



White Paper

OPC Interface Discussion

July 2007

Background

Panther uses OPC as its main interface technology to third party systems to send and receive data at the mill level environment. OPC stands for “OLE for Process Control” and is a widely used international standard for data communication in the process industries. The standard was initially developed in conjunction with the advent of Microsoft OLE technology and the widespread use of MS Windows in the industry. The standard has allowed many different vendors to communicate their respective product data to each other.

The interface applications typically used by Panther include:

- Interface to PLC’s controlling winders, scales, roll lines
- Interface to QCS systems controlling paper machines
- Interface to QCS systems for profile data and reel turn-up events
- Interface to DCS process control systems for data retrieval
- Interface to lab test devices (when available)

Use of OPC reduces user cost for software, programming, custom code support and project time. Since all Panther product technology is delivered on Microsoft platforms, OPC is a natural interface medium for Panther. (Note: Any other interface method can be developed. Each version requires a customized proposal based on the supplied interface design and data mapping manuals).

The OPC standard is maintained by the OPC Foundation, a public non-profit organization that has a large number of equipment suppliers who are members. Documentation providing detailed descriptions of the OPC standard as well as example source code and downloaded software interfaces is available from the OPC Foundation web site at www.OPCFoundation.org.

Introduction

The Panther OPC Client is a configurable software interface application module that integrates Panther MillTools, TestCentral and ICIS systems to third party systems and products. Data connections are made between the third party and the Panther system from the Panther data base **using** SQL data structures. It is therefore the single method for transferring data to and from the (server) real time systems and the (client) Panther system.



White Paper

OPC Interface Discussion

July 2007

Three unique software modules must communicate properly in order for the OPC interface system to work. These are: the OPC Server, the OPC Client and the Panther application using the data. The user or a third party supplies the OPC server along with its respective user manuals. Panther supplies the other two items.

Program and Data – General Functionality

The Panther OPC client moves data between two areas, namely the OPC server and the Panther system SQL database. The general data path is as follows:

Typical “From” path Host Application Node (PLC, QCS, or other)---> Host OPC server--->OPC Physical link---> Panther OPC Client ---> Panther Database --->Panther Application

Typical “To” path Panther Database--->Panther OPC Client--->OPC Physical link--->Host OPC Server--->Host Application Node

(Note: the Panther ICIS system is an exception to this general path description. This system data path most often does not go through the Panther Database, but reads and writes directly to mapped tags. This is desirable for situations where data rates and response times to real world situations are critical.)

A typical data element is defined with a tag name and attribute. The naming conventions will change from system to system. OPC does not enforce any particular naming convention on all suppliers. For example "3basiswt.pv" is the name and attribute for a PM#3 basis weight profile. Individual elements in the array are addressed by index. For example, "3basiswt.pv(12)" is the 12th element of the basis weight profile. All the OPC array indexes start at "1", while the SQL indexes start at "0". The OPC link can handle any data type of array or single values. An array must be of the same type however. For example, arrays of dates, floating point numbers, flags, or strings may be used. The SQL database is organized by Tables and Fields within the table. The client will allow mapping from one set of data to another as well as handle most data type conversions automatically.



White Paper

OPC Interface Discussion

July 2007

OPC names are defined by the Server but are referred to by different labels depending on the location. For example, OPC Servers refer to Item Id's, or Items. Browsers refer to Leaf's, and DCS/QCS systems refer to Tag Names. **These are all the same thing.** Within the Client, OPC Item Id's are often called OPC Tag Names and are referred to using the entire Item Id plus the Group name. Example: if item id 13LABTST.PV has been added to group ""ManTests", the client displays that name as "13LABTST.PV(12)|ManTests".

It is possible for the same Item id to appear in more than one group. The OPC standard does not restrict adding the same item id to any group or groups. The client implementation, however, does not allow the same item id to appear twice in any single group.

The OPC Server does not maintain its own tags, but does maintain pointers or handles to tags in the real time system.

Appendix

OPC - What is it?

The following description is taken from the OPC Data Access Standard Version 1.0 A and is available in the Panther OPC Client directory under "Documentation". The OPC Foundation is also available on the Internet at www.OPCFoundation.org where all documentation is available for public use. Please refer to the OPC documents for a complete discussion of OPC. There are currently revision 1.0 and 2.0 interfaces available. The Panther OPC Client supports both interface standards. This section is taken from the document "OPC Custom Interface Rev 1.0a"

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.

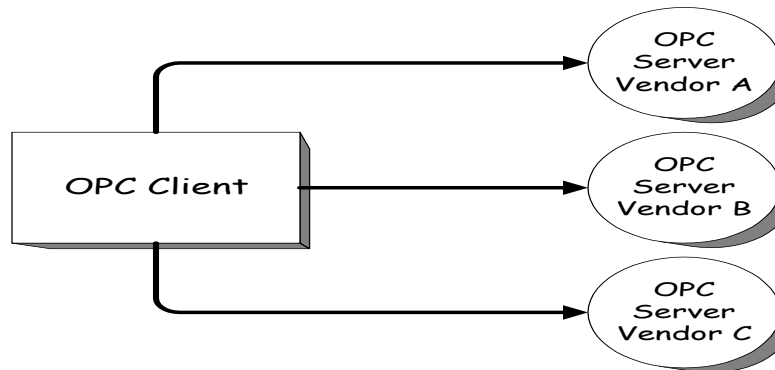


Figure 0-1 OPC Client

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.

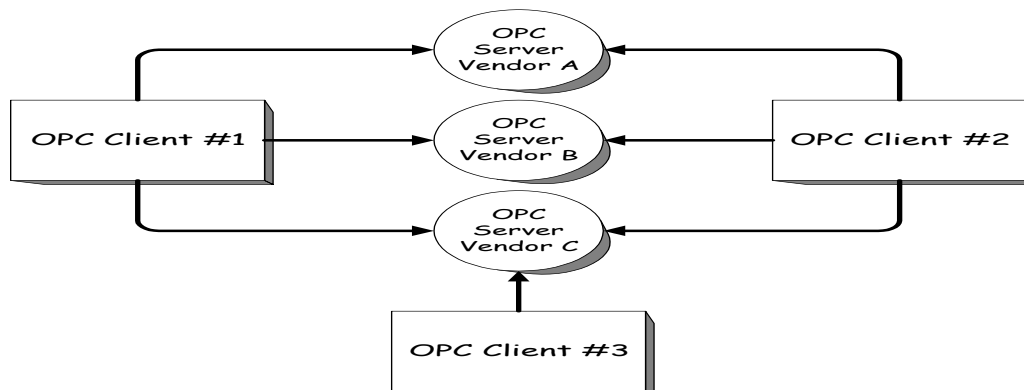


Figure 0-2 OPC Client/Server Relationship

At a high level, an OPC 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.



White Paper

OPC Interface Discussion

July 2007

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 OPC standard 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.

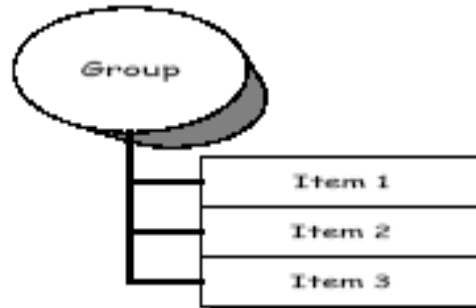


Figure 0-3 - Group/Item Relationship

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 a status condition.

Note that 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.