

# ASTRA

OPC Client Device Driver

[www.renuelectronics.com](http://www.renuelectronics.com)

## ⌘ Tables of Contents ⌘

Preface	3
Introduction	3
Process Control Information Architecture	4
General Information of OPC	5
OPC Fundamentals	6
OPC Historical Data Access Overview	8
General OPC Architecture and Components	9
Design Modification as per OPC	10
Special Notes and Examples	12
About Registry	15
About the delays of Read and Write	17
Optimizations	18
Limitations	19
Errors	20

## ⌘ Preface ⌘

This document introduces user to the OPC Client for Astra. It contains technical information about OPC Client. This document gives you a broad idea of how to use OPC Client with Astra. This document broadly tell you about the capabilities and technical details of OPC Client.

## ⌘ Introduction ⌘

Today manufacturers face the task of integrating plant floor data into their business systems. This task is made difficult because of the almost unlimited number of methods plant floor devices utilize to communicate to other systems.

The current situation in the process control market is analogous to the early stages of the computer industry where two machines from different manufactures could not communicate (IBM and DEC for example) without many hours of custom programming. Today, a wide variety of computers communicate over vast distances, and diverse networking systems. Applications written utilizing powerful Database and Client/Server tools that support these standards allow the application programmer to focus on the task at hand and not the communication underpinnings.

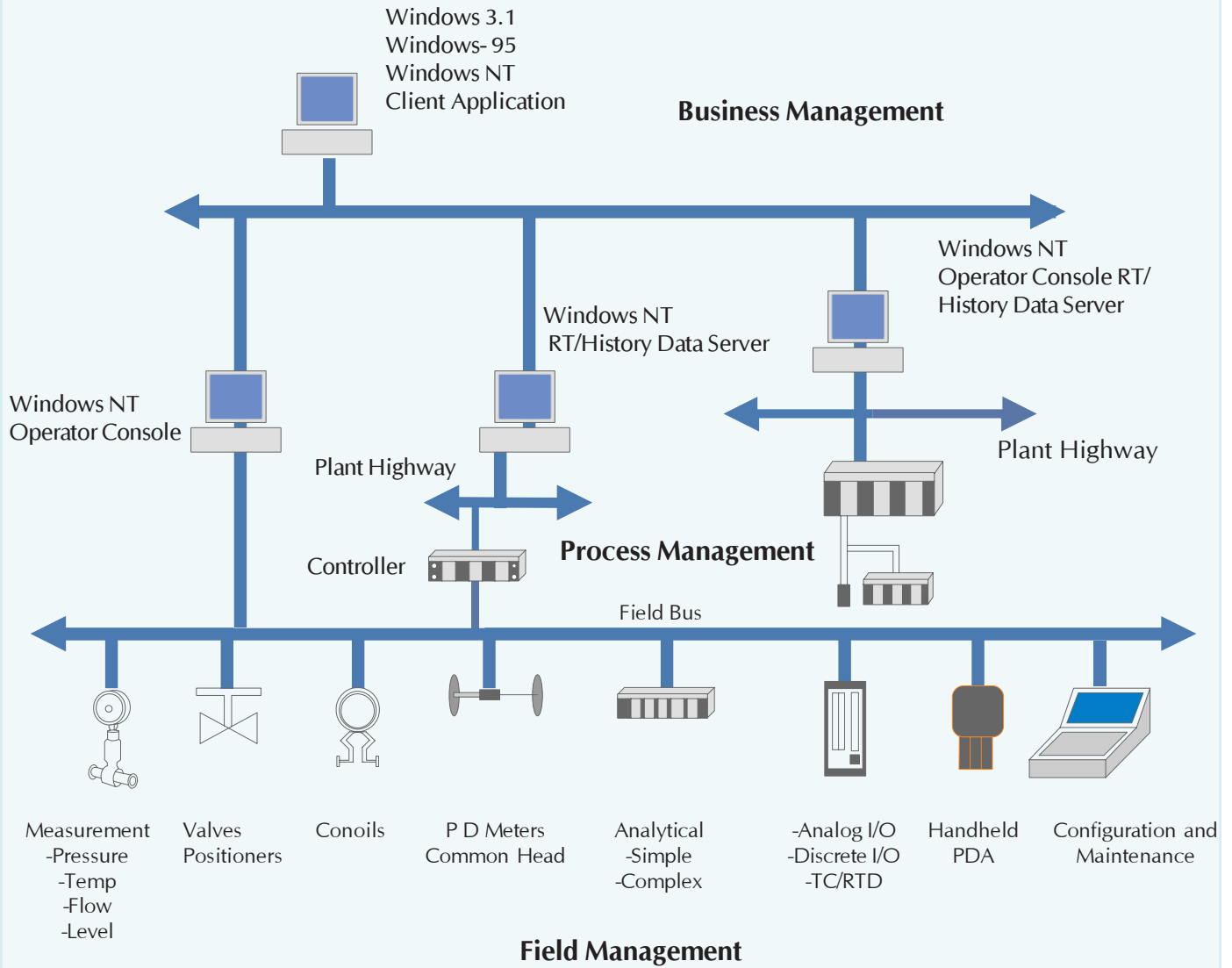
The information architecture for the Process Industry, shown in Figure on next page, involves the following levels:

**Field Management:** With the advent of “smart” field devices, a wealth of information can be provided concerning field devices that has not been available in the past. This information provides data on the health of a device, its configuration parameters, materials of construction, etc. All this information must be presented to the user, and any applications using it, in a consistent manner.

**Process Management:** The installation of SCADA systems to monitor and control manufacturing processes has made data available electronically which had in the past been gathered manually. This data must be provided in a consistent manner to the operators and engineers responsible for making decisions based on that data.

**Business Management:** Benefits can be gained beyond those typically cited when justifying the installation of control systems. This is accomplished by integrating the information collected from the process into the business systems managing the financial aspects of the manufacturing process.

# ⌘ Process Control Information Architecture ⌘



## ⌘ General Information of OPC ⌘

**OLE for Process Control (OPC™) is designed to allow client applications access to plant floor data in a consistent manner. With wide industry acceptance OPC will provide many benefits:**

- ⊙ Hardware manufacturers only have to make one set of software components for customers to utilize in their applications.
- ⊙ Software developers will not have to rewrite drivers because of feature changes or additions in a new hardware release.
- ⊙ Customers will have more choices with which to develop World Class integrated manufacturing systems.

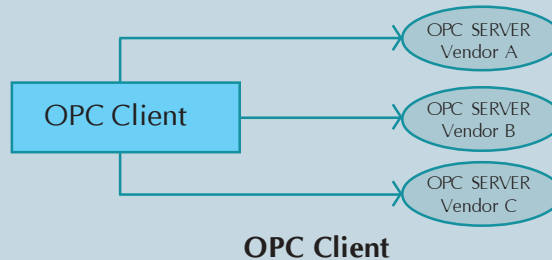
With OPC, system integration in a heterogeneous computing environment will become simple. Leveraging OLE/COM the environment shown in on next page becomes possible.



## OPC Objects and Interfaces

OPC is based on Microsoft's OLE/COM technology. This specification describes the OPC COM Objects and their interfaces implemented by OPC Servers. An OPC Client can connect to OPC Servers provided by one or more vendors.

OPC Servers may be provided by different vendors. Vendor supplied code determines the devices and data to which each server has access, the data names, and the details about how the server physically accesses that data.



Vendor supplied code determines the devices and data to which each server has access, the data names, and the details about how the server physically accesses that data.

## OPC DataAccess Overview

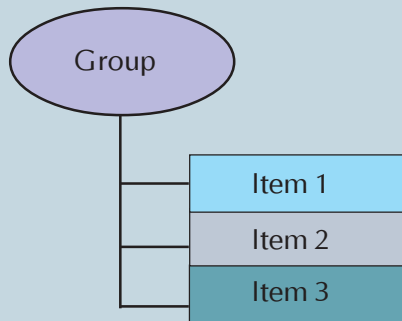
At a high level, an OPC DataAccess Server is comprised of several objects: the server, the group, and the item. The OPC server object maintains information about the server and serves as a container for OPC group objects. The OPC group object maintains information about itself and provides the mechanism for containing and logically organizing OPC items.

The OPC Groups provide a way for clients to organize data. For example, the group might represent items in a particular operator display or report. Data can be read and written. Exception based connections can also be created between the client and the items in the group and can be enabled and disabled as needed. An OPC client can configure the rate that an OPC server should provide the data changes to the OPC client.

There are two types of groups, public and local (or 'private'). Public is for sharing across multiple clients, local is local to a client. Refer to the section on public groups for the intent, purpose, and functionality and for further details. There are also specific optional interfaces for the public groups. Within each Group the client can define one or more OPC Items.



# OPC Fundamentals



**Group / Item Relationship Diagram**

The OPC Items represent connections to data sources within the server. An OPC Item, from the custom interface perspective, is not accessible as an object by an OPC Client. Therefore, there is no external interface defined for an OPC Item. All access to OPC Items is via an OPC Group object that contains the OPC item, or simply where the OPC Item is defined.

Associated with each item is a Value, Quality and Time Stamp. The value is in the form of a VARIANT, and the Quality is similar to that specified by Fieldbus.



**Note:** The items are not the data sources - they are just connections to them. For example, the tags in a DCS system exist regardless of whether an OPC client is currently accessing them. The OPC Item should be thought of as simply specifying the address of the data, not as the actual physical source of the data that the address references.

For example, the tag FC101 may have the following conditions associated with it: HighAlarm, HighHighAlarm, Normal, LowAlarm, and LowLowAlarm.

On the other hand, an event is a detectable occurrence that is of significance to the OPC Server, the device it represents, and its OPC Clients. An event may or may not be associated with a condition. For example, the transitions into HighAlarm and Normal conditions are events that are associated with conditions. However, operator actions, system configuration changes, and system errors are examples of events that are not related to specific conditions. OPC Clients may subscribe to be notified of the occurrence of specified events.

## ⌘ OPC Historical Data Access Overview ⌘

The IOPCEventServer interface provides methods enabling the OPC Client to:

- ⊙ Determine the types of events that the OPC Server supports.
- ⊙ Enter subscriptions to the specified events, so that those OPC Clients can receive notifications of their occurrences. Filters may be used to define a subset of desired events.
- ⊙ Access and manipulate conditions implemented by the OPC Server.

In addition to the IOPCEventServer interface, an OPC Event Server may support optional interfaces for browsing conditions implemented by the server and for managing public condition groups (defined in the following section).

### **OPC Historical Data Access Overview**

Historical engines today produce an added source of information that must be distributed to users and software clients that are interested in this information. Currently most historical systems use their own proprietary interfaces for dissemination of data. There is no capability to augment or use existing historical solutions with other capabilities in a plug-n-play environment. This requires the developer to recreate the same infrastructure for their products, as all other vendors had to develop independently with no interoperability with any other systems.

In keeping with the desire to integrate data at all levels of a business, historical information can be considered to be another type of data.

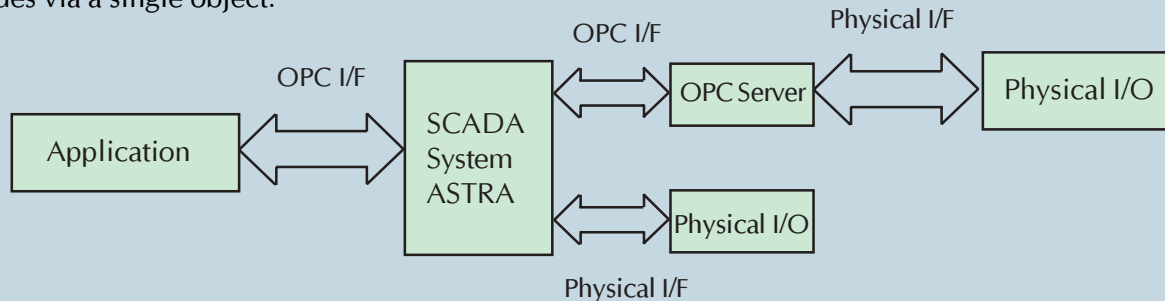
There are several types of Historian servers. Some key types supported by this specification are:

- ⊙ Simple Trend data servers. These servers provided little else then simple raw data storage. (Data would typically be the types of data available from an OPC Data Access server, usually provided in the form of a tuple [Time Value & Quality])
- ⊙ Complex data compression and analysis servers. These servers provide data compression as well as raw data storage. They are capable of providing summary data or data analysis functions, such as average values, minimums and maximums etc. They can support data . updates and history of the updates. They can support storage of annotations along with the actual historical data storage.

## ⌘ General OPC Architecture and Components ⌘

### Where OPC Fits

Although OPC is primarily designed for accessing data from a networked server, OPC interfaces can be used in many places within an application. At the lowest level they can get raw data from the physical devices into a SCADA or DCS, or from the SCADA or DCS system into the application. The architecture and design makes it possible to construct an OPC Server which allows a client application to access data from many OPC Servers provided by many different OPC vendors running on different nodes via a single object.



### General OPC Architecture and Components

OPC specifications always contain two sets of interfaces: Custom Interfaces and Automation interfaces.

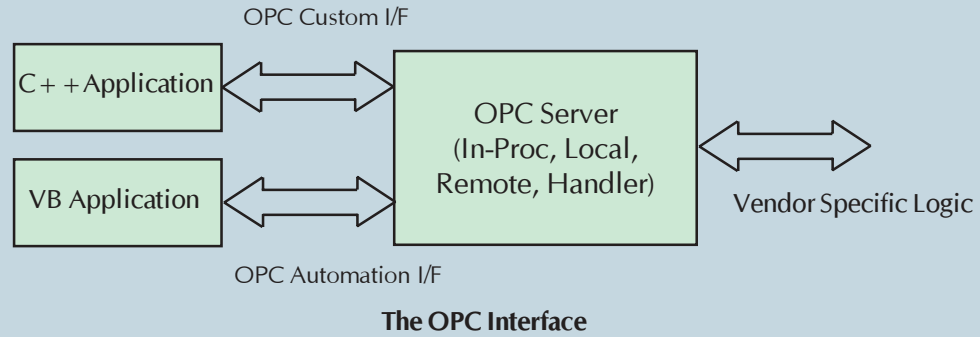
The OPC Specification specifies COM interfaces (what the interfaces are), not the implementation (not the how of the implementation) of those interfaces. It specifies the behavior that the interfaces are expected to provide to the client applications that use them.

Included are descriptions of architectures and interfaces that seemed most appropriate for those architectures. Like all COM implementations, the architecture of OPC is a client-server model where the OPC Server component provides an interface to the OPC objects and manages them.

There are several unique considerations in implementing an OPC Server. The main issue is the frequency of data transfer over non-sharable communications paths to physical devices or other data bases. Thus, we expect that OPC Servers will either be a local or remote EXE which includes code that is responsible for efficient data collection from a physical device or a data base.

An OPC client application communicates to an OPC server through the specified custom and automation interfaces. OPC servers must implement the custom interface, and optionally may implement the automation interface. In some cases the OPC Foundation provides a standard automation interface wrapper. This “wrapperDLL” can be used for any vendor-specific custom-server.

## ⌘ Design Modification as per OPC ⌘



### Modules -

- ⊙ Application Interface
- ⊙ Request Manager
- ⊙ Derived Files for OPC Driver
- ⊙ Files provided by OPC foundation

Each module is responsible for implementing a group of functions.

## ⌘ Design Modification as per OPC ⌘

Module	Major Objects Implements...	Source Directories/Files.	Remarks
Application Interface	CALLBACK DriverProc,  DriverDLL::Initialise	Api.cpp,  Main.cpp	Handle all user defined messages from Astra.  Initializes and creates all managers
Request Manager	RequestManager, RequestContainer,  RegisterContainer, TagContainer, NodeContainer, Error	Requestman.cpp, Request.cpp,  Register.cpp, Tag.cpp, Node.cpp, Error.cpp	Compose, Decompose algorithms and updating Astra tag database and interactions with OPC Server.
Derived Files for OPC Driver	OpcRequestManager, OpcNode, OpcTag, OpcRequest  COPCAdviseSink	Opcrm.cpp, Opcnode.cpp, Opctag.cpp, OpcRequest.cpp  AdviseSink.cpp	Manage connection and data transfer with the OPC Server  Use for callback by the server for asynchronous communication.
Files provided by OPC foundation	IOPCServer, IOPCSyncIO, IOPCAsyncIO, IOPCItemMgt, IOPCGroupStateMgt,  IDataObject	Opc.h      Opc_i.c	Contains all the interface definitions provided by OPC Foundation     Contains IID values of all the interface

## ⌘ Special Notes and Examples ⌘

This chapter gives you idea of how to use ASTRA OPC driver for various OPC servers.

To use the ASTRA OPC Client OPC server should be installed. Follow the procedure given below.

1 > Before using the ASTRA OPC client read the instruction manual of OPC server carefully.

2 > Install the OPC Server on the PC where client is going to run. The OPC server installation will update the registry of the system. From the registry of the PC find the CLASSID of the program. This entry will be in **MyComputer\HKEY\_CLASSES\_ROOT\ServerName\CLSID**.

e.g. **MyComputer\HKEY\_CLASSES\_ROOT\EEI.DirectNET\CLSID** will be **46F74E88-7BFF-11D2-9DE2-0020AF9BB61D**.

or

**MyComputer\HKEY\_CLASSES\_ROOT\KEPServer\CLSID** will be

**70182231-B2D0-11D1-B12A-00A02482A9CB**

3 > Connect the PLC to the PC and create a sample project using the OPC Server. Save the project.

4 > Some server uses the last project saved when initialised. Whereas some server has setting to get particular project invoked. Check that your sever belongs to which category. And make proper settings. ( KEPWare's KEPServer has setting of using particular project. DirectNET server uses last project saved.)

5 > Create a project in ASTRA using OPC Driver. Use proper tag address.

e.g. **a > Using DirectNET OPC Server**

If in the Server project you have a tag named voltage1 in group XYZ and XYZ group is in device INSTRUMENT1, then ASTRA tag address should be **INSTRUMENT1.XYZ.voltage1**.

If in the Server project you have a tag named tag1 in group "group1" and "group" is in device "device1", then ASTRA tag address should be **device1.group1.tag1**.

**b > Using KEPWare's KEPServer**

This Server is similar to DirectNET server except the group field. i.e. suppose if a channel is having device INSTRUMENT1 and tag voltage1 then in ASTRA the tag address should be **INSTRUMENT1.voltage1**.

In case of other OPC server please refer to help of respective server. And set the address of tag in ASTRA accordingly.

## ⌘ Special Notes and Examples ⌘

6 > Make the entry in OPC.INI file as LOCAL = 1. ( Before testing with remote server it is advisable to check with Local server)

7 > Save, compile and run the project.

8 > Check if all the tags defined are working properly by writing to tags. In case of Local servers check the tag values with monitor option of the Server, if provided. You may use simulation mode if actual device is not available. However these options are server dependent.

9 > If any tag is not working check the event logger message and refer to chapter 8 for errors, their causes and remedies.

10 > If all tags are working it indicates that the communication between Client and server is working.

11 > Create the final project in the server and in the client (ASTRA) and check project finally.

The ASTRA OPC Client works across the PC's provided following conditions are satisfied.

1 > The client PC should be running Windows 95/98/NT with DCOM installed.

2 > **The OPC Server PC must be running Windows NT with service pack 3 installation.**

3 > The OPC Server should be installed on the server (OPC server).

4 > Using the DCOMCONFIG.EXE (DCOM configuration utility of the Windows NT) proper access should be given to the Server. e.g. if user XYZ is going to use the OPC Server he ( his login ) should be given full access rights to the OPC Server.

5 > Set the OPC Server properties, as program will run on same computer.

6 > Similarly DCOM should be installed in Client PC.

7 > **The user must login on Client PC with the same user name and same password. (If user name is astra1 and password is astra1 on the OPC server PC , then same user name and password should be used to login Client PC). Otherwise both OPC Client PC and OPC Sever PC should be logging to common Windows NT Server to validate user information via same domain name.**

8 > In the OPC.INI file, the setting of "LOCAL" key should be 0 if user wants to run OPC Server on a remote PC.

## ⌘ Special Notes and Examples ⌘

9 > If the Server program is not installed on Client machine then the CLASSID of the Server Program must be supplied to Client program through OPC.INI file. In the OPC.INI file if the entry "REGISTRY = 1" then ASTRA will look for this information in Windows Registry. If you have installed the Server program on same OPC client machine then this entry is valid. If not, then make "REGISTRY = 0" and write the CLASSID as follows

*CLASSID = 46F74E88-7BFF-11D2-9DE2-0020AF9BB61D ( for KEPWare )*

This you should have obtained from registry entry of remote PC where OPC Server is installed.

10 > The OPC sever PC's IP address should be written next to key IPADDRESS.

*[OPC1]*

*NODE = OpcNode1*

*SERVER = EEI.DirectNET*

*TIMEBIAS = 0*

*DEADBAND = 0.0*

*LOCAL = 0*

*IPADDRESS = 192.10.10.1*

*REGISTRY = 1*

*CLASSID = 46F74E88-7BFF-11D2-9DE2-0020AF9BB61D*

Here the address 192.10.10.1 indicates the IP address of PC running OPC Server. To know the IP address of the OPC server PC check Network properties of PC running OPC Server.

If any of the above condition is not met then the ASTRA will report error in event logger. Refer to next chapter for errors and their explanation.



**Note:** DO NOT CLOSE OPC SERVER PROGRAM FORCEFULLY. THE ASTRA WILL CLOSE THE PROGRAM AUTOMATICALLY. IF THE PROGRAM IS CLOSED FORCEFULLY THEN SERVER/ CLIENT MAY NOT RELEASE ALL RESOURCES, IN SUCH CASE NEXT STARTUP WILL ALWAYS FAIL. IN CASE , THIS HAPPENS ACCIDENTALLY THEN REBOOT THE CLIENT AND SERVER.



# About Registry



## About Registry

Windows Registry plays a very important roll in this client server application. One important point is to be noted, that if you have configured a node to work on local PC i.e. LOCAL = 1 in OPC.INI and the windows registry entry of the PC points to some other PC (remote server) then the Server will be invoked on the PC where registry points.

Normally installation of server will create following entries. If the Client PC is not having server program installed then you should note down the CLASSID of the server from the registry of OPC Server PC. And provide this in OPC.INI file. To avoid noting down of CLASSID information there is another way.

- 1 > On the Server PC open the registry by running program **REGEDIT**.
- 2 > Go to the position **MyComputer\HKEY\_CLASSES\_ROOT | ServerName | CLSID**. Where ServerName is name of OPC Server that you have installed on PC.
- 3 > Go to Registry Menu option **Export Registry File**.
- 4 > Give the name of target file and export this portion of registry.
- 5 > After exporting this registry file, go to client PC and again Run the **REGDIT** program.
- 6 > Here you run the option **Import Registry File**. Locate the file, which you have exported previously.
- 7 > Check if the entry is added or not.
- 8 > Quit the **REGEDIT** program.

*Since you are changing the entries of Windows Registry manually be careful. If you do this then there is no need to give the CLASSID information in OPC.INI file. In such case there should be corresponding entry REGISTRY = 1 in OPC.INI file.*

## Check Points.

In case of any errors check following points.

- 1 > ASTRA and OPC Driver are installed properly.
- 2 > DCOM for windows is installed properly.
- 3 > OPC Server is installed properly.
- 4 > OPC.INI file is present is ASTRA project directory.



## About Registry



- 5 > The Class ID of the Server is provided to ASTRA (either through INI file or through registry).
- 6 > Correct IP Address of Server PC and correct OPC server name is entered in OPC INI file.
- 7 > ASTRA project is correctly configured.
- 8 > Project is created on OPC server with correct settings. The same project name is provided as startup project.
- 9 > Proper User rights are given to current user and same password is used on OPC Server and Client.
- 10 > Node Name in the OPC.INI file should match the names in ASTRA Project. E.g. if "INSTRUMENT" is name of node in ASTRA then in OPC.INI file should contain same node name INSTRUMENT along with respective Server settings.

### **Using Multiple ASTRA OPC clients**

One special point should be noted while using multiple ASTRA OPC clients. If you want to use Multiple OPC Clients and one of the clients is located on OPC Server machine then **first client on the OPC server PC must be started first**. Other clients must be started after that.

e.g. If PC A is going to run OPC Server and PC B , PC C , PC D are other machines which are going to run client then first client program should be started on PC A ( which is going to run OPC Server). After this only other clients should be started. If this sequence is not followed then user may not get connection to the OPC server.

## ⌘ About the delays of Read and Write ⌘

### **About the delays of Read and Write**

Some OPC Servers return the read request immediately by reading the Tag value from the cache, this can cause problem in some cases. In such case you should consult your OPC Server vendor. e.g. suppose there is one tag in ASTRA with scan time 0. This means that the tag will not be scanned normally. Now suppose user writes value Y to this tag (previous tag value is X). Now ASTRA will write this value to OPC Sever and read the value back from the Server after write returns. However some server takes the write value into buffer and writes it lateron but returns immediately (if synchronous write semantics are not adhered to). So ASTRA on its next read will be given value which was present before write. So ASTRA will display the earlier value returned by OPC Server, which is OLD value. This can be noticed only if tag is having scan time 0, since there will not be any further read.

### **Setup under which the product is tested.**

The ASTRA OPC Client is tested with the project included with the installable disk. The OPC servers used for the testing are KEPWare's KEPServer , DirectNET , and Factory Soft's FactorySoft.Sample. This project is tested on following setup.

OPC Server - Windows NT Workstation.  
Ver 4.00.1381 , Service Pack 3 Installed.  
IP Address 192.10.10.194

OPC Client -  
Client 1: Windows NT Workstation.  
Ver 4.00.1381 , Service Pack 3 Installed.  
IP Address 192.10.10.122

Client 2: Windows 95  
Ver 4.00.950 , DCOM Installed.  
IP Address 192.10.10.53

Client 3: Windows 98  
Ver 4.10.1998 , Service Pack 3 Installed.  
IP Address 192.10.10.146

ASTRA version used for all clients is ASTRA 32.1.4

## ⌘ About the delays of Read and Write ⌘

The Disk provided to you contains the projects of the Server also. However check the OPC server you have. If it is different then you may have to create new project for the same. You may take help from the OPC server documents. It is necessary that the OPC Server you have should run the project which is having the tags defined in OPC client i.e. ASTRA or you may reconfigure ASTRA project as per your requirements.

## ⌘ Optimizations ⌘

There is no optimization done in driver.  
The current working condition is – one Group, One tag, one\_fetch.



## Limitations



### **There are following limitations of the OPC Driver for ASTRA.**

- 1 > The Remote Server must be installed on Windows NT. Client can be Windows 95/98/NT.
- 2 > In case of Remote server, on server PC using DCOM Config utility, the Server should be configured to provide access to users. This access should include complete access.
- 3 > In case of Windows 95/98 Client, DCOM should be installed.
- 4 > Also user should be created on the PC with same password which is having access to COM Server. This is must when there is no Windows NT server to validate the password of the user.
- 5 > In case of Windows NT client, same things should be checked.
- 6 > The ASTRA OPC Client uses default security internally.
- 7 > The correct Class ID should be provided to OPC client either in OPC.INI file or registry.
- 8 > In case of abnormal termination of OPC Server the Client and the Server should be restarted.

### **Using Multiple ASTRA OPC clients**

One special point should be noted while using multiple ASTRA OPC clients. If you want to use Multiple OPC Clients and one of the clients is located on OPC Server machine then **first client on the OPC server PC must be started first**. Other clients must be started after that.

e.g. If PC A is going to run OPC Server and PC B , PC C , PC D are other machines which are going to run client then first client program should be started on PC A ( which is going to run OPC Server). After this only other clients should be started. If this sequence is not followed then user may not get connection to the OPC server.

### **About the delays of Read and Write**

Some OPC Servers return the read request immediately by reading the Tag value from the cache, this can cause problem in some cases. In such case you should consult your OPC Server vendor. e.g. suppose there is one tag in ASTRA with scan time 0. This means that the tag will not be scanned normally. Now suppose user writes value Y to this tag (previous tag value is X). Now ASTRA will write this value to OPC Server and read the value back from the Server after write returns. However some server takes the write value into buffer and writes it later on but returns immediately (if synchronous write semantics are not adhered to). So ASTRA on its next read will be given value which was present before write. So ASTRA will display the earlier value returned by OPC Server, which is OLD value. This can be noticed only if tag is having scan time 0, since there will not be any further read.



## Errors



The entire time a Astra project is running, the Event Logger displays the status and any errors that the program generates. The driver utilizes the Event Logger to display error messages regarding the driver. Below are the error messages, the probable cause and most likely solution to all the errors the driver can generate.

### Errors displayed as strings

1. NULL Pointer for Login Data
2. NULL Pointer for Project Path
3. NULL Pointer for Tag Table
4. NULL Handle for Data Manager

**Explanation :** Internal Fatal Error.

**Action :** Contact Astra support.

5. Insufficient Memory for Request Manager
6. Insufficient Memory for Transaction Manager
7. Insufficient Memory for Device Manager

**Explanation :** Internal Fatal Error.

**Action :** Try making more memory available for the project.

8. Cannot Pagelock Tag Table

**Explanation :** Internal Fatal Error.

**Action :** Contact Astra support.

9. Cannot Open File PLCTAG.DAT
10. Cannot Read File PLCTAG.DAT

**Explanation :** Internal Fatal Error. The input file PLCTAG.DAT does not exist or is corrupt.

**Action :** Open the project in the configuration mode and close it, this process recompiles the PLCTAG.DAT file.

11. Insufficient Memory for Tag
12. Insufficient Memory for Tag2
13. Insufficient Memory for Tag Container
14. Insufficient Memory for Node
15. Insufficient Memory for Node Container

**Explanation :** Internal Fatal Error.

**Action :** Try making more memory available for the project.

16. No Tags in the Project

**Explanation :** Internal Fatal Error. The driver detected no valid tags in the project.

**Action :** Recheck the project in the configuration mode. See if any tags are assigned to this particular device. See if the Node details are correct.



## Errors



### 17. No Valid Nodes in the Project

**Explanation :** Internal Fatal Error. The driver detected no valid nodes in the project.

**Action :** Recheck the project in the configuration mode. See if the Node details are correct.

### 18. Multidrop not Supported

**Explanation :** Internal Fatal Error. An attempt was made to attach two nodes on the same driver when Multidrop is not supported.

**Action :** Recheck the project in the configuration mode. See if the Node details are correct.

### 19. Multiple nodes with same ID

**Explanation :** Internal Fatal Error. An attempt was made to attach two nodes on the same driver with same Node IDs.

**Action :** Recheck the project in the configuration mode. See if the Node details are correct.

### 20. Insufficient Memory for Request

### 21. Insufficient Memory for Request2

### 22. Insufficient Memory for Request Container

### 23. Insufficient Memory for Dummy Request

### 24. Insufficient Memory for Action

### 25. Insufficient Memory for Action Container

### 19. Cannot Create Communication Window

**Explanation :** Internal Fatal Error.

**Action :** Try making more memory available for the project.

### 20. Cannot Open Communication Port

**Explanation :** Internal Fatal Error. Could not initialize the Communication port for the given settings.

**Action :** For the selected Communication port, check for -

- ⊙ If the port physically exists.
- ⊙ If the Communication hardware uses standard base addresses. COM1 uses hex 3F8 and COM2 uses hex 2F8.
- ⊙ If there is any IRQ contention at the hardware level. COM1 uses IRQ4 and COM2 uses IRQ3.
- ⊙ If any other program is already using the Communication port you have requested for
- ⊙ If any DOS level TSRs are running which are using the Communication port you have requested for.



## Errors



- ⦿ If a mouse driver is installed on the same Communication port you have requested for in Windows environment.
- ⦿ If a mouse driver is installed on the same Communication port you have requested for on DOS environment.
- ⦿ If you have directly manipulated the PROJECT.INI file section [COM1] or [COM2], check if the settings for Baud Rate, Data Bits, Stop Bits and the Parity are standard. Try using the Communication port setting utility provided with Astra in case you are in doubts about the standard settings.

### 21. Cannot Build Communication DCB

**Explanation :** Internal Fatal Error. Could not initialize the Communication port for the given settings.

**Action :** If you have directly manipulated the PROJECT.INI file section [COM1] or [COM2], check if the settings for Baud Rate, Data Bits, Stop Bits and the Parity are standard. Try using the Communication port setting utility provided with Astra in case you are in doubts about the standard settings.

### 22. Cannot Set Communication State

**Explanation :** Internal Fatal Error. Could not initialize the Communication port for the given settings.

**Action :** If you have directly manipulated the PROJECT.INI file section [COM1] or [COM2], check if the settings for Baud Rate, Data Bits, Stop Bits and the Parity are standard. Try using the Communication port setting utility provided with Astra in case you are in doubts about the standard settings.

### 23.NULL Pointer for Model Names

**Explanation :** Internal Fatal Error.

**Action :** Contact Astra support.

### 24.Read Queue Full

### 25. Device Time Out

**Explanation :** The Device did not respond and the Device driver timed out. The Driver will retry the request to Device for a specified number of times and if the Device still does not respond the driver will HALT its transactions with the Device.

**Action :** If this happens during **initialization**, check –

- ⦿ Whether the Device power is on.
- ⦿ Whether the cable connections to the device are proper.



## Errors



- ⊙ Whether the Node ID settings are proper in case the Device supports it.
- ⊙ Whether the Device model is the same as configured in the Node Configuration.
- ⊙ Whether the Communication hardware is proper and works.
- ⊙ Whether strong EMI or RFI fields are existent which cause noise on the Communication line.
- ⊙ Whether some turnaround delay is required, try changing the entries in the DRIVERS.INI file. This may be typically required for faster PCs on which Astra runs.

If this happens during the **Run**, check –

- ⊙ Whether other applications block the Windows, in such a case the retry mechanism will normally re-establish the Communication.
- ⊙ Whether the cable connections have been disturbed.
- ⊙ Whether the Device has malfunctioned.
- ⊙ Whether the Communication hardware is proper and works.

### 26. Invalid IEEE Format

**Explanation** : The 32 bits read from the Device contained bit values such that it could not be interpreted as a valid IEEE format.

**Action** : Use OEM software and initialize floating type tags in the plc.

### 27. Write Queue Full

**Explanation** : The write request sent by the Astra is queued for faster execution, the current limit for the queue size is 300. If the queue is full this message will be prompted and the latest request will be ignored.

**Action** : Go to the project configuration file and put an entry with section name "QueueSize". Under this section name, put a key name "WriteQueue", so that it looks like: [QueueSize]

WriteQueue = WXY

Where,

WXY can be upto 5000.



## Errors



28. This error number is no more in use !!!

**Explanation :** Internal Fatal Error. The driver detected no valid tags in the project.

**Action :** Recheck the project in the configuration mode. See if any tags are assigned to this particular device. See if the Node details are correct.

29.No Valid Tags in the Project

30. Insufficient Memory for Register

31. Insufficient Memory for Tag Container2

32. Insufficient Memory for Register Container

**Explanation :** Internal Fatal Error.

**Action :** Try making more memory available for the project.

33. Tag Address Invalid

34. Tag Address Invalid2.

**Explanation :** The address entered for a Tag is invalid.

**Action :** Reconfigure the project and check.

35. Driver Scan Halted

36. Driver Scan Halted2

**Explanation :** The driver has stopped communicating with the device. This may happen in two situations –

⊙ When the initial scan is complete - in this case this is just a status information.

⊙ When time-out has occurred and retry for establishing communication has failed.

**Action :** In the second case check –

⊙ If the cable connections have been disturbed.

⊙ If the Device has malfunctioned.

⊙ If the Communication hardware is proper and works.

37. Cannot Find INI File Entry, Setting Default Port

**Explanation :** The [PROTOCOL] section in PROJECT.INI does not have the driver name against the COM1 or the COM2 entry. In such a case default COM1 is selected as the Communication port.

**Action :** Run the Communication port setting utility provided with Astra and set all the parameters properly.

38. Cannot Initialise Driver Twice

**Explanation :** Due to some abnormal termination in a previous run the Device Driver has not unloaded itself and hence could not reinitialise itself.

**Action :** Restart the project.



## Errors



39. Cannot Run Without Initialisation

40. Cannot Run Without Initialization2

41. Cannot Write Without Initialisation

42. Cannot Build Frames Without Initialisation

**Explanation :** Due to some abnormal termination in a previous run, the Device Driver has not unloaded itself and hence could not reinitialise itself.

**Action :** Restart the project.

43. NULL Pointer for Queue

**Explanation :** Internal Fatal Error.

**Action :** Contact ASTRA support.

44. Invalid IEEE Format2

**Explanation :** The 32 bits read from the Device contained bit values such that it could not be interpreted as a valid IEEE format.

**Action :** Use OEM software and initialise floating type tags in the device.

45. Cannot Pagelock Buffer

**Explanation :** Internal Fatal Error.

**Action :** Contact ASTRA support.

46. Device Response Delay

**Explanation :** The Device did not respond and the Device driver timed out. The Driver will retry the request to Device for a specified number of times and if the Device still does not respond the driver will HALT its transactions with the Device.

**Action :** If this happens during **Initialisation** check –

⊙ If the Device is powered on.

⊙ If the cable connections to the device are proper.

⊙ If the Device model is the same as configured in the Node Configuration.

⊙ If the Communication hardware is proper and working.

⊙ If strong EMI or RFI fields are existent which cause noise on the Communication line.

If this happens during the **Run** check –

⊙ If in case other applications block the Windows, in such a case the retry mechanism will normally re-establish the Communication.



## Errors



- ⊙ If the cable connections have been disturbed.
- ⊙ If the Device has malfunctioned.
- ⊙ If the Communication hardware is proper and working.

### 47. Response Check Sum Error

**Explanation :** The Device did respond but the bytes received were corrupt. The Driver will retry the request to Device.

**Action :** If this happens during **Initialisation** check –

- ⊙ If the Communication hardware is proper and working.
- ⊙ If strong EMI or RFI fields are existent which cause noise on the Communication line.
- ⊙ If the Communication port settings are proper.

If this happens during the **Run** check –

- ⊙ If the cable connections have been disturbed.
- ⊙ If the Device has malfunctioned.
- ⊙ If the Communication hardware is proper and working.

### 48. Data Over Flow

**Explanation :** Unexpected data in large volume was received on the Communication port.

**Action :** Check –

- ⊙ If the cable connections have been disturbed.
- ⊙ If the Device has malfunctioned.
- ⊙ If the Communication hardware is proper and working.

### 49. Model Name Invalid

**Explanation :** Internal Fatal Error. The model name associated with a particular Node was invalid.

**Action :** Open the project in the configuration mode. Check the model in the Node Configuration and close it.

### 50. Cannot Open File PLCTAG.DAT 2

### 51. Cannot Read File PLCTAG.DAT 2

### 52. Cannot Read File PLCTAG.DAT 3

**Explanation :** Internal Fatal Error. The input file PLCTAG.DAT does not exist or is corrupt.

**Action :** Open the project in the configuration mode and close it, this process recompiles the PLCTAG.DAT file.



## Errors



53. This error number is no more in use !!!

54. This error number is no more in use !!!

55. Invalid number for conversion to BCD for WORD

**Explanation :** The 16 bits given for write from ASTRA to the Device contained bit values such that it could not be interpreted as a valid BCD format. Write will not be done in these cases.

**Action :** Avoid such values.

56. Invalid number for conversion to BCD for DWORD

**Explanation :** The 32 bits given for write from ASTRA to the Device contained bit values such that it could not be interpreted as a valid BCD format. Write will not be done in these cases.

**Action :** Avoid such values.

57. Invalid number for conversion to BCD for WORD

**Explanation :** The 16 bits given for write from ASTRA to the Device contained bit values such that it could not be interpreted as a valid BCD format. Write will not be done in these cases.

**Action :** Avoid such values.

58. Invalid number for conversion to BCD for DWORD

**Explanation :** The 32 bits given for write from ASTRA to the Device contained bit values such that it could not be interpreted as a valid BCD format. Write will not be done in these cases.

**Action :** Avoid such values.

59. Error Composing Write Request

**Explanation :** Write request could not be composed. This may happen in two cases –  
⊙ Invalid number for write.

⊙ Write Queue full.

**Action :** Avoid non interpretable values, Avoid writing too fast. Increase WriteQueueSize in project.ini file section [QueueSize]. Lower limit for WriteQueueSize is 300 and the upper limit is 5000.

60. Error Composing Read Request After Write

**Explanation :** A read request immediately following a write request could not be composed. This may happen in two cases –

⊙ Invalid number for write.

⊙ Write Queue full.

**Action :** Avoid non interpretable values, Avoid writing too fast. Increase WriteQueueSize in project.ini file section [QueueSize]. Lower limit for WriteQueueSize is 300 and the upper limit is 5000.



## Errors



61. Node Failed.

**Explanation :** Internal Fatal Error. The Node was not able to communicate. In case of Multidrop DEVICE system the node id given to the nodes may be same or cable from PC to DEVICE may be faulty.

**Action :** Open the project in the configuration mode check the Node Configuration and close it. For Multidrop communication check the node ID. Check the cable.

62. Cannot Open File NODES.DAT

63. Cannot Read File NODES.DAT 2

**Explanation :** Internal Fatal Error. The input file NODES.DAT does not exist or is corrupt.

**Action :** Open the project in the configuration mode and close it, this process recompiles the NODES.DAT file.

64. Node set on by user.

**Explanation :** Not an error . It indicates that node is selected by the user. For ASTRA generated default tags for a DEVICE, in that if command tag is 0 then this message is displayed.

**Action :** None , as it indicates that node is selected by the user.

65. Node set off by user.

**Explanation :** Not an error . It indicates that node is unselected by the user. For ASTRA generated default tags for a DEVICE in that if command tag is 1 then this message is displayed.

**Action :** None, as it indicates that node is unselected by the user.

66. Node manager proc address not defined.

**Explanation:** Internal Fatal Error.

**Action:** Contact Astra support.

**Errors displayed (numbers are internal to program) Following errors are specific to OPC Driver.**

600 Invalid server key, entry not found in opc.ini.

**Explanation:** This type of an error occurs if the server field in the OPC.INI file is not assigned any value, i.e. the server key is not provided. This is required when client uses registry to find CLASSID of the server. For example:

```
[OPC1]
```

```
NODE=OpcNode1
```

```
SERVER= (The error is that the server name is blank)
```

```
TIMEBLAS=0
```

```
DEADBAND=0.0
```

```
REGISTRY=1
```



## Errors



**Action:** Make appropriate entry as above in the OPC.ini file.

601 Invalid server key, entry not found in Registry.

**Explanation:** This type of an error occurs if the server field in the OPC.INI file is assigned a wrong value, i.e. the server key is provided, but it is wrongly spelt.

For example:

```
[OPC1]
```

```
NODE = OpcNode1
```

```
SERVER = tt (The error is that the server name tt is non existent)
```

```
TIMEBLAS = 0
```

```
DEADBAND = 0.0
```

```
REGISTRY = 1
```

**Action:** Correct the entry in OPC.ini. Install the OPC Server program. The program will update the registry.

602 Connection to the server failed due to unknown reason.

**Explanation:** The operating system is Windows 95 and the DCOM is not installed. The other reason can be that the executable file has become corrupt. Server could not be started due to some reason such as incomplete installation, already running instance etc.

**Action:** Install DCOM if in Windows 95 or Service pack 3.0 under NT. If the problem persists contact the vendor who has supplied the OPC server. If you are using Windows NT check the user access rights, permissions. Check user login. Also refer to chapter 7.

603 Error disconnecting the server.

**Explanation:** One or more interfaces could not be released.

**Action:** This can also come due to abnormal termination of any program running. If the problem persists Contact server vendor.

604 Error performing read.

**Explanation:** Some faults in the physical connection. Normally this error will come if there exists a tag in ASTRA with address not existing in Server project. This will also come if the connection between Client/Server fails.

**Action:** Correct the Tag address and run ASTRA again. Ensure that tag address, name, group and all other properties of tag (in ASTRA as well as Server) are correct. Or check the network connection.



## Errors



605 Data Quality bad.

**Explanation:** The data quality received is bad. For example, if a server is reading a variable from the PLC and while reading, the PLC stops due to reasons like power failure. This can also occur if the connection between client and server fails due to some reason.

**Action:** Check the physical connections between OPC Server and device.

606 Data type returned by the server is not supported by ASTRA.

**Explanation:** ASTRA supports mostly all the data types like INT, UINT, FLOAT etc. However, if at all there is a data type that is not supported, then this error is generated.

**Action:** Check the data type for the item in server and change as per Astra.

607 Error performing writes.

**Explanation:** Some faults in the physical connection. Normally this error will come if there exists a tag in ASTRA with address not existing in Server project. This will also come if the connection between Client/Server fails.

**Action:** Correct the Tag address and run ASTRA again. Ensure that tag address, name, group and all other properties of tag (in ASTRA as well as Server) are correct.

608 Error adding group in the server.

**Explanation:** Errors in getting the interfaces. This can be due to improper installation of Server program or incorrect project settings.

**Action:** Check server programs settings, project settings.

609 Error adding item in the server.

**Explanation:** The access path is not provided as per the requirement. This can also occur if the tag properties, address is not matching with server project.

**Action:** Check project settings of client as well as server.

610 Error removing group in the server.

**Explanation:** One or more interfaces of the Server program could not be released properly.

**Action:** Check the access rights of the user.

611 Error removing item in the server.

**Explanation:** One or more interfaces could not be released.

**Action:** Check the access rights of the user.



## Errors



612 IP Address String entry empty for remote Server

**Explanation:** The IP Address String is Empty. If this string is empty the client will not know about server PC.

**Action:** Provide the IP Address of the OPC server PC.  
IPADDRESS = 192.10.10.1

613 Class ID for the server not found in INI file.

**Explanation:** There is no entry in OPC.INI file for CLASSID. This error can occur due to one of the following reasons.

1 > OPC Server program is not installed in Client or there is no entry in registry for the same.

2 > **REGITRY = 0** and there is not entry for CLASS ID.

**Action:** Either make entry in registry for the OPC Server or add the CLASSID information in the INI file. This is necessary only in case of remote server.

```
[OPC1]
NODE = OpcNode1
SERVER = KEPServer
TIMEBLAS = 0
DEADBAND = 0.0
REGISTRY = 0
CLASSID = (Missing)
```

614. Error in Class ID entry of INI file

**Explanation:** The CLASSID entry in INI file is not in specific format. The entry needs in the format mentioned in this document. However separator can be "-" or ".". This is so to keep compatibility with Registry entry.

```
[OPC1]
NODE = OpcNode1
SERVER = KEPServer
TIMEBLAS = 0
DEADBAND = 0.0
REGISTRY = 0
CLASSID = 46F74E887BFF11D29DE20020AF9BB61D
Separators as missing.
```

**Action:** Make the CLASS ID entry as per registry of OPC server. Use the proper separators between various fields. The should be dot (.) or dash (-) as a separator.



## Errors



615 Node Stopped due to BAD Quality Data received from all Tags , to restart Enable Node again

**Explanation:** All the tags of the node are giving BAD quality data or read has failed to this node. Due to this the node is SET OFF. To start the node user should again SET the node ON using NODE\_COMMAND tag. This can occur due to device failure or network failure.

**Action:** Check the communication between device and OPC Server. Similarly check the network connection.

**In case of Remote Servers there can be long delay between messages when there is error. If the server is connected and then later on connection fails due to some reason, in such case this delay can be considerable. User should check for Network connection or check if the server is working. In any case do not terminate Server Program forcefully. Otherwise you may face problem during next startup. In such case you should restart your OPC Server PC.**

**Normally, the Client program should terminate OPC Server program or who has invoked the server. If the OPC Server program is not terminated properly, then it might have locked the Serial ports. In such case the Owner of OPC server (login user with Full Access Rights) should terminate. Second instance of the object while first is running may give errors.**



Renu Electronics Pvt Ltd.

S.No. 2/6, Baner Road,

Pune 411045, India.

Tel: + 91 20 2729 2840,

Fax: + 91 20 2729 2839

Email: [info@renuelectronics.com](mailto:info@renuelectronics.com)

Website: [www.renuelectronics.com](http://www.renuelectronics.com)

