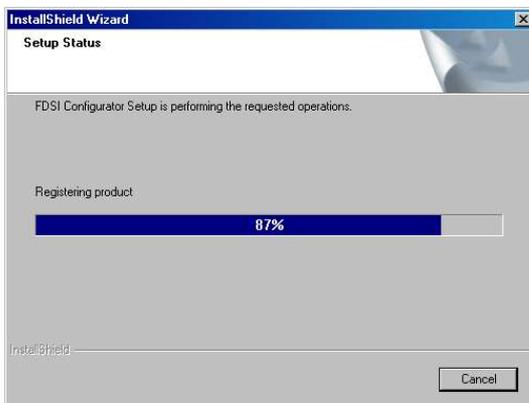


Figure 59 Almost done



Figure 60 Restart your PC (or AW through FoxView)

9.4.2 Installing the FDSI OPC OPC Client driver.

The FDSI OPC Driver comprises a set of executables and dll's that should, eventually, end up in the FBM. The OPC driver files normally come packed as a single file called "OPC.ziph" that should be placed on the AW host of the CP where the FBM is connected to. The directory is "/usr/fox/sp/files/devices", and this directory is used for all other configuration files associated with the FBM.

Depending on how the driver was shipped to you, it should always end up as file with the extension ".ziph" and should be placed in the mentioned directory.

The official CD-ROM containing the driver installation software should be installed on the host-AW and will place the driver, some example files and user documentation in the correct directories for you . After loading the CD in the CD-ROM drive (and possibly clicking setup.exe) you will get the screen as shown below:



Figure 61 FDSI OPC Driver installShield wizard .

To install the driver click **Next >** twice and eventually a dialog will inform the user of a successful installation.

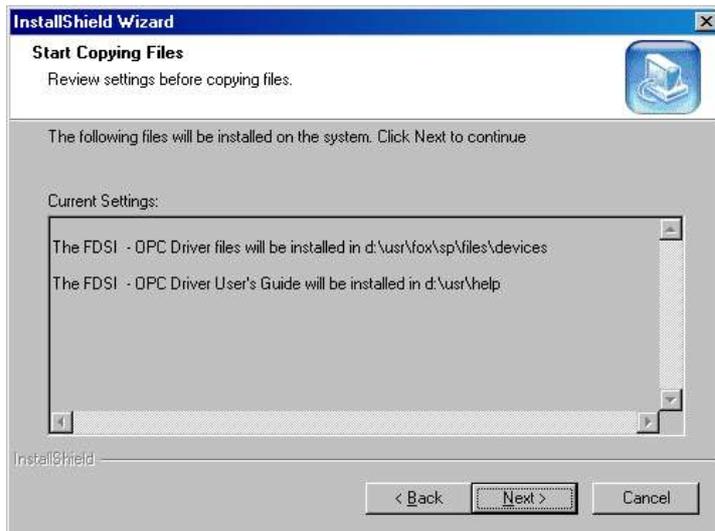


Figure 62 Directory information

The OPC Driver is now installed on your I/A AW Station. Later in this document it is explained how to load the driver in the FBM.

9.4.3 Configuration needed on the server side.

The configuration of the server can be difficult. Often, finding the right DCOM-settings to allow the client access to the OPC server can take a while. DCOM also tends to cache accepted clients, so some changes that were needed to get access suddenly make no difference once the connection is succeeded. In this example, we focus on the FBM, so we chose the easiest way to get the connection running.

In this example, we used the Matrikon Simulation Server. It was installed on an AW70 (Windows XP with SP1).

The FDSI OPC Users guide recommends these DCOM settings on the server:

Default Authentication level : None

Default Impersonation level : Identify

In my setup, I used: (Please refer to the other Server-Setup chapters in this document to find screenshots and detailed information where these settings should be made (dcomcnfg))

Default Authentication level : None

Default Impersonation level : Anonymous

Further more, I added "EveryOne" to the Com Security Access and Launch permissions, and selected "The interactive User" in the Identify Tab of the DCOM settings of the Matrikon

Simulation Server.

9.4.4 The Port Configuration File.

9.4.4.1 What is the port configuration?

The port configuration file defines the basic connection settings, like IP-addresses, OPC Server name etc. It does not have anything to do with the opc-items, this is solely a matter for the device-ECB and device-configuration file. The Port-configuration file is associated with the Port-ECB, the ECB200 (single) or ECB202 (redundant).

9.4.4.2 Creating a Port Configuration file.

Start the FDSI configurator. The first dialog will be:

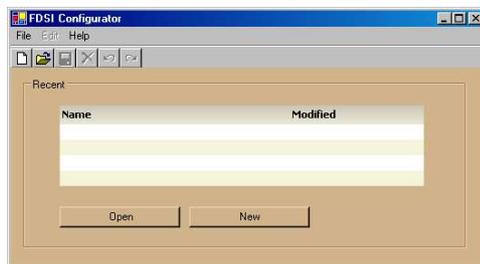


Figure 64 FDSI Configurator

Click “New”.

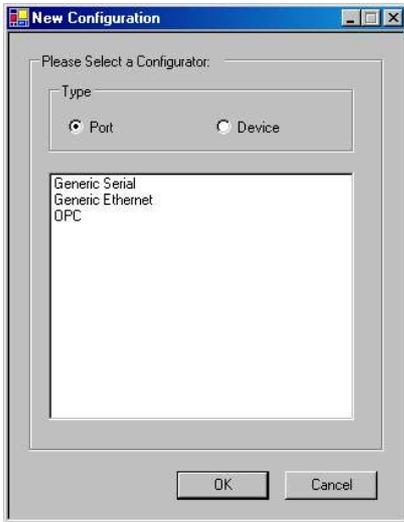


Figure 65 Config File Type selection

This dialog needs you to select which configuration-file you want to create. In our case, we start with the so called “Port” file which defines the basic information for the FBM to connect to the OPC Server. The FDSI-type in to select in our case is “OPC”.

The next dialog needs you to fill in all connection configurations for your FBM(s).

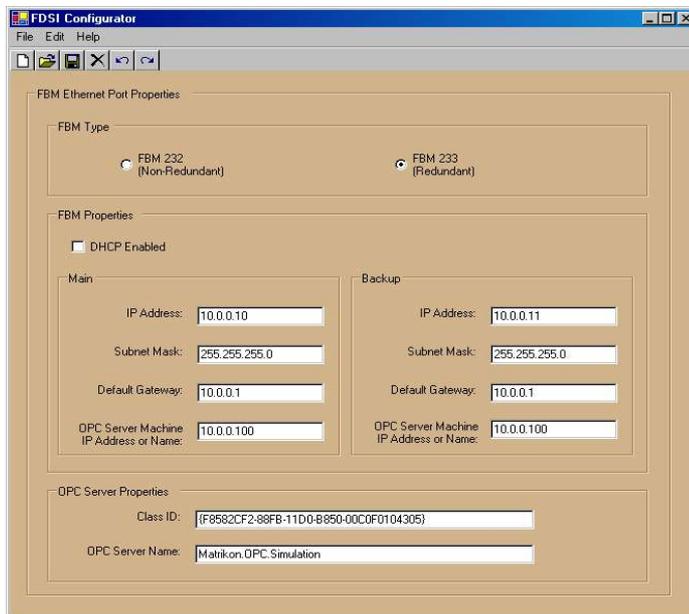


Figure 66 OPC Port Configuration

FBM Type: You can select here if you want to use a FBM232 or a redundant pair of FBM233's. In our example we will use the redundant configuration. In case of the 232-option, you cannot fill in the backup-FBM information and thus becomes only easier.

DHCP Enabled: If you select this option, the ethernet-card driver of the FBM hopes to get an IP-address from a DHCP server that should be connected to the same network as the FBM, and you cannot specify the IP information manually here. Although this is a nice feature, in most applications it is easier and faster to use a fixed IP address, because you have full control over the FBM and you know which FBM should have which IP address beforehand, which makes troubleshooting much easier. In our example we will therefore specify fixed addresses.

IP Address: Type in an IP address for the Main only when using a 232, or for both Main and Backup if using a redundant FBM. The IP addresses for a redundant pair should of course not be the same! Depending on the way the FBM's are used in a redundant setup (one or two OPC servers, one or more networks, one or more network-cards), the subnet might also differ for the Main and backup. In our case, we have 1 server with 1 network-card and the 2 addresses will therefore be 10.0.0.10 and .11 (because the Server will have .100 as address. .10, .11 and .100 are just "random" numbers, we also could have used 1,2,3 or 81, 33 and 245.)

Subnet Mask: Type in the subnetmask. This of course depends on the complexity of the network. In this case, we will have a simple OPC network between the 2 clients (Main and Backup FBM) and the server connected by a hub, and the subnets of the Server will be exactly the same so we use 255.255.255.0.

Default Gateway: This field is required by the FDSI Configurator, even if you do not use an actual gateway. The FBM will not use this gateway unless the IP address of the server is outside the subnet of the FBM's IP address, however we fill in an IP address that is not used to avoid any problems later on. (.1 address is not used in my test network)

OPC Server Class ID: The Class Identifier is a complex string that uniquely identifies every COM-object on a PC. An OPC Server also uses a CLSID and an OPC Client needs to know this CLSID. For most clients, this is done by registering the server on the client-pc in its registry. We cannot do this with the OPC FBM, therefore you need to specify the server CLSID here in the port config file. There are several ways of finding the CLSID of a server, for instance using the Registry Editor or using the FoxOPC-Diagnostic tool, which is explained later in this document. Pick any method you like.

Regedit is started though Start/run and typing "regedit". It might be difficult to find in the registry. (See the registry path in the status bar of this screenshot for guidance)

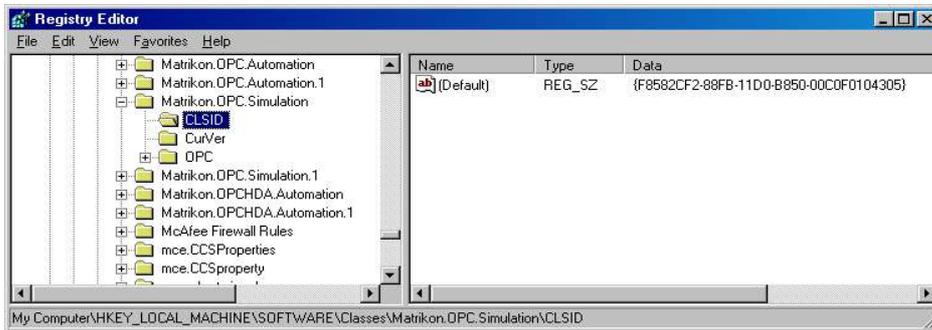


Figure 67 Finding the CLSID of the server in the Registry Editor

Another easy way is using the OPC Diagnostic tool. There is a separate chapter on this.

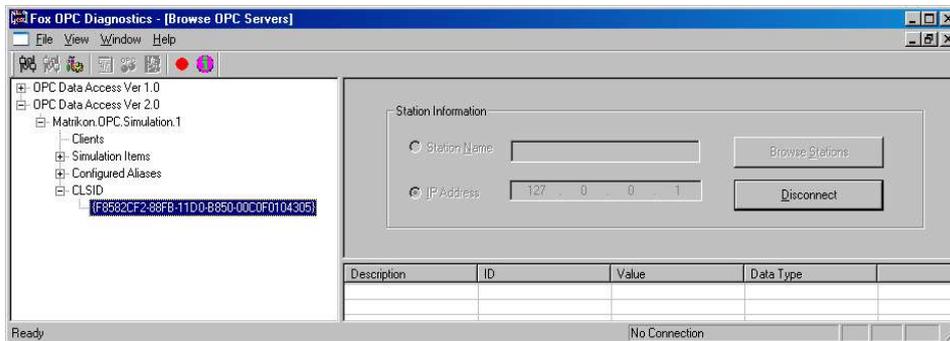


Figure 68 Finding the CLSID of the server using the OPC Diagnostic tool

Now type in the CLSID in the OPC Server Properties section ***{including the brackets}***.

OPC Server Name: The OPC server name may be any name you like, but it's obvious to use the actual OPC servers' name, in this case “Matrikon.OPC.Simulation”.

You can now save this file by clicking the save-icon (or file-save). If you are on the AW, you can directly save this file in the <D:/usr/fox/sp/files/devices> directory. If you save this file somewhere else, remember to copy this file to the mentioned directory later! The name of this file is important because you need it later when you configure the ECB. You could call it anything you like, but giving it a short name that identifies it with the FBM/ECB it's going to be used for seems to be a good idea. I have called it “OP620A_port.XML”.

The configurator has now produced a XML text-file with the following contents:

```
<FBM233 Assembly="OPCPortLib.dll"
Type="OPCPortLib.OPCPortCtrl">
  <Port>
    <Main>
      <IPAddress>10.0.0.10</IPAddress>
      <SubnetMask>255.255.255.0</SubnetMask>
      <DefaultGateway>10.0.0.1</DefaultGateway>
      <OPCIPAddress>10.0.0.100</OPCIPAddress>
    </Main>
    <Backup>
      <IPAddress>10.0.0.11</IPAddress>
      <SubnetMask>255.255.255.0</SubnetMask>
      <DefaultGateway>10.0.0.1</DefaultGateway>
      <OPCIPAddress>10.0.0.100</OPCIPAddress>
    </Backup>
    <OPCServerName>Matrikon.OPC.Simulation</OPCServerName>
    <OPCClassID>{F8582CF3-88FB-11D0-B850-00C0F0104305}
  </OPCClassID>
</Port>
</FBM233>
```

9.4.5 The Device Configuration File(s)

9.4.5.1 What is the Device Configuration?

As with the OPC-Io gate, at least one device (which is a child-ECB of the Port-ECB) must be created to communicate OPC-items with the server. The question of how many devices must be created is almost philosophical, but it depends mostly on the presence of actual external devices that are communicated through the OPC Server. This will enable watchdog monitoring, system-alarming and communication enabling/disabling per actual Device. If these things are not required, a single device is recommended, since it will decrease the amount of work for configuration, operation, fault finding and maintenance to a minimum. In this example we will therefore use only one device-ECB. (ECB201)

The Device configuration is also complemented with a file created with the FDSI configuration tool. This file is NOT mandatory but it is recommended.

The device configuration file has the following functions:

- Configuring the Watchdog point for the device. The watchdog Point is used to let system-management generate an alarm when the connection with the OPC Server is lost. If the watchdog point is not specified (either by not filling in, or by not using the file at all), system management can only detect total failure of the connection (cable removed or server shut-down). The watchdog point can be any point of any type (boolean, integer, real), as long as it keeps changing periodically. If such a point is not available in the server, Foxboro recommends to create a changing value on the I/A side, write this to an OPC-item in the server and read it back as a watchdog. This watchdog configuration is the reason that the device configuration file is always recommended, even if the next items in this list are not needed.
- Configuring OPC Item Aliases. As with the OPC-IO-gate where alias files are also used, the block-parameters in I/A are limited to 32 characters. If the OPC-items in the server are longer, you need to configure an alias that is shorter. This file will link the alias to the actual opc-item.
- Configuring the OPC-groups and there update-period.

9.4.5.2 Creating a Device Configuration file.

Start the FDSI configurator. The first dialog will be:

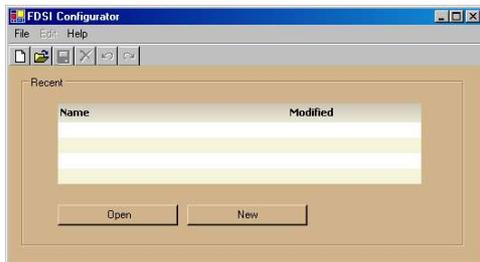


Figure 69 FDSI Configurator

Click “New”.



Figure 70 Config File Type selection

Select “Device” and “OPC” and click OK.

The device configuration dialog consists of three Tabs: General, Groups and I/O Points.

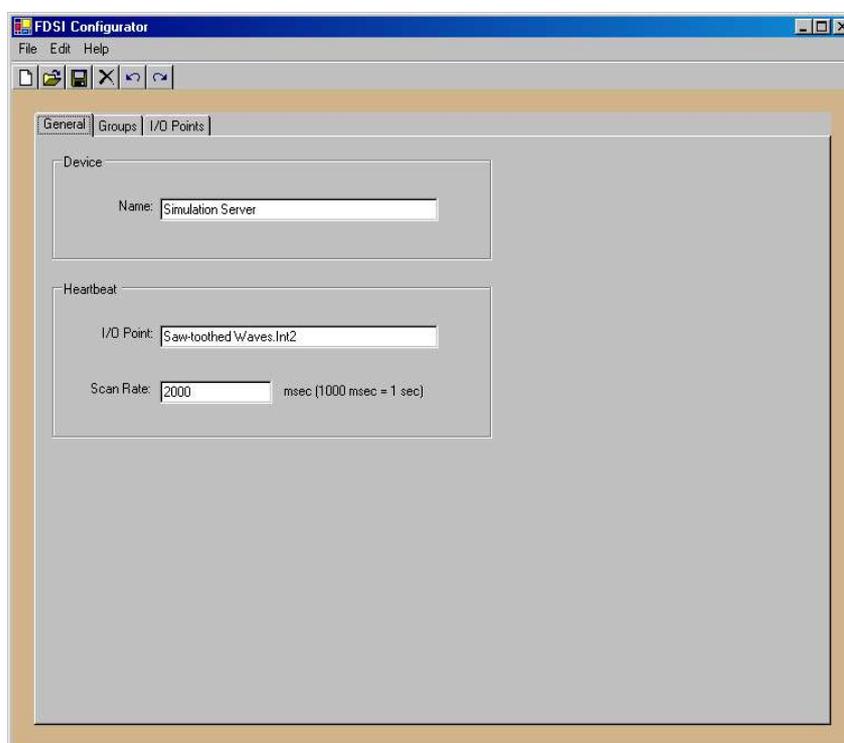


Figure 71 Device Configuration, General Tab

Device Name: This is the name of the device and can be anything you want, you won't see it ever again. (could be used for versioning the file for instance..). I just used "Simulation Server".

Heartbeat I/O Point: This is the OPC-item of the heartbeat point. This must be an continuously updating point in the server. For the Matrikon Simulation Server, this is easy because there are many points that change automatically all the time. (Thats why it's a simulation sever, right?). We use the point "Saw-toothed Waves.Int2".

Heartbeat Scan Rate: This is the period in milliseconds in which the OPC Driver checks to see whether the heatbeat has changed. If it has not for three consecutive periods, the Device ECB is set to Failed, all I/O will be OOS and a system alarm will be raised. The item we use in the server changes every second, so a scanrate of 2 seconds ensures a change in every period.

If you do not need Aliases, you can skip the Group- and I/O-points-tab.

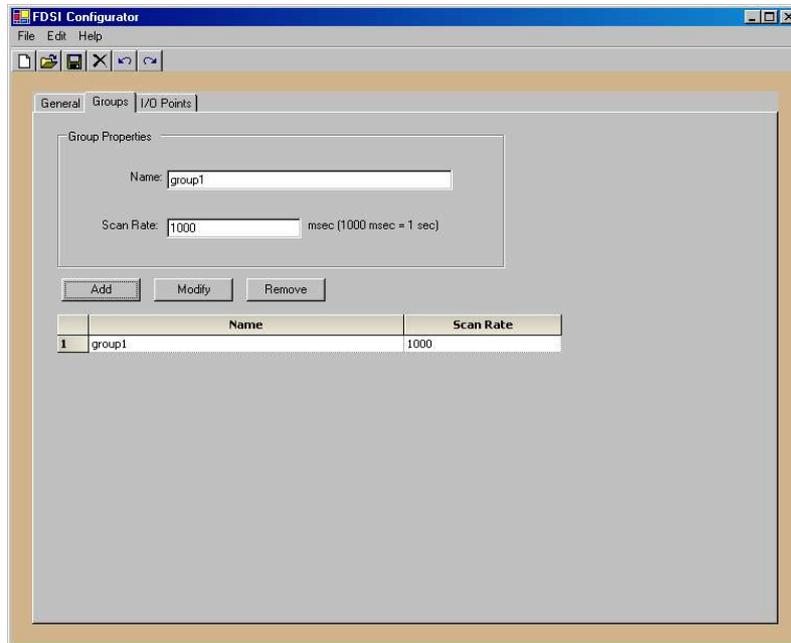


Figure 72Device Configuration, Groups Tab

Group Name: Here you can specify which OPC-groups you want for the points that need aliases and how to call them. The number of groups depends on the need for more than one scan-rate for the points, which is configured per OPC-group. If one scan-rate is sufficient, than the recommendation is again to keep it simple and use only one group for the alias-points.

Scan Rate: The scan rate for the group you typed in, in milliseconds.

Press “Add” button to add the group to the list.

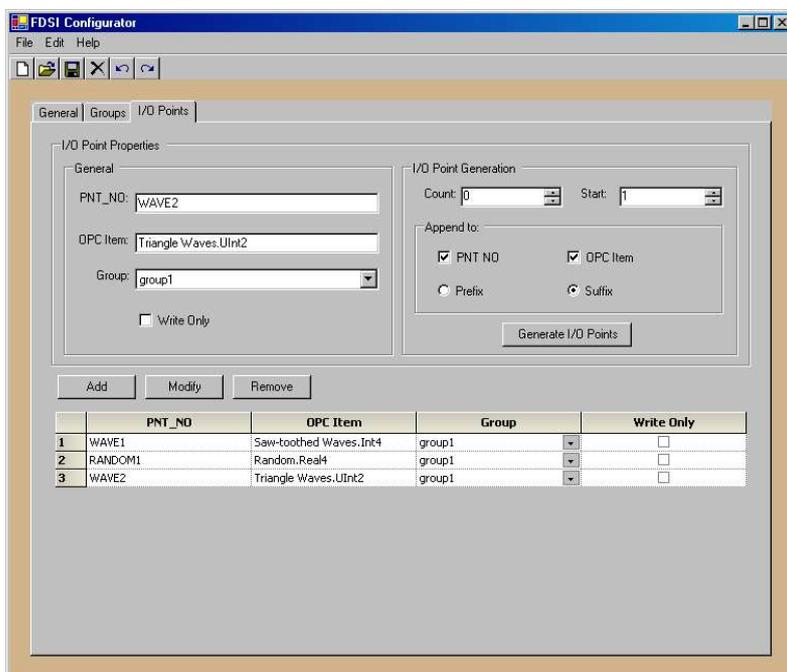


Figure 73 Device Configuration, I/O Points Tab

The I/O points tab enable you to add OPC items and their aliases that are to be used in the block configuration. It can also generate many I/O points and aliases automatically if they have an increasing number as prefix or suffix. As this is not often the case, we will leave the I/O Point Generation part of this dialog for you to figure out.

To add I/O points:

PNT_NO: This is the alias that will be used in the I/A DCI-blocks parameter “PNT_NO”.

OPC Item: This is the actual OPC-item as it is known in the OPC Server.

Group: This is the group you want to have the point in, and must be a selection of the groups you have created on the “groups” tab.

Press “Add” to add the point to the list.

In this example, we have added three points. Normally we would not use the aliases in this case, since the opc-items are all shorter than 32 characters, but this is just an example.

Exactly the same as with the Port-Configuration file, you can now save this file by clicking the save-icon (or file-save). If you are on the AW, you can directly save this file in the <D:/usr/fox/sp/files/devices> directory. If you save this file somewhere else, remember to copy this file to the mentioned directory later! The name of this file is important because you need it later when you configure the ECB. You could call it anything you like, but giving it a short name that identifies it with the FBM/ECB it's going to be used for seems to be a good idea, however it can off course not be the same name as the Port-config file. I used “OP620A_dev.XML”

The file that is now created looks like this:

```

<Device Assembly="OPCDeviceLib.dll"
Type="OPCDeviceLib.OPCDeviceCtrl" Name="Simulation Server">
  <Groups>
    <Group Name="HeartbeatGroup">
      <ScanRate>2000</ScanRate>
      <Points>
        <Point Name="Saw-toothed Waves.Int2" />
      </Points>
    </Group>
    <Group Name="group1">
      <ScanRate>1000</ScanRate>
      <Points>
        <Point Name="WAVE1">
          <OPCItem>Saw-toothed Waves.Int4</OPCItem>
        </Point>
        <Point Name="RANDOM1">
          <OPCItem>Random.Real4</OPCItem>
        </Point>
        <Point Name="WAVE2">
          <OPCItem>Triangle Waves.UInt2</OPCItem>
        </Point>
      </Points>
    </Group>
  </Groups>
</Device>

```

9.4.6 Creating a Control Database

The database can now be created. This can be done with any Database configurator, such as ICC, IACC of Foxcae. In this example I use IACC, but it should be easy enough to extract the parameter-information and to use that with another configurator if required.

The database needs the following items:

- Port ECB (ECB200/ECB202)
- Device ECB(s) (ECB201)
- DCI-blocks for I/O.

9.4.6.1 The Port ECB (ECB200 or ECB202)

In IACC ECB's are automatically created when the FBM's are added to the system configuration. The FBM was created (OP620A) and an ECB202 by the name of "OP620A" was automatically created in the OP6200_ECB compound.

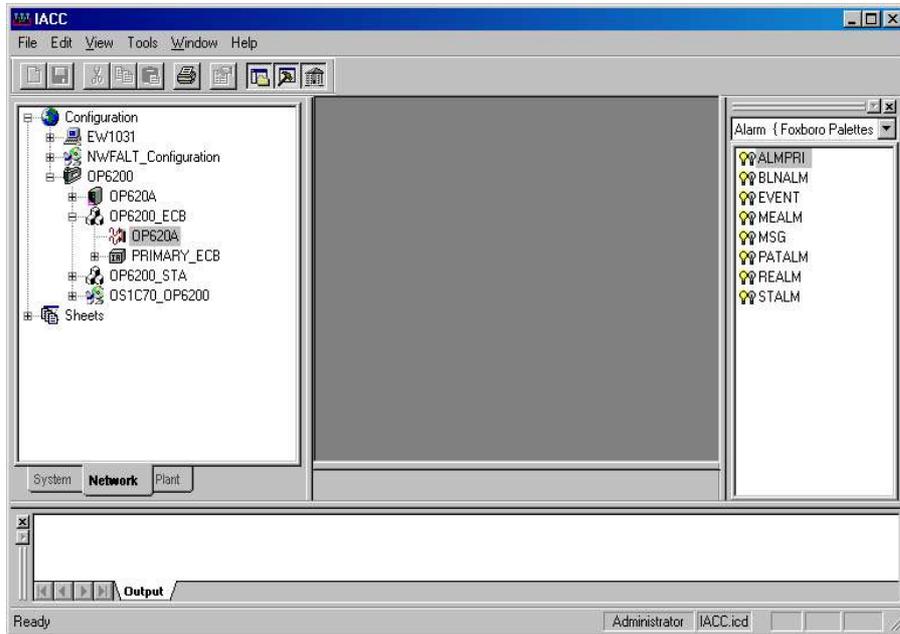


Figure 74 Creating an FBM233 in IACC, including ECB

The OP6200_ECB:OP620A block has the following configuration:

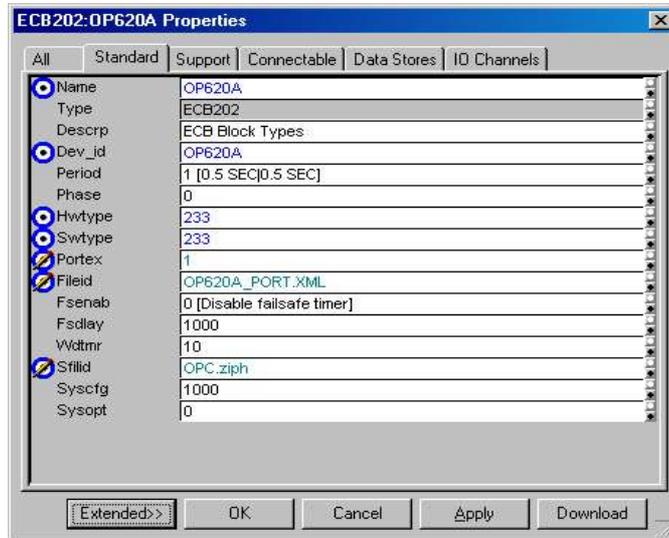


Figure 75 ECB202 Configuration.

The important parameters here are:

Dev_id: This ID defines the FBM to use, not the blockname. It is also needed in the Device-ECB later on. It's easiest to have the blockname the same as the device-id, which is also what IACC does. In our case both are "OP620A"

HwType: Must be "233" for redundant FBM and "232" for single.

SwType: Must be "233" for redundant FBM and "232" for single.

PortEx: This number defines the existence and configuration of the ports on the FBM. For

Ethernet FBM's this is always "1".

FileID: This must match a port-configuration file's name in /usr/fox/sp/files/devices of the host AW. The port-config file we made was "OP620A_PORT.XML". (This is not case sensitive, i.e. the filename does not need to be in capitals on the hard-drive)

SfileID: This is the Software File ID. This parameter must also match a filename in "/usr/fox/sp/files/decives", which where the Driver should be that is going to be used in the FBM. It must always be a file with extension ".zip". In this case, it's "OPC.ziph".

9.4.6.2 The Device ECB (ECB201)

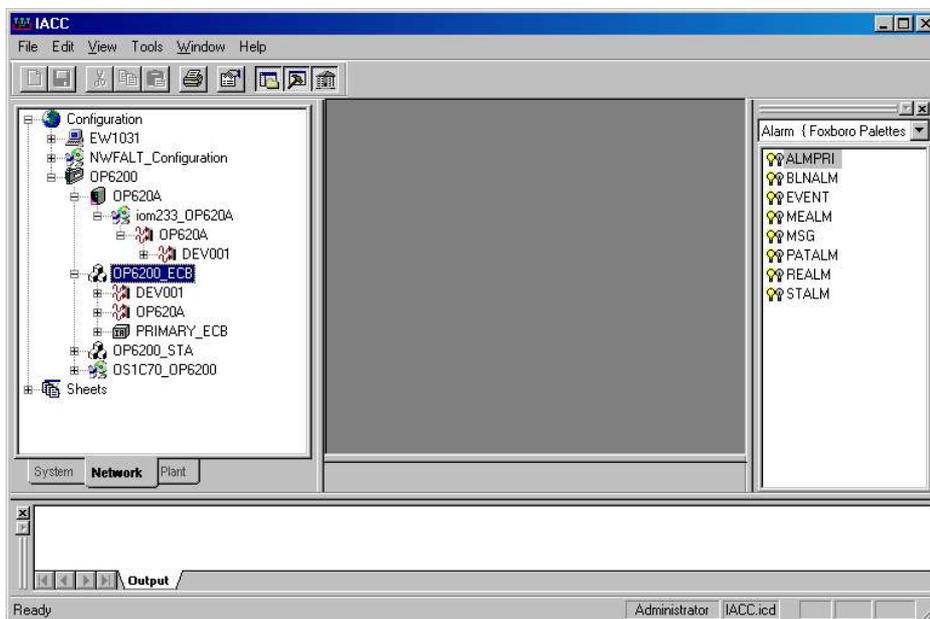


Figure 76 Added a child ECB, the Device (ECB201)

A child ECB must now be created. In IACC, this is done by right-clicking the ECB202/ECB200 (either under the FBM itself or in the ECB-compound) and selecting 'Add Child ECB'. The name is completely free, I called it 'DEV001'.

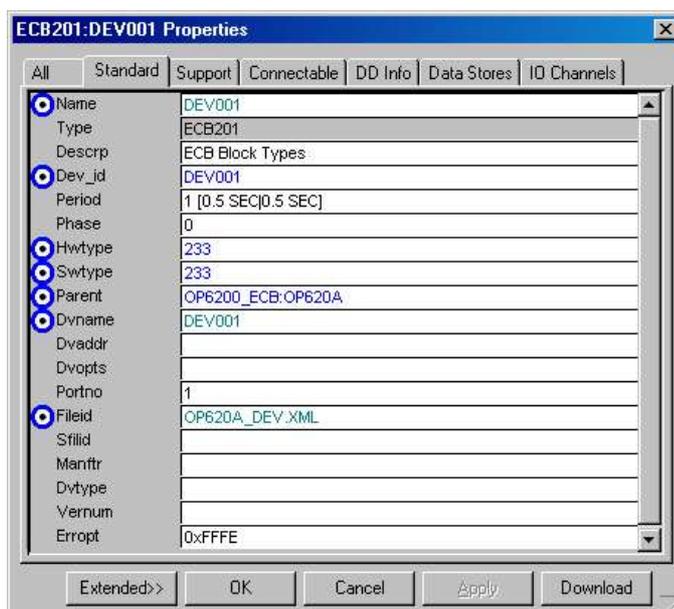


Figure 77ECB201 Configuration

The important ECB201 parameters are as follows:

Dev Id: This is the actual name of the ECB and must be used in the DCI blocks later on to refer to this ECB. Best practice is to keep the name of the block and the Dev_Id the same.

HWType: 233 for redundant FBM or 232 for simplex.

SWType: 233 for redundant FBM or 232 for simplex.

Parent: This refers to the parent ECB, the ECB202. IACC automatically fills this parameter including compound name, but only the blockname ("OP620A") of the 202 will do fine also.

DvName: This is the name of the device, and can be anything you want. The OPC Driver uses this string to create a OPC-groupname if you did not create a group in the Device Config file, or if you are not using one. Again, to keep things simple, you could keep it the same as the blockname and dev_id.

Portno: Default value is 1 and this should stay that way for FBM's with one port.

DvOpts: This parameter must be empty when you use a device config file, like we do here. If you do not use such a file, you need to fill in **"NOXML"** here. (without the quotes)

Fileid: If DvOpts is empty, you need to specify the filename of the device configuration file in "/usr/fox/sp/files/devices" here. In our case it's OP620A_DEV.XML.

Sfileid: Not used for ECB201!!

9.4.6.3 The DCI I/O blocks

The FDSI FBM uses the DCI blocks to get the value into I/A's Compound-Block world. This means you cannot directly use AIN or CIN blocks, you must use RIN, BIN, IIN, IOUT, ROUT etc. blocks. The functionality of the DCI blocks is roughly the same as FIIN, FRIN etc block of the OPC IO-gate.

For the OPC FBM, if you use an Device Configuration File, the OPC Driver will think all PNT_NO items that are configured in the DCI blocks are alias and tries to find these in the Device configuration file. You can prevent the driver from doing this by prefixing the OPC-item that has no alias and can be directly requested from the server with a “#” (aka hash- or Pound-sign).

In this example, we have three points in the alias-file (as the device-configuration-file is sometimes referred to). We will build these three blocks and a additional block to use the # option to see a value without alias.

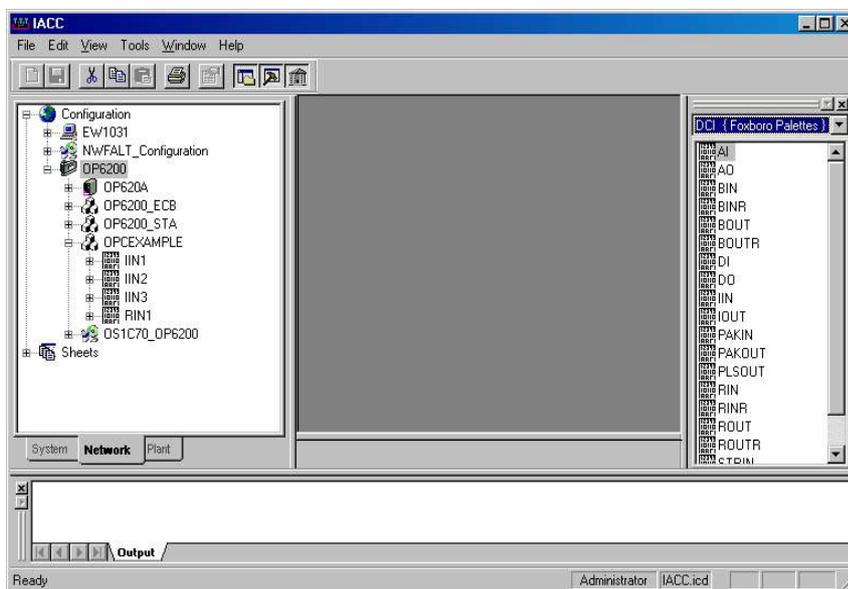


Figure 78DCI-block compound

In IACC a compound “OPCEXAMPLE” was created and four DCI-blocks were build, three IIN-blocks and one RIN-block. Their configurations are as follows, where the IOMID and the PNT_NO parameters are the only changes from the default values. Note that these correspond with the Device-Configuration file's aliases.

9.4.7 Loading the driver in the FBM.

As soon as the ECB202 exists, you can load the Driver in the FBM. The other blocks are not necessarily needed for this. Also, this is a task that is only needed with a new FBM or when you want to update or change the driver in this particular FBM.

Make sure you have the driver available in the /usr/fox/sp/files/devices directory as “OPC.ziph” (or any other .ziph file you have defined in the ECB200/202)

Start System Management, and navigate to the FBM.

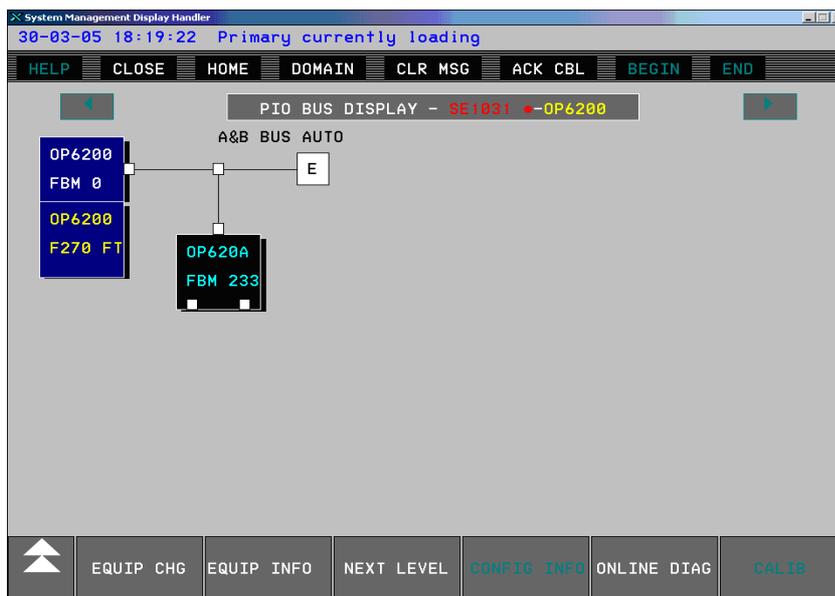


Figure 79 Find the FBM in system management.

Select the FBM and click 'EQUIP CHG'.

Before the Driver can be downloaded, the FBM must be online. Click “Go Online”. If an FBM233 is used, click “MAIN” to set only the main FBM online. The FBM should go online, and the red light on the FBM should go off.

To download the driver, click “EEPROM/SOFT DOWNLOAD”, and then click “SOFT_DOWNLOAD”.

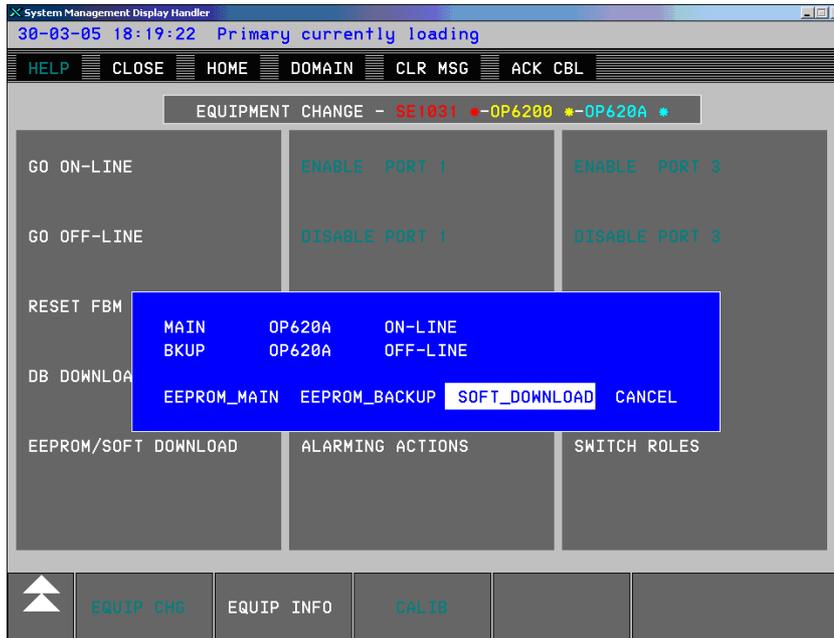


Figure 80 Start Driver Download

The OPC Driver will now be downloaded. When this is finished, the FBM will automatically reboot. This will also automatically happen for the Backup FBM if a redundant FBM233 is used. . After the download is finished and the FBM is rebooted, it will come back up On-Line.

To check the driver download, look at the second page of “Equip Info” of the FBM. This will show the “Driver Type” and the “Driver Version”. The driver type represents the build-name of the driver (not necessarily the same as the driver filename) and the driver version is a combined checksum of the driver-files. This checksum can be checked with the version-text file that came with the driver installation. (stored in /usr/fox/sp/files/devices)

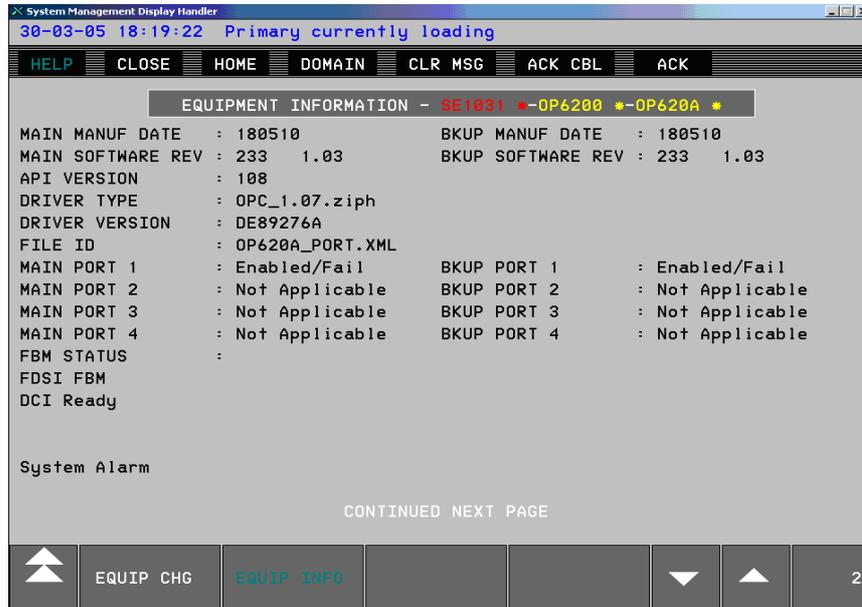


Figure 81 Driver Version information (FBM was already running in this screenshot)

NOTE: Whenever the driver is re-downloaded in a working FBM both Port and Device configuration files are deleted from the FBM's memory and must be re-downloaded.

9.4.8 Downloading the Port configuration file.

The port configuration file can now be downloaded. This is always necessary whenever the ECB200/202 shows “DCI Not Ready” in system management. This can be seen on page 2 of the Equipment Info in the FBM Status-field. This will show “DCI Ready” or “DCI Not Ready”, and this will correspond with a Yellow/White and Blue (OOS) FBM in system management. At this stage in our example, the FBM is blue and the Equipment Info page show “DCI Not ready”, telling us to download the Port Config File.



Figure 82 Download the Port Config File when the ECB/FBM is "DCI Not Ready" (Blue)

Click "DB Download" on the FBM (ECB200/202) Equipment Change page in system management. During this download, the equipment Info page will show "DB Download Active" in the FBM Status field, however this download will be so quick the download is normally finished when you reach the the Equipment Info page...

If the Download is successful and there are no errors in the Port Config file, the FBM should go Yellow in system management and the FBM status should go to "DCI Ready". This is very Important, DO NOT CONTINUE if this status is not reached. Also, the port should be enabled. Often this goes automatically but check to see whether the you can enable the port "ENABLE PORT 1" on the Equipment CHG page.

9.4.9 Downloading the Device Configuration File.

Next step will be getting the Device file in the FBM. In system management, Click on the FBM and select "Next Level" to look at the devices connected to the ECB200/202.



Figure 83“Next level” of the FBM shows the device(s)

Now click “Equipment Chg”.



Figure 84Click DB Download to download the Device Config File.

To download the device-configuration file, click “DB Download”. (This will be only pickable if the ECB200/202 is DCI ready and the communication of the device (ECB201) is not enabled)

The ECB201 also has a “DCI (not) Ready” state that will also go to “Ready” when the Device Configuration file is downloaded successfully. Also the “DB Download Active” status can be seen on the Equipment info page of the Device ECB201.

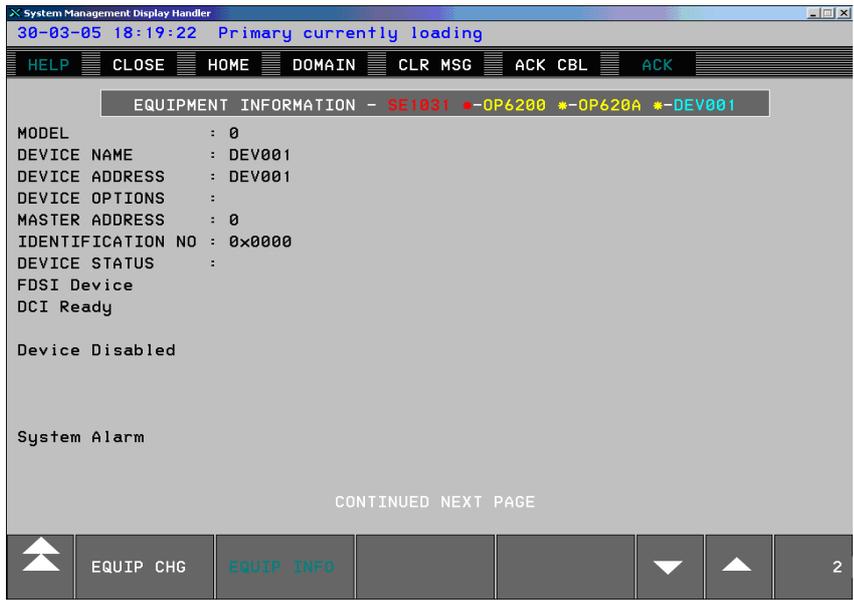


Figure 85 Device ECB201 is DCI Ready!

9.4.10 Starting communication

If the Device Config file is downloaded, the communication can be started by enabling the communication.



Figure 86 Enable Communications

The FBM should now start communicating with the server. If this is successful, the device-state will go from blue to yellow/white, Enabled and Not Failed.

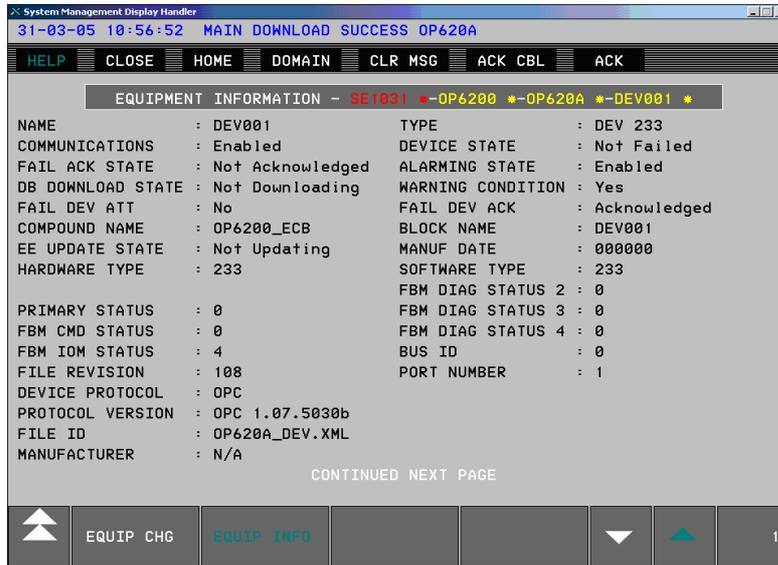


Figure 87 Device ECB201 to yellow state

The DCI-block should now not be OOS anymore and show values!

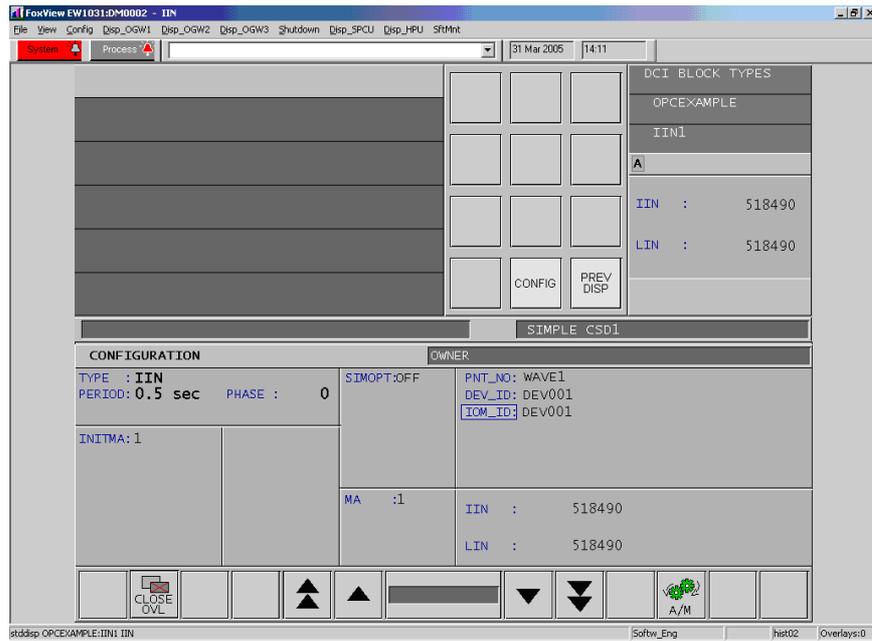


Figure 88 Values are coming from the Server

9.4.11 Backup FBM in redundant configuration.

To get a redundant FBM working, the best way is to get the MAIN completely up and running first, and then turn your attention to the BACKUP.

So if a redundant FBM is used, now would be a good time to set the backup FBM online. This FBM

will now be automatically updated with all configuration information from the master FBM. This will be visible in the ECB202 on the third Equipment Info page as “Trker Edu In Proc” which means “Tracker Education in Progress”. This can take a couple of minutes.



Figure 89 Master FBM is updating (educating) the Tracker FBM

When the education is finished, the backup FBM should start communicating with the server without other user intervention.

The FBM and device (ECB's 202/200 and 201) should now be white in system management.



Figure 90 FBM Healthy

9.4.12 Master/Tracker Role switching in redundant configuration.

Now the FBM is redundant, you should be able to test the fault-tolerance of the FBM by, for instance, disconnecting the cable from the Master FBM. Which FBM has what role can be seen on the FBM's themselves (Master/Tracker lights), in system management and on the ECB202/201 parameters (Detailed display). When the connection on the master FBM is lost, the driver will request a Role Switch and the Tracker FBM becomes master.

The roles can also be switched manually, by clicking “Role Switch” on the Equipment CHG page of the ECB202 in system management.

9.5 OPC Diagnostic Client

The FDSI FBM with OPC driver has a lot of diagnostic possibilities on the ECB's, however none of these inform you about the OPC Items and in depth information on the OPC connection. The OPC Diagnostic client can help.

The Client can be installed on any PC that is connected to the same network as the FBM. The server will therefore qualify always!

The installation of the client will speak for itself, this is not explained in this document.

9.5.1 Connecting the client to the FBM.

When the client is started, the following screen appears when “File -> Connect” or the left-most icon in the toolbar is clicked:

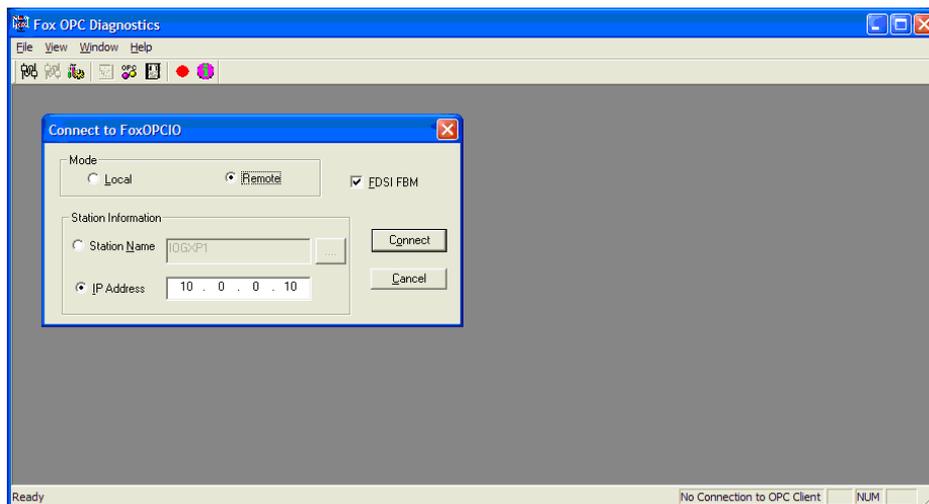


Figure 91 OPC Diagnostic Client with Connect Dialog

To connect to the FBM, select “Remote” for the mode, check the “EDSI FBM” checkbox, specify the FBM's IP address (or station name if present in host-table), and click “Connect”.

9.5.2 Diagnostic view

The diagnostic view that will appear now looks like this for our example configuration:

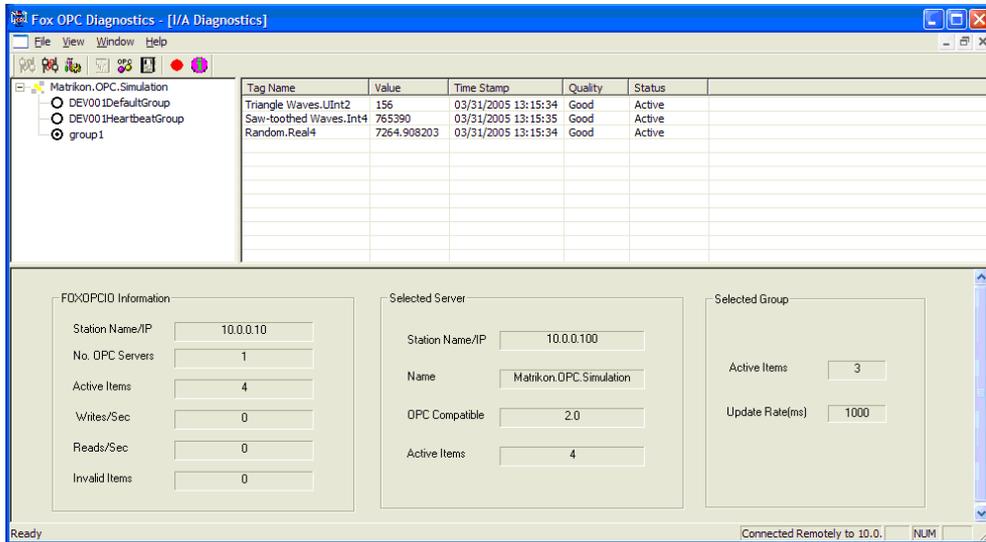


Figure 92 Diagnostic View

In the left pane, the connected server is shown, as well as the OPC-groups that are connected to the server. By selecting a radio-button of a group, the right pane will show all points connected, including the OPC-tag, current value (live), time stamp, quality and Status. The status is important, because it will tell you if there are any syntax errors in the OPC-Items.

The lower part of the diagnostic view will show general information, such as number of points active, and number of invalid items. (per group)

9.5.3 Log Files

The Diagnostic client will also generate log files with detailed information on the FBM and the OPC Driver in it, but only while it's connected to the FBM. These files are located in the “log” subdirectory of the OPC Diagnostic installation directory. The logfiles can be viewed using the “View Log” function in the “View” menu and the log functionality can be turned on/off in using the “Log Configuration” option in the “File” menu.

9.5.4 OPC Browser functionality

The OPC Diagnostic client also has the possibility to connect to a Server and show information like the CLSID, connected clients and the available points. To start the OPC-browser, use the “View” menu or the button on the toolbar.

Type in a station name or IP-address, and connect to the server. In the left pane, rightclick the 'OPC Data Access version 2' and click “Populate”.

The Browser will now connect the the server and add the found OPC Servers on the machine. Now, you can right-click the found OPC-Servers which will show the detailed information of this server.

This view can be showed at the same time as the connection-view of the FBM diagnostic. See screen shot below.

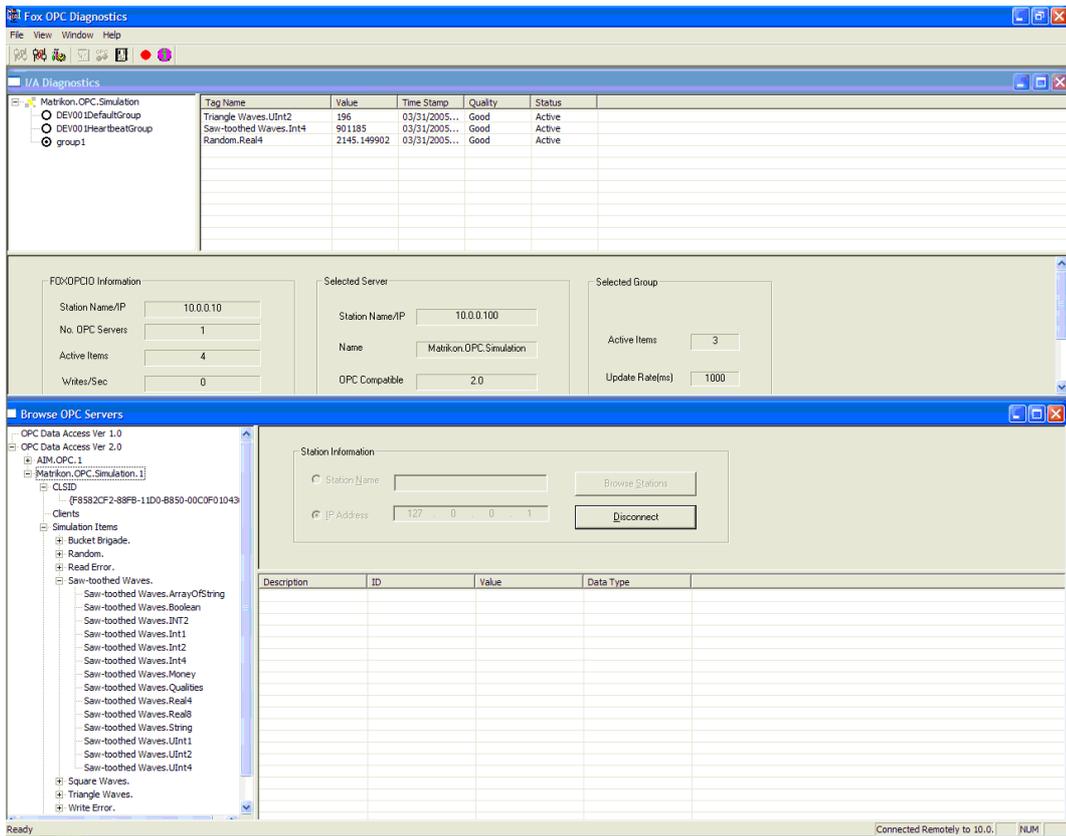


Figure 93 OPC Diagnostic of FBM and Server simultaneously

10 Troubleshooting / FAQ.

OPC without troubleshooting, that would be something! Until we get there we go here...

10.1 General tips for troubleshooting OPC.

- Use the **Matrikon OPC Explorer** as a reference OPC client see if you can reproduce the problem.
- Show **Objectmanager** settings with `/opt/aim/bin/show_params`.
- Check for API processes (`/opt/aim/bin/P`).
Output should look similar to this:
- More advanced, use `/opt/fox/bin/tools/som` for *om*-lists on the host AW (or `/opt/fox/bin/tools/rsom` for *om* list in remote stations like CP's).
- Try `/opt/aim/bin/apispy` with the menu functions "913" and "1" to see all connected api tags
- Try `/opt/aim/bin/apispy` then via function "916" then the index nr. (you got that from the previous step via the 913 option) to all api info for one tag including delta settings.
- Review explanation of the bad bits (UNIX only): `/opt/fox/hstorian/bin/histutil50.doc` (in middle of document).
=> bad bit is the 8th bit (first bit is nr 0: (the 9th bit from the right is the bad bit):
if error value is 0023 = 0000 0000 0010 0011 bad bit is not active.
if error value is 0123 = 0000 0001 0010 0011 bad bit is active.

When facing a system that is unresponsive after working fine for a long period (points going to BAD, data frozen in the OPC clients etc.) there are some steps you can take:

1. Try to stop the OPC Client (OPC I/O Gate etc.) and wait some time before bringing the client back on line. If a first attempt fails, retry and allow approx. 10 minutes before going on line again. 10 minutes is a general total DCOM timeout after which all DCOM processes are killed.
2. If that fails, Kill client and also stop the OPC Server. Let the system rest for a little while and then restart the OPC client application. The client should then automatically start the OPC Server.
3. For OFF platform OPC Server configurations it may be an idea to reboot that OPC Server machine.
4. Beware of this step. But when possible it may be worth to stop/start the AIMAPI interface program on the host. Take good notice of other applications that may be using this interface (AIMHist, AIMDataLink etc.).
5. If that fails, a more drastic measure may be in order: reboot both OPC Server and Client machines. Boot the Server first while client remains off-line. When Server machine is up and running, then (and not earlier) start the OPC Client.
6. Try to make sure that ALL OPC items you are addressing are indeed existing on either platform. OPC does not deal with non-existent items very well.
7. If that fails: **Panic or read on.**

10.2 Local Clients connect but remote clients can't.

Problem: Local Client can connect to the OPC server, but remote clients cannot.

Apart from the obvious problems that are account and DCOM related and which are covered in the previous chapters there can be something else.

Solutions: in Windows XP the enabled simple file sharing must be unchecked. This option is found under **Start/Programs/Accessories/Windows Explorer** and then choose:

Tools/Folder/Options/View, you will find the **Simple File sharing** option at the bottom of the list.

10.2.1 Unable to add groups.

Problem: When starting **OPC I/O gate** to a remote Windows XP AW70 from a local Windows XP AW70, the OPC I/O gate only shows the OPC server, but no groups (ECB100's) are shown.

A remote connection using the **Matrikon** OPC client also connects to the server, but attempting to add a GROUP results in the following error:

```
"AIM.OPC caused COM/OPC error 0x80070776 on IConnectionPoint.Advise()  
Error string: The object exporter specified was not found.  
Additional: Could not establish data callback connection on group 'test'"
```

Solution:

Several changes have been made and then this problem was solved, the exact cause is not clear, Problems were solved by checking and correcting the following items:

- Corrected the Gateway IP address on the OPC client.
- Corrected the Gateway IP address on the OPC server.
- Corrected errors in the Host table on the OPC server. (In this particular case the host name was available, but the IP address was incorrect).

10.2.2 Tags in OPC clients show *bad* status, even though the blocks on the server are healthy in I/A:

=> Possible solution: Quickfix that solves the problem described for IA 7.1 and 6.5.1 as:

"For applications using AIM*API or FOX*API, once initialization is completed, any action that triggers an update of a connected point will correct its status and value and causes it to function correctly thereafter.

The quickfixes are:

- QF1005113 for IA 7.1 on Unix
- QF1005118 for IA 6.5.1 on Unix
- QF1005127 for IA 6.5.1 for Windows XP
- QF1005128 for IA 7.1 for Windows XP

There may be another possible solution: the block in api does not have enough time to get connected therefore in the `aimapi.cfg` add the text: "CTDLAY=200" (50 is default)

CTDLAY is defined as: "A delay in hundreds of seconds in various stages of the connection process to ensure success".

10.2.3 While attempting connection: RPC server not found

While trying to setup a connection between the OPC client and OPC Server you receive an error stating: RPC server not found.

Some possible causes for this:

- No network connection available, check network cables and/or IP addresses etc.
- OPC server is not registered on this machine. The remote OPC Server must be registered in the OPC Client PC.

10.3 Foxboro OPC I/O Gate

The OPC I/O Gate related stuff.

10.3.1 The I/O Gate windows stays empty.

When the OPC I/O Gate is started nothing appears in the screen. No servers are detected. There can be numerous reasons for this, some of the most obvious:

- The network connection between the server and client is broken? Try to see if that is the case. (ping works?)
- If the network is OK, it may be *DCOM* related. Make sure that the User **Fox** is enabled on the Server. Check the *DCOM* settings for the **Fox** user.
- Check if *DCOM* application are set the use the defaults regarding security, and if not make sure that you check the custom settings for that application.
- Did you put the ECB99 and corresponding ECB100's on-line?
- Did you wait long enough? It takes time for *DCOM* to setup communications.
- If there is no network, fix that first.
- When using a Foxboro OPC Server: The OPCDAO/OPCHDA license has expired or was never entered.
- When getting data from an I/A Series system: Possibly the API server has stopped.
- The OPC Server machine name **MUST** be added to the APIAdmin user list and the OPC package should be linked to that user id.

10.3.2 Get error message: "Interface not registered".

After installing the **OPC I/O gate 3.0.2** on a **I/A 70 Series Windows XP** machine: During startup of the **I/O Gate** (where the ECB's remain **RED** and with the **Matrikon OPC Explorer** (which produces the following error message: "**Interface not registered**").

To solve this issue:

You should perform this task whenever you re-install the OPC I/O Gate version 3.0.2 on your Windows XP system or when you upgrade from 3.0.1 to 3.0.2.

To register the OPCproxy DLL once again:

To do this start a command window on the system that is running the OPC I/O Gate and type:

```
"regsvr32 d:\opt\diw\iogates\opc\bin\opcproxy.dll"
```

Note: This may also help in a system with an I/O gate 3.0.1 on a AW70 Windows NT machine with a OPC server 3.2 on a WinXP machine! At one customer site we had this "Interface not registered" error and running the registry command fixed the problem.

10.3.3 Registering the OPCProxy.dll.

For this problem see: [9.3.2 Get error message: "Interface not registered"](#)

10.3.4 Can't get ECB's on-line.

You can not get the ECB's On-Line at all. One of the possible causes can be that the **OPC I/O Gate** has left the building. If the **OPC I/O Gate** is not running for any reason you can't get the ECB's on-line. Run the OPC I/O Gate from the Start menu and try again.

10.3.5 ECB's in System Management repeatedly switch red/white.

The symptom is that the ECB's go on-line briefly and then go off-line (failed state) after which the whole thing start over again. This is particularly seen on 2 Foxboro stations on different I/A Series systems trying to connect to each other via the second Ethernet.

The reproducible cause we have found:

When both the OPC Client and the OPC Server are Foxboro systems (separate systems that is) and both have identical Nodebus or Ethernet IP addresses (yes we are talking about the I/A Series network side). You will have this problem. Note that the Secondary network is 100% OK in this setup, IP addresses etc. are correct. OPC connects through this secondary interface only but still the primary network configuration causes this problem to exist.

The workaround for this is to change the IP address (via System Definition) of one of the two platforms.

10.3.6 None of the configured items work.

Most probably the syntax is incorrect. If the items show up in the OPC I/O Gate window, select the item and find out what the messages is.

Try to install an **OPC Explorer** from **Matrikon** and browse the Server. The item's syntax should show. Enter that string exactly as shown into the Integrated Control Configurator for the F-block VALID field and you should be OK.

10.3.7 OPC Items do not work and group scan rate shows "0 msec".

After starting up the I/O Gate ,the OPC items do not show up or appear very slowly. A very important symptom is the scan rate in the OPC I/O Gate window for that group. This shows a value of "0" instead of a more realistic value like "500" or "2000". This should be investigated as the I/O Gate may loose communication with the Object Manager (System Management goes red on you).

10.3.8 OPC I/O Gate crashes after installing QF1001908.

The quickfix for the OPC I/O gate: QF1001908 MUST NOT BE installed on Windows XP based I/A Series stations (I/A 6.4? and higher) because this version contains newer **dll** files than the quickfix provides.

10.3.9 Blocks in the OPC I/O Gate appear, but slowly.

Symptom: It takes very long for all blocks to appear in the OPC I/O gate tree (they come like 10 every 30 seconds).

Most likely solution: remove incorrect VAL_ID's or incorrect references in the *Aliasfile* on the OPC server. NOTE: if a C:B.P as mentioned in the *aliasfile* is not found in the CSA database, this will not result in an error status in the OPC I/O gate!!

In the PCTCFGAIM.1.OPC.log file on the OPC server, these items can be found by searching on the text string "CSA error".

10.3.10 Only read of OPC items possible, No writes!

Symptom: All works fine, adding groups and items is not a problem and the system is responsive. However, it is only possible to read data from the OPC Server. Generally speaking this an issue that must be solved through **ApiAdmin**. The user account created there will by default allow reads of all objects only. Use the checkboxes provided in ApiAdmin under the Security tab to make the modifications desired.

10.3.11 When writing OPC items, all ECB's go OFF-Line.

I don't have a clue yet.

10.4 AIM*OPC Server.

The AIM*OPC related FAQ section.

10.4.1 Can't enter a license key because OPCDAO or OPCHDA are not in list.

If you have installed the AIM*OPC Server you need to enter the license key to get it working. On the installation CD you will find a file "**aimreg.dat**" This file MUST be copied to the d:\opt\aim\bin directory on the AW.

10.4.2 No packages at all in the list when trying to enter the license.

You copied the "**aimreg.dat**" file to the proper location but forgot to remove the "read only" attribute from the file. Locate the file with explorer, right click and toggle the **Read-only** attribute (see picture)

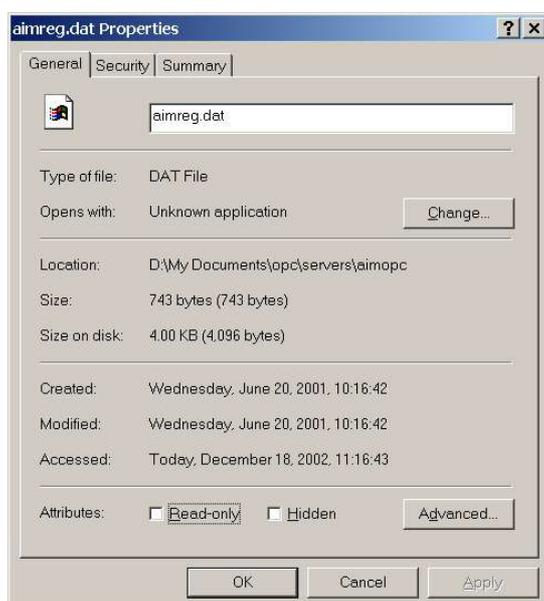


Figure 94 Remove read-only from aimreg.dat file

10.4.3 OPC Server does not start (does not show in systray).

Starting the AIM OPC server (through the Windows START menu) to access the configurator, does not appear to have any effect. i.e.:the red cross does not appear in the **Systray**.

Symptom: After maybe up to five minutes the OPC Server/configurator may show in the Systray after all.

A possible cause for this may be that the API server is not available (eg aimapi is not running on the AW51).

10.4.4 Error “PSTCFG: Interface Not Supported”.

Problem: when starting the OPC client the error PSTCFG "Interface not supported"

Solution: This problem is caused by making changes to the DCOM settings on the OPC Server side. The cause for this error can be found in the Windows Registry on the OPC Server machine.

To correct the problem: Start **dcomcnfg** and open the properties screen for the AIM AT OPC application, Using the clipboard copy the Program ID (example {335E7241-6B49-101B-9CA8-9240CE2738AE}). Next start **regedit** and using the search function, locate that particular key. The value of this key should be “1”. In this registry, the **authenticationlevel** must be set to "1". (by doubleclicking on the **authenticationlevel** and entering the value "1").

10.4.5 Error: Access Violation after re-installing OPC Server 3.3.

Problem: after re-installing the AIM OPC server 3.3 and opening the AIM OPC configurator the following error pops containing texts like "Access violation" and "PSTCFG" and a bit address.

Cause: the DCOM settings have not been configured for the AIM OPC server After the new installtion all settings are back to the default!!

Solution: Re-configure the DCOM settings for the OPC server.

10.5 APIAdmin tool.

Some problems encountered when using ApiAdmin for NT/XP or UNIX

10.5.1 No Servers found when invoking APIAdmin.

On Windows platform this can related to the fact that broadcast is disabled and no server aliases are configured.

On UNIX when starting **apidamin** from the prompt **You must start the **apidamin** application from C-shell . Doing this from the Bourne shell results in an empty Server list no matter what you have configured!**

10.6 Only read of OPC items possible, No writes!

Generally speaking this an issue that must be solved through **ApiAdmin**. The user account created there will by default allow reads of all objects only. Use the *checkboxes* provided in **ApiAdmin** under the Security tab to make the modifications desired.

10.7 Network connection setting not available after installing OPC.

Note for Windows 2000 Family (and possibly XP hosts):

Setting the Default Impersonation to Anonymous may cause Network and Dial-up connections previously defined to be no longer visible.

Windows Explorer uses the **Default Impersonation** level settings to access these objects. Anonymous access does not give the correct privileges to display these. See also *Microsoft knowledge Base Article 273461*.

Another workaround: OPC clients running on Windows 2000 could can run with dcom settings: "connect" and "identify".

10.7.1 Not all items become active.

Problem: When **OPC I/O Gate** starts, not all items become active (It does NOT appear in the I/O Gate display).

Possible solution: Increase the *scanrate* of the F-blocks, note that fastest *scanrate* in a group is forced on all FBLOCKS in that group.

Add (per OPC group) two dummy F-blocks as the first two blocks in that Compound and/or turn on the Compounds one by one with 10 seconds poses.

Note that each dummy block should have its own parameter on the OPC Server. Do not point all blocks to the same OPC item!

10.7.2 Analog values shown in the OPC I/O Gate show “.0000”.

Symptom: in the OPC I/O gate window an analog output value gives the value ".0000".

Solution: This does not always indicate a BAD tag. Try to modify the value of this output. This will probably show the correct value after an update.

This can be related to an incorrect AIMAPI version.

10.7.3 Cannot start the OPC Server Configurator.

Symptom: When trying to start the **OPC Server** via the executable `\aim\opc\aimopcsvr.exe`, nothing happens and the error "**Could not register class factories**" appears.

Possible cause for this: If the Windows taskbar is hidden (in case of operator workstation functionality on AW70 this is likely to be the case) then the OPC server configurator can not be started.

Solution: The OPC Server configurator can only be accessed/started by enabling the Windows taskbar: **Windows explorer>My computer>control panel> foxboro icon>** and then: (start>settings>control panel>foxboro icon, enable taskbar(Full control or something like that) To activate this new settings, do the following (this has not been intensively tested!):

- Kill the `explorer.exe` process in the **taskmanager**.
- Then via the **taskmanager>file>new task(Run)>explorer.exe**

The **taskbar** should now appear, and the **AIMOPCSVR** can now be started via the the Start menu.

10.7.4 Initial connection fails, retry always works.

Problem: During startup of AW70 with on platform OPC server, the OPC server is started before AIM API is started.

Probable solution: Delay the start of the OPC server process until AI and API are fully started.

10.8 FDSI FBM & OPC Driver (Work In Progress)

10.8.1 Hey... It doesn't work.

No OPC connection is ever easy, especially not when you try to set up a particular connection for the first time. The black-box nature of the FDSI FBM does not really improve this. Most of the time, OPC connections fail because of the security (DCOM) settings on the Server. If you use Windows XP SP2, then things can get really difficult.

The following things can be done to rule out the FBM:

10.8.1.1 Check the port configuration.

The first thing to check is the IP configuration, and the network. Can the FBM and Server reach each other? (No firewalls in between?) Try to ping the FBM from the server and vice-versa:

(In this example the IP addresses from the FBM and Server are 192.168.11.25 and 192.168.11.13 respectively)

```

c:\ E:\WINDOWS\System32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

E:\Documents and Settings\fox>ping 192.168.11.25

Pinging 192.168.11.25 with 32 bytes of data:

Reply from 192.168.11.25: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.11.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

E:\Documents and Settings\fox>telnet 192.168.11.25_
  
```

Figure 95 Ping the FBM from the server

Then, telnet to the FBM (see figure above) and try to ping the server:

```

c:\ E:\WINDOWS\System32\cmd.exe
Welcome to the Windows CE Telnet Service on FBM23x

Packet CMD v 4.10
> ping 192.168.11.13
Pinging Host 192.168.11.13
Reply from 192.168.11.13: Echo size=32 time=2ms TTL=128
Reply from 192.168.11.13: Echo size=32 time=2ms TTL=128
Reply from 192.168.11.13: Echo size=32 time=2ms TTL=128
Reply from 192.168.11.13: Echo size=32 time=1ms TTL=128
> exit

Connection to host lost.

E:\Documents and Settings\fox>_
  
```

Figure 96 After telnetting to the FBM, try to ping the server back

If this works, you know that the port-configuration file was successfully downloaded, the FBM assigned the IP address to its network card correctly and the network is also correctly connected. Recheck also the other information in port-config file, such as the Server's IP-address and the CLSID. See also next tip.

10.8.1.2 Are the correct files downloaded?

If you are making changes to the Port or Device config files but the changes don't seem to help, you might want to check to see the correct file is downloaded and active in the FBM. Start a cmd.exe and telnet to the FBM. CD to the directory /fox/lib. In this directory, the downloaded configuration files (.XML) will appear, with their CRC's as file-names. There can be more than one file present, as old files will not always be removed. In the example below there is 1 port-configuration file and 2 device configuration files. You can see the contents of the file with

the “type” command.

```

E:\WINDOWS\System32\cmd.exe
Welcome to the Windows CE Telnet Service on FBM23x
Pocket CMD v 4.10
> cd fox/lib
\fox/lib> dir

Directory of \fox/lib

03/30/05 09:28a          97060 DriverDownload.zip
11/01/04 12:01p       109568 DriverDll.dll
11/01/04 12:01p       59904 FoxOPCI0.dll
11/01/04 12:01p       47616 OPCWinCEPS.dll
11/01/04 12:01p       10752 FbmInit.exe
11/01/04 12:01p          179 version.xml
11/01/04 12:01p          <DIR>          Config
04/01/05 12:30a          660 90EDC4EC.xml
03/30/05 09:37a          695 B43C2797.xml
04/01/05 12:31a       242368 C24BB454.xml

Found 10 file(s). Total size 568802 bytes.

\fox/lib> type 90*.xml
\fox/lib\90EDC4EC.xml:

<FBM233 Assembly="OPCPortLib.dll" Type="OPCPortLib.OPCPortCtrl">
  <Port>
    <Main>
      <IPAddress>192.168.11.25</IPAddress>
      <SubnetMask>255.255.255.0</SubnetMask>
      <DefaultGateway>192.168.11.254</DefaultGateway>
      <OPCIAddress>192.168.11.13</OPCIAddress>
    </Main>
    <Backup>
      <IPAddress>192.168.21.25</IPAddress>
      <SubnetMask>255.255.255.0</SubnetMask>
      <DefaultGateway>192.168.21.254</DefaultGateway>
      <OPCIAddress>192.168.21.13</OPCIAddress>
    </Backup>
    <OPCServerName>KOPL.OPCServer</OPCServerName>
    <OPGClassID><78239CED-64B9-4969-8441-4C36F7367F26></OPGClassID>
  </Port>
</FBM233>
\fox/lib> type B4*.xml
\fox/lib\B43C2797.xml:

<Device Assembly="OPCDeviceLib.dll" Type="OPCDeviceLib.OPCDeviceCtrl" Name="Simulation Server">
  <Groups>
    <Group Name="HeartbeatGroup">
      <ScanRate>2000</ScanRate>
      <Points>
        <Point Name="Saw-toothed Waves.Int2" />
      </Points>
    </Group>
    <Group Name="group1">
      <ScanRate>1000</ScanRate>
      <Points>
        <Point Name="QUEUE1">
          <OPCIItem>Saw-toothed Waves.Int4</OPCIItem>
        </Point>
        <Point Name="RANDOM1">
          <OPCIItem>Random.Real14</OPCIItem>
        </Point>
        <Point Name="QUEUE2">
          <OPCIItem>Triangle Waves.UInt2</OPCIItem>
        </Point>
      </Points>
    </Group>
  </Groups>
</Device>
\fox/lib> exit

Connection to host lost.
E:\Documents and Settings\fox>
  
```

Figure 97Checking the files inside the FBM

If all configurations in the Port/Device files are correct and you can ping back and forth to the FBM, the problem is most likely NOT in the FBM.
If the ECB's are yellow/white, The OPC Diagnostic client and the DCI blocks can give some more information on OPC items themselves.

10.8.2 FBM Not responding to commands, Rebooting the FBM

The FBM OS is Windows... Sometimes it just helps rebooting the FBM, sometimes it's just plain necessary because it does not respond to commands. Generally speaking, it is not necessary to

reboot after IP or other configuration changes.

There are three ways of rebooting the FBM:

1. Click “Reset” on the Equipment Change page in system management.
2. If the FBM does not respond to system management anymore, you can try to Telnet to the FBM and type “fbmreboot”
3. If you cannot telnet the fbm anymore or there is no telnet session around, you have to pull and reseal the FBM on the baseplate.

10.8.3 Diagnostic Blocks and parameters

This chapter holds some tips and tricks that are not documented in the official documentation, and can therefore be subject to change by development.

10.8.3.1 ECB Parameters

Both ECB blocks (ECB200/2 and ECB201) have a lot of information available on their parameters, and it's really worth looking at them. There are some special status parameters that are not fully documented. These parameters hold a lot of information that is visible in system management. The parameters are 32-bit integers, and each bit has it's meaning. Here are some interesting ones, but there are a lot more:

Port Configuration File Download Active	: ECB200/2.FBM'sTS, Bit 25
Device Configuration File Download Active	: ECB201.DEVSTS, Bit 28
Port ECB DCI Ready:	: ECB200/2.FBM'sTS, Bit 32
Device ECB DCI Ready:	: ECB200/2.DEVSTS, Bit 32
Educating Tracker (FBM233)	: ECB202.MDIAG, Bit 14 (and ECB202.BDIAG)

Also have a look at ECB200's FDIAG, ECB202's MMODE and BMODE. Dissecting the ECB200/201/202 Detailed Displays will show a lot more interesting parameters.

10.8.3.2 Diagnostic DCI Blocks

You can build DCI blocks in your CP which can interface directly to the ECB200/202 to get interesting information on the FBM that is shown on the output parameters of the DCI blocks. To use this, you have to build a DCI block, fill in the ECB200/202's device-id as IOM_ID (NOT ECB201's IOM_ID as the normal I/O points do) and use the following PNT_NO values:

WARNING: These functionalities are created for debugging purposes by development and are not documented (yet). The drivers are still being developed at the time of this writing and this may or may not work with the OPC driver you have.

DCI BlockType	PNT_NO value	Function
STRIN	\$FBM0_IPADDRESS	IP address of FBM in Single mode or Master FBM IP Address in redundant mode.
STRIN	\$FBMM_IPADDRESS	IP address of FBM of Main FBM in redundant mode.
STRIN	\$FBMB_IPADDRESS	IP address of FBM of Backup FBM in redundant mode.
RIN	\$FBM0_CPU_USAGE	Single FBM CPU load or Redundant Pair Master's CPU load. (0-100 %)
RIN	\$FBMM_CPU_USAGE	Redundant pairs' Main FBM CPU load. (0-100 %)
RIN	\$FBMB_CPU_USAGE	Redundant pairs' Backup FBM CPU load. (0-100 %)
RIN	\$FBM0_MEM_USAGE	Single FBM Memory load or Redundant Pair Master's Memory load. (0-100 %)
RIN	\$FBMM_MEM_USAGE	Redundant pair's Main FBM Memory Load (0-100%)
RIN	\$FBMB_MEM_USAGE	Redundant pair's Backup FBM Memory Load (0-100%)
BIN	FBMM_OPC_ERROR	Main FBM in redundant pair has not connected all configured points correctly (yet) *
BIN	FBMB_OPC_ERROR	Backup FBM in redundant pair has not connected all configured points correctly (yet) *

**This feature is under development at the time of writing. An FBM232 version of this is not available yet.*

See example block configuration:

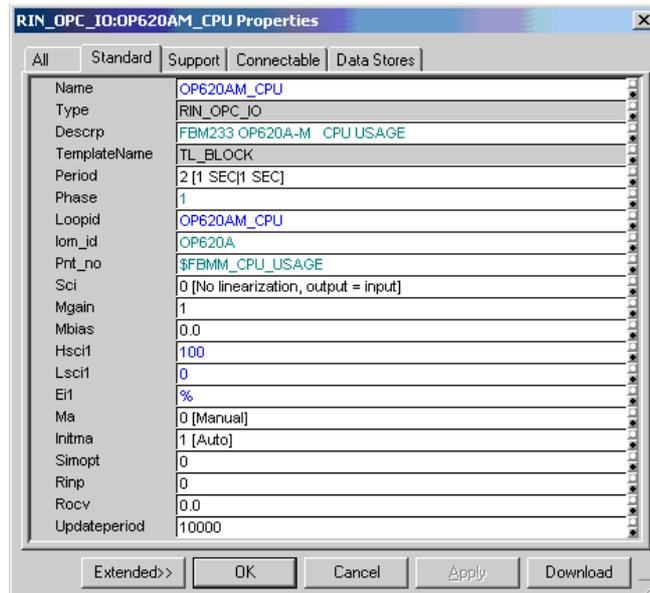


Figure 98CPU Usage Diagnostic block example

