Technical Information

TI 36J04A10-01E

Exaquantum
Plant Information
Management System Overview

YOKOGAWA

Yokogawa Electric Corporation
2-9-32, Nakacho, Musashino-shi, Tokyo, 180-8750 Japan
Tel.: 81-422-52-8067 Fax.: 81-422-52-5994

TI 36J04A10-01E

2nd Edition Nov. 2000

Blank Page

Introduction

Exaquantum is a Plant Information Management System (PIMS) product that provides business benefits to users in the Hydrocarbons, Pulp & Paper, Power & Utilities, Chemicals and other industries.

■ This Document

This technical information (TI 36J04A10-01E) provides an overview of the Exaquantum system and shows standard features and system configurations of Exaquantum. It is intended for people who are planning to purchase a new Information Management System.

■ Notes on This Document

The purpose of this document is not to warrant Exaquantum is well suited to any particular purpose, but rather to describe the functional details of Exaquantum.

No part of this document may be transferred or reproduced without prior written consent from YOKOGAWA.

YOKOGAWA reserves the right to make improvements to this document and product at any time, without notice or obligation.

For any questions, mistakes or omissions detected in this document, please contact our sales representative or your local distributor.

■ All Rights Reserved

■ Trademarks

Exaquantum
Plant Information
Management System

TI 36J04A10-01E 2nd Edition

CONTENTS

Blank Page

1. System Overview

Exaquantum is a Plant Information Management System (PIMS) combined with a powerful user interface.

The operations within process-based industries are generally controlled by a Process Control System (PCS). A PCS typically produces large amounts of data that must be converted into information to facilitate management decisions and optimise the operation of the plant. Exaquantum processes data from the various systems within a plant, including the PCS, to deliver high-value business information to all decision-makers throughout the organisation.

This Technical Information document describes the functionality of Exaquantum. It is described in enough detail to provide engineers with a concept of the product, and provide adequate knowledge of Exaquantum so they know how users can make best use of the product.

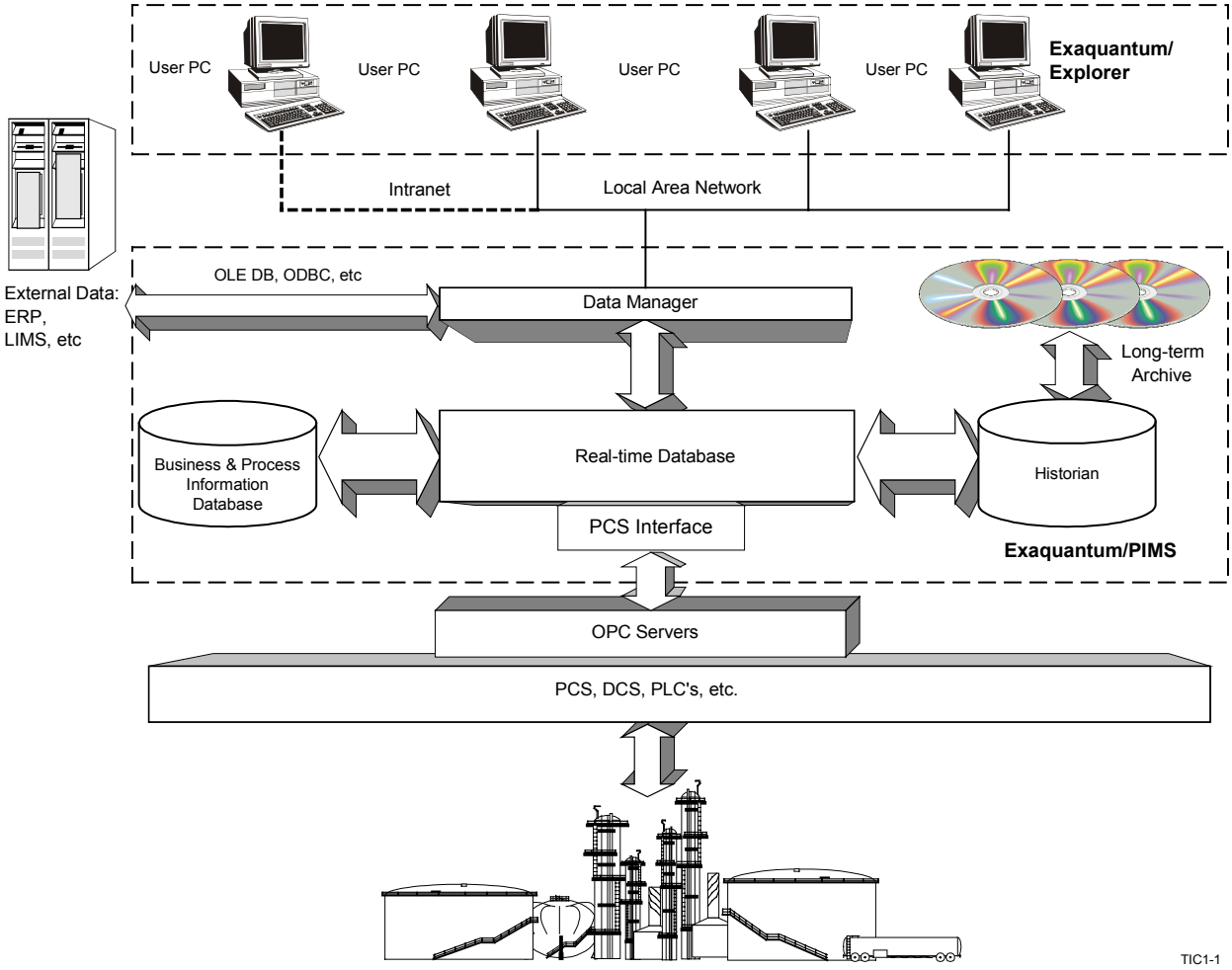


Figure: Exaquantum Overview

1.1 Benefits

Exaquantum is a powerful, easy-to-use solution that fulfils complex business information integration requirements.

Exaquantum provides visibility of key performance indicators (KPIs) using the ability to monitor and aggregate plant data, not just in process terms, but also in business terms. This all contributes to quicker decision making, improved quality and increased profits. Scalability and low engineering requirements mean that the system is quickly up and running and delivering results rapidly.

Both raw and historical data are made available for analysis. The analytical quality of Exaquantum means that both operators and engineers can be informed of potential problems, allowing them to alter plant operations before product quality is substantially affected. This maintains product quality, whilst reducing waste and rework. Historical data provides proof of compliance with standards and regulatory requirements.

Exaquantum/Explorer provides a Windows-based user interface that presents clear, easy-to-use and configurable displays in the familiar Microsoft format. Users can view existing displays, or build their own plant displays, data representation or analysis displays in the form of Exaquantum/Explorer documents. Exaquantum/Explorer provides an editing and reporting environment that is usable by non-programmers, but also has the power and flexibility of a powerful scripting language for programmers.

Exaquantum maximises the use of Microsoft's open standards. DCOM, ActiveX, OLE DB and ODBC allow easy integration with other Microsoft-standard tools and systems, irrespective of their function or who manufactured them. This benefits users as they can choose from the vast array of third-party components suitable for their needs and method of interfacing.

Exaquantum is supplied on CD-ROM, which also contains the set-up utility and User Manuals. The set-up utility performs the installation of products onto the PC with interactive guidance. The User Manuals are listed below.

Table: Exaquantum Document List

Document	Reference Number
Exaquantum Installation Guide	IM 36J04A13-01E
Exaquantum/PIMS User's Manual	IM 36J04A11-01E
Exaquantum/Explorer User's Manual	IM 36J04A12-01E
Exaquantum API Reference Manual	IM 36J04A14-01E
Exaquantum Engineering Guide – Administration	IM 36J04A15-01E
Exaquantum Engineering Guide – Network Configuration	IM 36J04A15-02E
Exaquantum Engineering Guide – Support Tools	IM 36J04A15-03E

1.2 System Components

The Exaquantum system components are shown in the following figure.

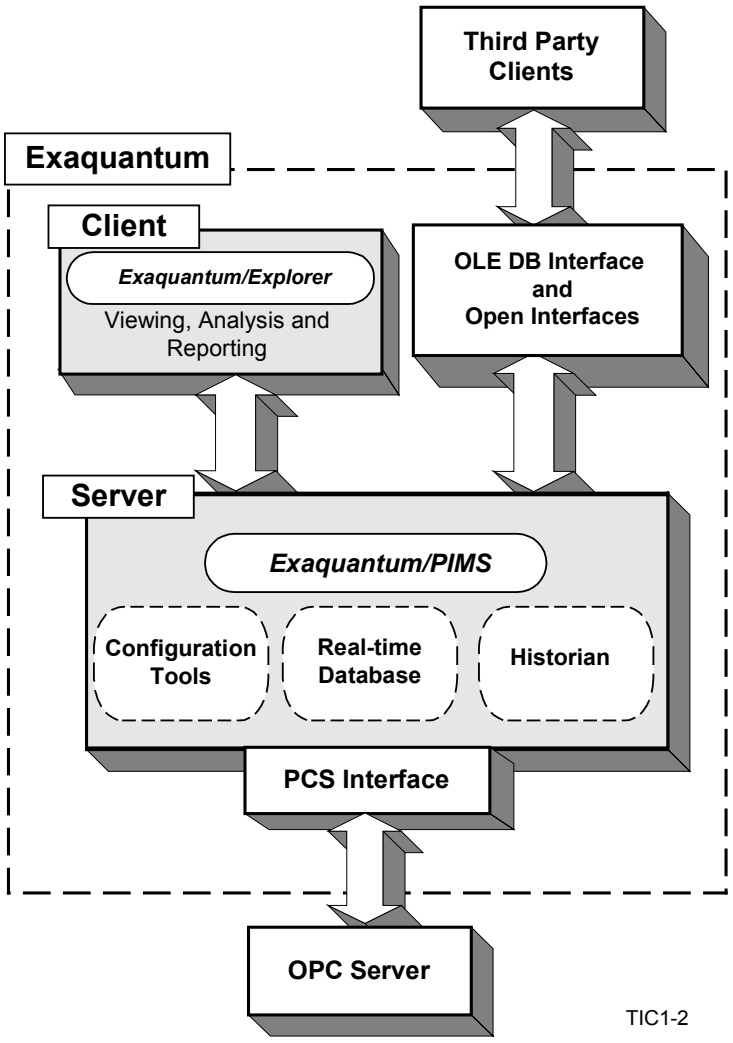


Figure: Exaquantum System Components

The system forms a multi-tier architecture, with OPC servers normally providing the raw process data. This data is first managed and accumulated within the Exaquantum/PIMS server. It then becomes available to Exaquantum/Explorer (a powerful analysis and reporting facility). Other third party clients via the OLE DB interface or open interfaces (application programming) can also access the data.

The main components of Exaquantum are described below.

■ Exaquantum/PIMS

The Exaquantum/PIMS server has the following three main functions.

● Real-time Database

The Real-time Database is a high performance real-time store for process and plant data that also provides flexible, user-defined calculations, and aggregations (mean, max, min, etc.) over multiple user-defined time periods.

- **Historian**

  An optimised, long-term Historian provides efficient storage and fast retrieval of vast amounts of plant data, over very long time periods.

- **Configuration Tools**

  A suite of easy-to-use tools is provided to build, deploy and manage the Exaquantum/PIMS environment.

Exaquantum/Explorer

Exaquantum/Explorer is a data utilisation client that can be run on user PCs. It is a powerful, flexible analysis and reporting environment through which business information may be presented in graphical displays and reports.

OLE DB Interface and Open Interfaces

- **OLE DB Interface**

  The industry-standard OLE DB interface (which includes ODBC) allows Exaquantum/PIMS to present PIMS information to third party clients. Since many of these tools provide users with 'non-programming' access (making use of such techniques as 'drag and drop'), this interface is suitable for users who have a requirement for the data without in-depth programming skills.

- **Open Interfaces**

  Exaquantum also provides a set of open interfaces by which application programs can browse, read and write configuration data. These are DCOM interfaces, suitable for application programmers.

1.3 System Features

■ Features of Exaquantum

Exaquantum is well positioned as a fundamental application platform for a wide range of solutions. The features of Exaquantum, which are described in subsequent sections, include:

- DCOM component-based
- Event-driven
- Full range of client functionality
- Simple and easy database creation
- Built-in aggregation and flexible scripted calculations
- Event Handling – Prompts user-supplied application programs to run upon receipt of Alarms and Events messages and internal events
- Access security control based on Windows standard
- International language support.

As Exaquantum is a native Windows solution, it can be deployed within the user's Windows network, and more general IT environment. In typical cases, Exaquantum can work as a gateway between the PIMS domain and office domain, keeping the two potentially different networks separate. Deployment options are described in Chapter 7 Deployment Options.

■ Features of Exaquantum/PIMS

Exaquantum/PIMS delivers comprehensive and varied features and functionality. These features are listed below and described in subsequent chapters.

● Integration with PCS

Exaquantum/PIMS provides support for a variety of PCSs through the OPC standard for process data transfer. Available PCSs and OPC servers are listed in Chapter 7 Deployment Options.

● Data Organisation and Structure

Comprehensive data organisation is achieved through simple and logical structures such as generic function block structures and flat tag structures. Grouping data using folders and populating databases by templates makes the structures even more manageable.

● Processing

Exaquantum/PIMS contains a fast, efficient real-time database (Chapter 2) for holding current process values and reference data (Description, Engineering Units, and Engineering Ranges).

There are also numerous flexible, user-defined calculations (including arithmetic and scientific functions) and aggregations (e.g. averages over 1 hour, 1 day, etc.) available.

Data changes and eventual processing take place in an event-driven way throughout the system. This ensures fast, efficient processing within Exaquantum/PIMS.

- **Historian**

  The high-performance Historian (Chapter 3) is an efficient data-storage mechanism. Its main function is to store and retrieve real-time process data, reference data, PCS alarms and events. The Archiving mechanism allows a virtually unlimited amount of information to be stored in the system. This is achieved by transferring part of the information between external media and the online storage.

- **Management and Configuration**

  A group of easy-to-use tools (Chapter 4) is provided to configure and administer Exaquantum. These tools use the Microsoft Management Console (MMC), fully integrated in the distributed Windows configuration management scheme.

■ Features of Exaquantum/Explorer

Exaquantum/Explorer is an easy-to-use component-based analysis and reporting tool. It has all the features and functionality required for monitoring the plant, and effectively analysing the data that is supplied to Exaquantum/PIMS.

Designed for the Microsoft Windows 95/98 and NT platforms, it provides the power to embed ActiveX technology into graphical displays. It has the same 'easy-to-use' features found in many Microsoft software packages.

The user interface screens in Exaquantum/Explorer are called documents. Documents can be created from a set of predefined graphics and controls (e.g. bar graphs, numbers and trend graphs) that present the plant information in an easy-to-understand clear way.

Exaquantum/Explorer operates in two modes: Run Mode and Design Mode. In Run mode, users can view and monitor the information contained in the documents that were created in Design mode. The number and type of actions that users can perform in Run mode depend on the attributes and functions added to documents in Design mode. Design mode is used to create and modify user screens (or documents) based on components available for configuring documents.

The following controls and features (described in Chapter 5) are available in Exaquantum/Explorer:

- Trends
- Data Entry Grid
- Data Write Back
- Report Times
- Alarms and Events viewing
- Web Browser
- Scripting
- Navigation
- Data Parameterisation
- Document Optimisation
- Embedded Documents
- Excel Add-in
- Excel Viewer.

Features of OLE DB Interface

Exaquantum/PIMS implements the OLE DB interface to allow users access to Exaquantum information through OLE DB/ODBC compliant client tools. The functions available include:

- Browse, for creating a list of tags
- Historian process values at a particular point in time, for a specified set of tags
- Historian process values for a period of time with a specified interval, for a specified set of tags
- Alarms and Events, for a period of time.

A wizard is provided that guides the users through the operation (using the 'drag and drop' technique); this allows users to easily specify times and tags.

Features of Open Interface

The open interfaces provide a way to access Exaquantum data through DCOM based interfaces. The two main purposes for this are:

- Application programming
- Standard data import/export tools based on these interfaces.

International Language Support

Exaquantum is NLS compliant. It is designed to run in any regional environment, and to support native languages. The basic concept of Exaquantum NLS support is:

- Windows NT/95/98 in the native language version is assumed.
- Any prerequisite software needed to host Exaquantum software is prepared in the Exaquantum CD-ROM, with the proper native language version. Such prerequisite software includes Microsoft SQL Server.
- Exaquantum 'look and feel' On-line documentation (and Help files) available in the native language.
- Yokogawa supplies Exaquantum in English and Japanese. It will operate in other languages, but the translation is the responsibility of the affiliates.
- Where a language other than English is to be used, a separate CD-ROM is prepared for the installation.

Once the software is installed, it behaves in accordance with the Windows regional settings.

Exaquantum supports Daylight Saving Time; the setting is retrieved from the Windows Date/Time Properties, and the software behaves accordingly.

Blank Page

2. Real-time Database

The real-time database provides a high performance real-time store for plant-wide data. It also provides flexible, user-defined calculations and aggregations (mean, max, min, etc.) over multiple user-defined time periods.

2.1 Organisation and Structure

Data for the many items in a typical production environment is acquired from various sources by the Exaquantum real-time database. To ease the task of managing and using such a wealth of data, Exaquantum provides powerful ways to organise the data appropriate for the customer's own business needs. The primary components within the system are:

- Folders
- Function Blocks
- Tags
- Items.

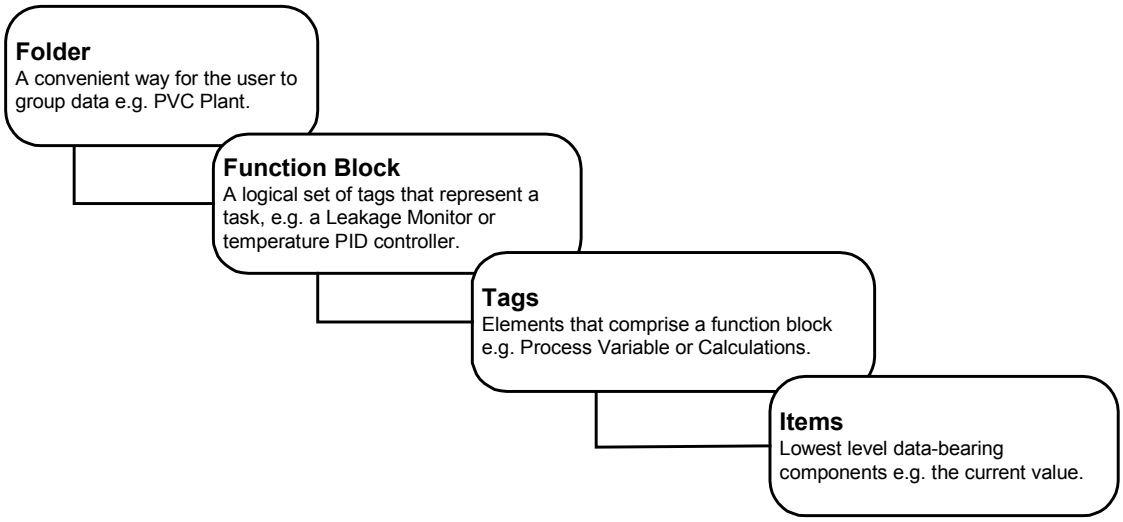


Figure: Relationship between Folders, Function Blocks, Tags and Items

Exaquantum tags are not the same as standard PCS tags. They are similar to CENTUM data points, but include aggregation and reference data as well.

Exaquantum also contains ‘Flat tags’ that do not form part of a function block and are located directly under a folder.

■ Folders

Folders can be used to break the plant up into logical functional units. The multi-level folder allows users to locate most items as required. The folder name will appear as part of the access string for all items in the folder. Folders can be created manually by users to group related data. A folder works in a similar way to a Function Block, in terms of grouping tags. However, in other respects it works differently. For example, creating a function block results in creating member tags too, whereas in the case of a folder, nothing else is created.

Function Blocks

An Exaquantum Function Block is a logical set of tags meant to represent a task or information. The Function Block name will appear as part of the access string for all items in the function block. Each function block must be associated with a data structure called a Function Block Template, which consists of:

- A name ('Function Block Name')
- A number of associated tag(s), each of which has a logical name and a tag template (from which it was created). For more information about these templates, see Chapter 4.

Two typical uses of function blocks are described below.

Mirror PCS Data Structure

Wherever a PCS has a functional and event-based data structure, such as PID controller, it consists of a number of process parameters. The PID controller, for example, may have PV (process variable), SV (setpoint value), MV (manipulated variable) and others like P, I, D, or MODE. The user may want to mirror a part or all of such a structure within Exaquantum and its Historian. In this case, the Function Block is used to group the process parameters to form a logical set.

Represent a Logical Functioning Unit

It often happens that the same functioning structure appears repeatedly within a system. For example, wherever a pipe diverges into two, flow balance might be calculated. In this case, the 'functioning unit (structure)' can be defined as one in-flow process value, two out-flow process values, a formula to calculate balance, and a constant to store an alarm limit of deviation. Since this structure may appear many times across the system, it should be defined as a logical unit to represent the structure. This facilitates creating a new instance, changing an instance, or deleting an instance. The Exaquantum Function Block represents such a logical unit.

Another example of a function block is for differential summation. In order to calculate flow quantity in an accurate way, one commonly used practice is to generate accumulated summation by the PCS and then Exaquantum calculates the summation by subtracting the current summation from the last value. The best data structure to represent this function consists of the following set of elements:

- PCS accumulated summation value
- A temporary storage to store the last value
- A constant value to define the range of the accumulated summation, as the value wraps around the range
- A constant value to define deadband value
- A calculated summation value.

A function block can form a unit with these elements. The user can then easily create new differential summation points, modify parameters or change calculation formula all at once.

■ Tags

A tag is the basic unit of the Exaquantum data structure. It is configurable, and the settings are categorised as:

- Structural settings
  - What reference data to associate with
  - Which aggregations to perform.
- Property settings
  - Type of tag: OPC, Manual or Calculated
  - Data type: integer, string, floating-point, etc.
  - Nature of tag: continuous or discrete
  - OPC settings: OPC update rate and percent deadband
  - Historian notification rule: none, on-change or time (specific frequency)
  - Expression of calculation.

A typical tag may consist of a number of items/tags, as shown below:

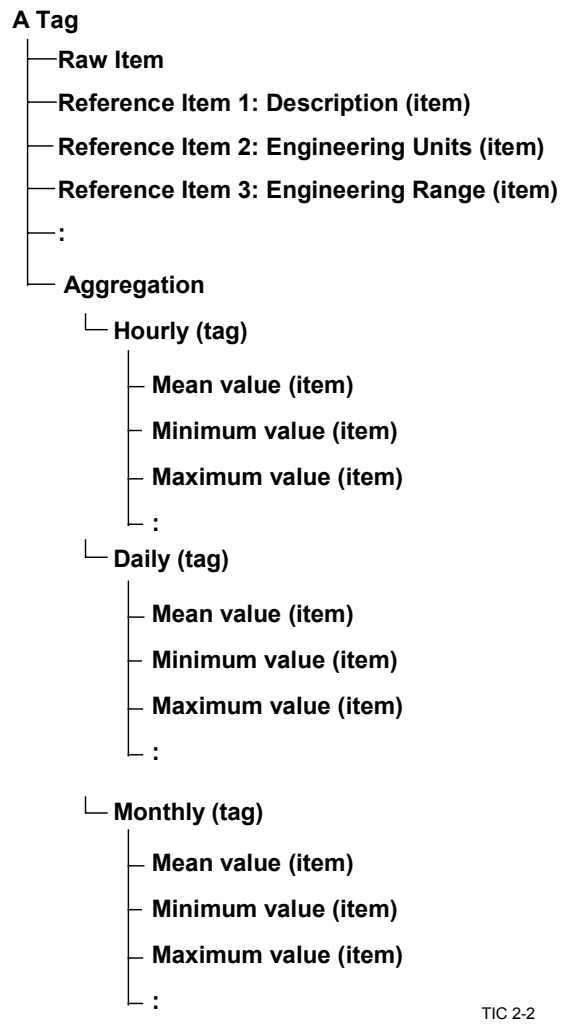


Figure: Typical Tag Data Structure

'Aggregation' is just a label – it does not have any substance.

Since a tag's elements are configurable, the complexity of the structure depends on the definition. Whenever a tag is discussed without a detailed definition, a 'standard configuration' is referred to. For example, Exaquantum performance is measured based on a standard configuration of tags. The following configuration is considered as this standard:

A tag with eleven (11) items:

- One raw item
- Four (all) reference items
- Six aggregation items (two aggregation period times three calculations for each).

The tag name will appear as part of the access string for all items in a tag. The tag name allows a 'shortcut' to be used. The intended use and characteristics of a shortcut are as follows:

- One typical usage of a shortcut is to refer to mirrored tags within function block tags. This is natural because the former type of tags is meant to be referred to by other tags.
- When a tag is first created, an instance of the tag is created; this results in History memory being allocated for the instance. Whenever a shortcut is made for such an instance, no more instances are created.
- There is no significant distinction between an original instance tag and a shortcut tag. Both shares the same 'instance' (memory allocated for the tag) with different names. Therefore deleting either of the tags is identical; it is not the case that deleting an original tag removes memory allocated for that tag. When all tags referencing an instance of the tag are deleted, the instance (memory allocated for the tag) will be removed from the system.
- A shortcut is only allowed for OPC tags. This means that an OPC tag can reference another OPC tag.

Only function block OPC tags can use shortcuts; flat OPC tags must always be instance tags.

■ Items

The item is the smallest data-bearing component in the system. All items have a:

- Identifier
- Value
- Timestamp
- Quality code.

The Identifier is different in nature from value, timestamp and quality. It has no memory allocated to it, as it is simply a name; the others have memory spaces allocated to store their representation.

- **Identifier**

  The Identifier is a string by which the subject item is addressed. It always starts from the *Root* folder down to the subject of interest, each of elements being delimited by a dot (.).

  These are typical identifiers:

  - Raw item

    `Root.Folder.Tag.Value`

  - Reference item (Description in this example)

    `Root.Folder.Tag.Description`

  - Aggregation item (value)

    `Root.Folder.Tag.Aggregation.Hourly.Mean.Value`

  - Aggregation reference item (Description in this example)

    `Root.Folder.Tag.Description`

  The Aggregation reference identifier differs from that of Exaquantum Release 1.0 (R1.0). In R1.0, the string 'Root.Folder.Tag.Aggregation.Hourly.Mean.Description' was allowed, even though the instance it referenced was exactly the same as 'Root.Folder.Tag.Description'. Because of this, and to eliminate a potential misunderstanding that the aggregation might have a separate reference data (e.g. Unit), R1.0 reference data addressing has been removed.

- **Value**

  Value is the fundamental item of a tag. It is the data source from which the value originates that determines a tag's type:

  - OPC tag: the value comes from the OPC server (or from the PCS via the OPC server)
  - Manual tag: the value comes from an external unsolicited source such as human entry, ASCII file import, ODBC/OLE DB, or other system through data access API
  - Calculated tag: the value is generated by the Exaquantum Calculation mechanism.

  OPC data is transferred to Exaquantum in an event-driven way. The transfer of data is further limited by the configurable update rate and percent deadband. The update rate is the minimum time between updates. The percent deadband is specified by a percentage of value range, changes within which do not result in data transfer.

  All three types of tags can have aggregations. Unlike a tag's original raw values, which are event-driven in nature, aggregations are time-dependent.

- **Timestamp**

  The timestamp is used for assigning the value to history and aggregations, allowing them to be sorted. It is therefore very important how a timestamp is assigned for values.

  The source of the timestamp for each tag type is as follows:

  - OPC tag timestamp is assigned either by the OPC server or by Exaquantum. Whenever time synchronisation is implemented, it is recommended that OPC server time be used, as this is the time that is closer to the source. However, when there are more than one OPC servers that are not time-synchronised, Exaquantum time should be used. Both options can be set up with registry keys.
  - Manual tag timestamp is assigned by the program that writes the value. The Tag Editor (an Exaquantum configuration tool) does not allow a timestamp to be entered which results in the system time is used at the point of update.
  - Calculated tag timestamp is assigned by Exaquantum when the calculation takes place.

  Exaquantum assigns an Aggregation (an attribute of a tag) timestamp whenever the value is updated.

  Time synchronisation between Exaquantum and the OPC server is important because the raw value timestamp is used in aggregation calculations that take place at Exaquantum time.

  Time resolution is always ‘seconds’, although the OPC specification defines a 100 nanoseconds resolution interface. Time is rounded to seconds because most client programs are based on second precision, and therefore values with a finer time resolution may cause confusion. For example, if a user shows a list of raw values, and performs a summation with Excel, the summation will look incorrect because Excel rounds the times to ‘second’ resolution. In this case, the calculated summation would be correct if the times were not rounded to seconds. The preference, therefore, is to allow Excel (and other similar tools) not to round the results.

  Whenever the time resolution is of particular concern, for example the time of event messages, Exaquantum does not round such time to seconds.

- **Quality**

  The quality code contains 16 bits of information. The lower 8 bits represent the OPC status (Primary Quality); this status can be GOOD, UNCERTAIN or BAD. The higher 8 bits represent a sub-status (Secondary Quality) that qualifies the OPC status.

  How the status UNCERTAIN is interpreted, and hence used, is often vendor dependent. Exaquantum applies the following policy:

  “Uncertain state is the best guess of value”; this means that values can be used with an understanding that it is not perfectly accurate.

  Reflecting this policy, the interpretation of quality in an Exaquantum/Explorer trend is as follows:

  - Uncertain points are shown in the graph as GOOD points
  - Uncertain-Shutdown points are treated as BAD, resulting in no graph beyond this point.

Assigning Quality

The logic of assigning quality depends on the type of items:

- OPC tags inherit the quality code provided by the OPC server (except BAD-OPC err, which is assigned by Exaquantum when the communication fails).
- Manual tags accept the source specification along with the value.
- Calculated tags – the default is determined as follows:
  - If any source item is BAD, then the result is defaulted BAD
  - If all of the source items have GOOD, then the result is defaulted GOOD
  - In all other cases the result is defaulted UNCERTAIN.

  The calculated tag code can explicitly override the normally default quality code. If an error occurs while executing the expression, BAD is always assigned.
- Aggregation items

  If all the input values within the period have GOOD quality, then the result is GOOD. If the period of BAD quality exceeds a system-wide defined period (%GOOD period), then the result is BAD. In other cases, the result is UNCERTAIN.

UNCERTAIN is a kind of GOOD in terms of calculating the %GOOD period. For example, if the entire values for an aggregation period are Uncertain, the resulting aggregation has a quality code Uncertain, not Bad.

Quality at Shutdown and Start-up

At shutdown and start-up, quality codes are allocated as follows:

- OPC tag. Uncertain-Shutdown is assigned. Soon after a subsequent start up, the value is recovered by OPC value.
- Manual tag. Preserved across the period when Exaquantum is shut down.
- Calculated tag. Uncertain-Shutdown is assigned. Soon after a subsequent start up, the value is calculated and recovered.
- Reference items. Preserved across the period when Exaquantum is shut down.
- Aggregation items. At a subsequent start up, the aggregation value is calculated based on the values saved at the time of shutdown, with a special quality code of Uncertain-Estimate. This potentially results in the correct value for the period interrupted by the shutdown and BAD value for the entirely missed periods.

Quality of History Information

There are several cases where the quality of history information is affected, for example:

- When an archive (of historian) is created and put offline, the missing period will be marked BAD
- When a system database is restored with a backup made some time before (catastrophic recovery), all values from the end of database up to the time when the database is restored will be marked BAD.

Flat Tag Structure

The ‘flat tag’ structure is used to locate tags directly under a folder. Unlike function block tags, a flat tag has no relation to other tags when one is created, modified or deleted. This structure is inherited from R1.0 of Exaquantum, and is preserved for the following potential usage:

- To replicate non-structured PCS tags

  Some types of PCS, such as a PLC, may not have a structured functioning unit. In this case, flat tags may be used to replicate PCS tags in Exaquantum. Other Exaquantum logical units, like function blocks, may refer to such replicated tags by way of a ‘shortcut’. The benefit of this approach is that PCS data can be configured to acquire data and to store historian data, even before the usage is designed and implemented in the form of function blocks.

- To allocate constants

  If their usage is system-wide, it may not be practical to configure constant values (represented by Manual tags) in function blocks.

Usage of flat tags is not limited to those mentioned above. Generic Function Blocks may also use flat tags.

Function block tags can reference not only flat tags but also other function block tags, by way of ‘shortcuts’.

2.2 Calculations and Aggregations

■ Calculations

Exaquantum Calculated tags can contain calculations ranging from simple, equation-based calculations (e.g. Result = SQR(TagA + TagC)) to complex, scripted calculations using VB scripting.

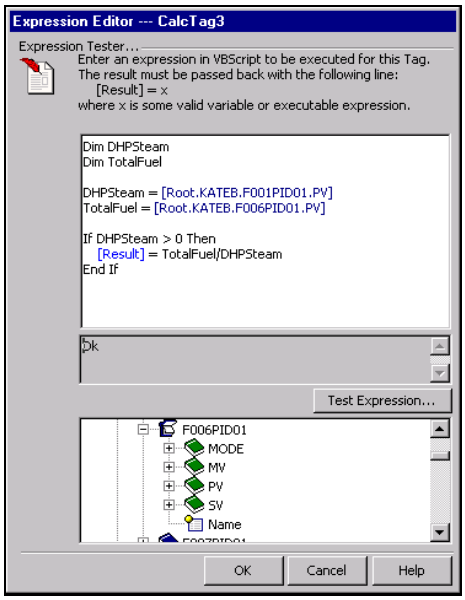


Figure: Example of a Scripted Calculation

The results of the calculations are no different from 'raw' process values; they may also be historised, aggregated, and used in other calculations. Exaquantum calculations are event-driven. When a new value arrives from a data source, the real-time database decides whether any derived items need to be updated as a result of this new value (on the basis of the dependency relationships Exaquantum maintains).

Exaquantum provides a number of useful features:

- Aggregation items can be used in calculation expression(s). (The value referenced in the expression is always the last calculated and completed aggregation value.)
- A result can be written to other OPC and/or Manual tags by specifying:

  `[Result.[Root.Folder.tag1.PV]] = value`
- Quality code can be explicitly set by the expression logic by specifying:

  `[Result].Quality = QualityHelperFunction`
- A relative addressing of the Result is available in the form of:

  `[Parent.[Tag1.PV]] = value`

The primary usage of calculations and expressions is within function block calculated tags. The use of relative addressing allows such a function block to be available wherever it is created or even moved after being created.

'Parent' is typically a function block that contains the subject calculated tag, but it could also be a folder.

Aggregations

An aggregation is a special type of calculation that is carried out over a pre-defined period (aggregation period), according to a specified rule (aggregation method).

Aggregation Periods

A tag can have aggregations defined for one or more Aggregation Periods; for example, it may have both an hourly aggregation and a daily aggregation.

Aggregation periods are defined by a time period and an offset. This allows definitions such as:

- 'Every day at 2 am'
- 'Every hour at 30 min after the hour'
- 'Every 15 min at 0, 15, 30 and 45 minutes after the hour'.

Some aggregations are calculated based on raw values, while others are calculated based on other aggregation results.

The features of the various aggregation periods are summarised in the table below.

Table: Aggregation Periods

Aggregation period	Description	Time Period	Offset	Can be derived from
Hour	Once every hour	Fixed (60 mins)	0 to 59 (min)	Raw, Custom
Day	Once every day	Fixed (24 hours)	0 to 23 (hours)	Hour, Raw, Custom
Month	Once every month	Fixed (1 month)	1 to 28 days, or 'Last Day'	Day, Hour, Raw, Custom
'Custom' type	User-defined periods e.g. 8 hours as a Shift	Limited set of time periods (max. 12 hours)	Time period' minus 1 minute	Raw (fixed)

If the 'Automatically adjust clock for daylight saving changes' box is checked in the Windows Date/Time properties of Control Panel, Daily aggregation has these characteristics:

- Calculations on daylight saving changes days are for 23 hours or 25 hours accordingly
- Aggregation calculation offset stays at the specified local time over daylight saving changes.

Aggregation Methods

Two Aggregation methods (Discrete and Continuous) are standard in Exaquantum.

Selecting ‘Discrete’ as the aggregation method will display the following check boxes and field:

- Count
- On Time
- On State (this is used to define any single state or value you want to count or time).

Selecting ‘Continuous’ as the aggregation method will display the following check boxes:

- Mean
- Minimum
- Maximum
- Standard Deviation
- Summation
- Spot Value.

In order to calculate quality, an aggregation method called %GOOD is defaulted whenever particular aggregation methods are selected. These methods are On Time for Discrete, and Mean, Standard Deviation, or Summation for Continuous.

2.3 Additional Real-time Database Features

2.3.1 Alarms and Events

Exaquantum obtains alarms and events from the PCS (where supported) via OPC Alarms and Events (A&E) servers. Alarms and events are historised, and are available for viewing through Exaquantum/Explorer. Filtering exists so only the A&E messages of interest are presented as the Exaquantum/Explorer control allows the user to set up rules to filter in (or out) the wanted (or unwanted) message types.

In practice, however, users have no exposed way of configuring filters. Therefore all messages are always defaulted.

2.3.2 Event Handling

Exaquantum can provide a means to activate application programs when:

- Any A&E messages are received from the PCS
- All aggregation calculations for an aggregation period are complete
- A calculated tag expression sends an event.

The event processing is implemented by the user as an external program (most likely written in VB language) to read the Microsoft Message Queue (MSMQ) for prompting messages, and process them accordingly. Exaquantum provides a user interface program that registers events to execute application programs, defines parameters to be passed to application programs, and selects which MSMQ the message is directed to.

Typical uses of the Event Handling function, and thus application programs, include (but are not limited to):

- Triggering reports at the end of an aggregation calculation
- Doing batch-oriented tracking or processing.

Event Handling works only on an Exaquantum server.

2.3.3 Data Write to Control System

Exaquantum has the ability to write PCS values through the OPC server.

When an OPC tag is configured to ‘write-able’, writing to such a tag results in further write access of the value to the associated PCS. An Explorer client, calculation expression, or application programs that write values through Data Access API can also trigger write operations.

To ensure that data write takes place properly, it is often necessary to configure the PCS and/or OPC server so that the PCS accepts data write requests. Exaquantum security controls (described next) are available to restrict the ability to write to control systems.

2.3.4 Access Security Control

Just as a PCS provides access security control, Exaquantum provides an access security control mechanism that is designed to be Windows-compliant:

- General resources are protected in the standard ‘Windows’ way, e.g. User Account management and DCOM security settings
- Exaquantum users are granted a particular set of rights, depending on the user account assigned as being a member of certain Groups (e.g. Exaquantum/Explorer Design Group, Data Write Group, Administrator Group are identified by Exaquantum).

Blank Page

3. Historian

The Historian is a powerful, efficient and fast data storage and retrieval component that:

- **Stores real-time process data, aggregated data, and reference data**
- **Stores PCS alarms and events**
- **Historises all process data with Identifier, Value, Timestamp and Quality**
- **Offers secure long-term archiving.**

Clients can also make use of the Historian to retrieve information for a particular period, and then perform various types of analysis on it whose output is generally in the form of reports.

3.1 Historian Data Categories

The Historian can store any RTDB data under the following categories:

- **Raw Data**

  These are OPC tag values, Manual tag values, and Calculated tag values. They can be of any data type that Exaquantum recognises; integer, long integer, single-precision floating-point, double-precision floating-point, or string.

- **Aggregated Data**

  Similar to raw data except that data types long integer and double-precision floating-point are used for numeric values.

- **Reference Data**

  Description, Engineering Units and Engineering Ranges. These comprise a separate category because the rate at which data changes is significantly slower than in the other categories.

- **Alarms and Events**

  A structured set of data elements.

The Historian allocates different database tables for these different data categories.

Data can also be stored in tables depending on other characteristics:

- Same raw historian interval (interval of periodic update rate, or on change)
- Same data type (integer, etc.).

3.2 Historian Configuration Parameter

The Historian allows some configuration that is tailored to the user's usage of the Historian.

Users can choose from the following types of raw historian interval, on a per tag basis:

- None
- Time (set of selectable frequencies)
- On change.

The raw historian interval is not the same as the OPC update rate. Therefore if 'Time' has been selected, any changes (to the OPC update rate) that are a finer resolution than the raw historian interval are not stored in the Historian, even though the actual information shows changes.

Periodic time does not result in periodic historian storage. Information is stored only when the value has changed since the last Historian update.

Regardless of the type of raw historian interval selected, system-wide events such as start-up and shutdown cause values to be stored. In these cases, the information is allocated an appropriate secondary quality code.

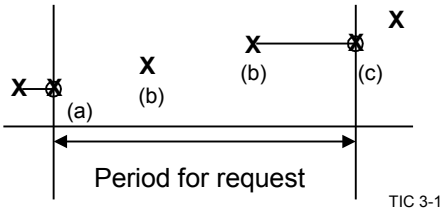
3.3 Interpolation

Because of the event-driven nature of Exaquantum, it is unlikely that values are actually stored in the Historian for the exact time that users want to look at. As a result of this, interpolation of data takes place. There are number of ways of interpolating data, including:

- Cast-forward : the previous value is extended in the future direction
- Cast-backward : the following value is extended in the past direction
- Linear : points are connected with straight lines
- Other more sophisticated ways, including Exponential.

In Exaquantum, interpolation is carried out either by clients, or by the data access layer that resides between the clients and server. It relies heavily on the way in which information is stored in the Historian. The Historian is tuned so that it can support Cast-forward only; there is therefore a certain limitation in other interpolation methods.

From the client viewpoint, the following figure illustrates interpolation in Exaquantum:



TIC 3-1

In this example, as a result of interpolating four points, point (a), multiple points (b), and point (c), will be passed to client; points (a) and (c) having been added by the data access layer.

When clients (Exaquantum/Explorer) are using the Trend control, they can choose one of two ways to interpolate data, either Cast-forward or linear. Anywhere else, only Cast-forward is supported.

If a request is made for future time (i.e. if the end time is in the future), the interpolation is made so that a BAD point is inserted at Now. This results in disabling any line that may otherwise appear in the Exaquantum/Explorer trend in the future time period.

3.4 Historian Timestamp and Relevant Processing

The Historian assumes that a value's timestamp is UTC; this is true for every item. The Historian stores values according to the UTC time series, and uses it as a key to identify stored values for read access. Two processes that are heavily dependent on the timestamp alignment issue are described below.

■ HMS (Historian Memory Snapshot)

Such a timestamp is as made at the source time, not when the request to the Historian was made, nor when the Historian received it. There could be a considerable difference between the time of receipt of a request, and the value's timestamp. The Historian tries to write information based on the value's timestamp. Because of this, there is a mechanism by which an amount of information is kept in memory for a certain time period, greatly reducing the need for frequent disk I/O.

■ Late-arriving Historian Data

The Historian makes use of RTDB technology (SQL Server), where the database schema is designed so the read operation is done most efficiently. In order that the search operation can take place less frequently, information for a certain period of time is combined in the form of a Binary Large Object (BLOB), and put into the database. This ensures high performance during read operations but low performance whenever there is a need to write to a past time.

In most cases, the system is tuned so that requests to write values to the past time period rarely happen. However, in those cases where requests (often referred to as late-arriving data) do happen; this results in a high CPU load for a period of time. Such cases include:

- LIMS data entry. Values are entered into Exaquantum some time after the sample timestamp
- A backward change in server time. A backward time change is likely to result in data loss whenever UTC duplicates. Therefore Exaquantum does not expect any large amount of time drift, especially backwards.

3.5 Archiving

Archiving is the process used to move part of the current Historian information to a separate archive that is to be handled separately from the online Historian database.

For example, an archive can be backed up by an external medium or it can be removed from the disk to make more space for the online Historian database. The backed up archive can then be restored into the disk so that it is seamlessly available, together with the online database, by Exaquantum. With this mechanism, the Historian can allow a virtually unlimited amount of its information to be available to the user.

An archive can be made on an Archiving Group basis. The groups are:

- Raw
- Aggregations (containing sub-types ‘Daily’ and ‘Monthly’)
- Reference
- Alarms and Events.

Raw (and possibly Alarms and Events) archives are likely to be used for analysing production, while aggregation archives might be used to (re)produce reports.

The user can control the size of an archive based on both the size of the media and the number of days to cover.

- Media size is meant to restrict the size of the archive to fit into the medium with which the archive will be backed up. This is a physical requirement that must be met before the following one is addressed.
- Number of days is meant to satisfy usability requirements. For example, where the user wants to make an archive on a time basis that is appropriate to his operational or management requirements (such as on a monthly or quarterly basis).

Where ‘media size’ and ‘period selected for archive’ are used together, media size always takes precedence.

3.6 Historian Management Tool

Exaquantum provides a tool to manage the Historian database. Typical cases that this tool will primarily address are when the disk gets full and when time is accidentally changed to the future. For further information, see Chapter 4 Management and Configuration Tools.

Blank Page

4. Management and Configuration Tools

Exaquantum is easy to install, configure and commission. This is achieved through a powerful combination of a ready-to-run platform and a comprehensive set of tools. The ready-to-run platform contains most of the infrastructure and data support required for the rapid deployment of large-scale integrated information systems. The set of tools, used for both management and configuration purposes, are integrated into a single system administration tool known as the Management Console.

The following Exaquantum tools are available from the Management Console:

- System Configuration
  - OPC Gateways
  - Production Calendar.
- Tag Configuration
  - Tag Template
  - Function Block Template
  - Tag Generation (four ways of generating function blocks/ tags)
  - Tag Editor.
- Database Management Archiving
- Event Handling
- Help and Support.

With these tools, users can carry out the following tasks:

- Define OPC servers
- Generate function blocks/ tags
- Show and edit function blocks/ tags
- Add folders
- Manage archives
- Define events that run application programs.

Certain tools used by Exaquantum are for system administration. These are available from within Windows NT (for more information refer to Microsoft Windows NT Help):

- Windows NT Tools
  - User Manager
  - Backup and Restore
  - Performance Monitor
  - Event Viewer.
- Microsoft SQL Server Enterprise Manager, and other tools for database administration.

Some important concepts for configuring Exaquantum are described in the following sections.

4.1 Management Console

The Management Console interface enables access to all Exaquantum Management and Configuration tools. It is based on the Microsoft Management Console (MMC), ensuring that Exaquantum tools are handled in a manner consistent with the administration tools of many Microsoft products, e.g. SQL Server.

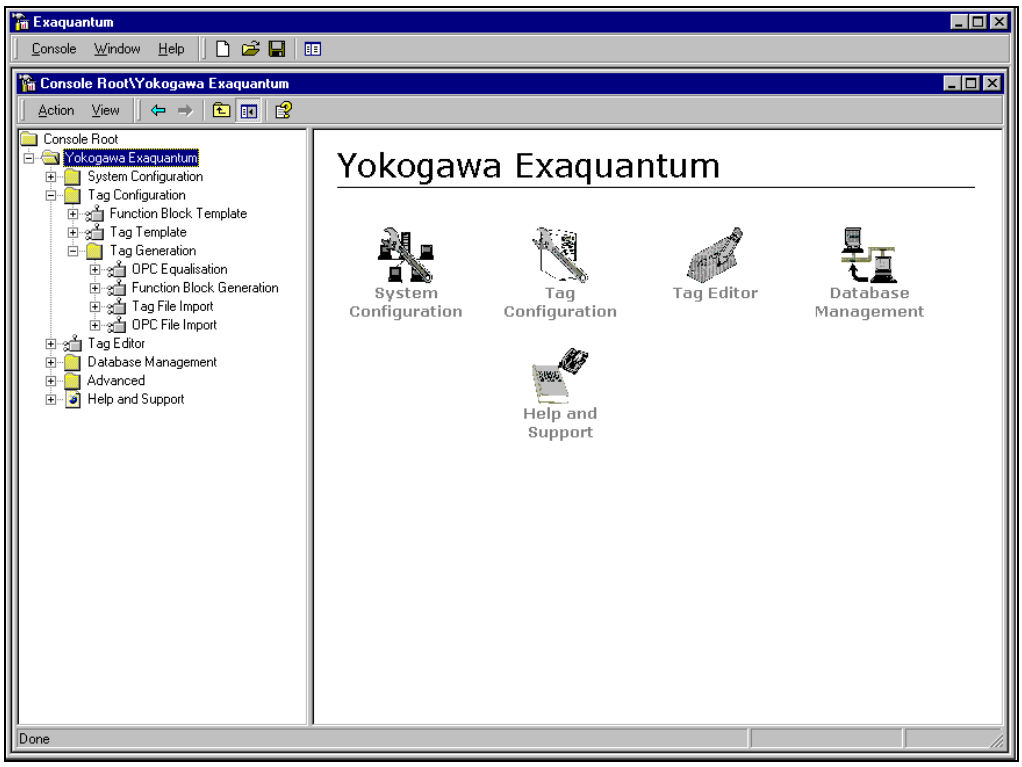


Figure: Exaquantum Management Console

The tools available within Exaquantum are displayed in the left part of the screen as a tree view. The user can expand each group to see the related sub-groups and/or the tools.

When a group is selected, the sub-groups and/or tools are displayed (as icons) in the right part of the screen. The user can access any tool either by selecting it from the tree view, or by clicking on the appropriate icon. At each level there is an option to return to the Main Menu (by clicking on the 'Main Menu' icon).

The Exaquantum Management Console can be opened either on the Exaquantum server or on any installed client machine, provided that the user is authorised to use this tool.

Not all of the functions available from the Management Console are accessible to installed client machines; some can only be used on the server.

4.2 System Configuration Tools

4.2.1 OPC Gateway Configuration

The OPC Gateway Configuration tool is used to define the OPC gateways that Exaquantum/PIMS connects to. It is primarily used:

- During initial system installation and set-up
- Whenever OPC servers are added
- Whenever OPC server details change during the life span of the plant
- To enable/disable OPC Alarms and Events for the Exaquantum server.

The OPC Gateway Configuration screen allows OPC gateways to be added, deleted and viewed, and their details modified. The screen has two main components, the OPC Gateways list and OPC Gateways details.

4.2.2 Production Calendar

The Production Calendar is used to set the Production Calendar periods. There are two types of production calendar periods:

- System-defined production calendar periods (Hour, Day, Month)
- Custom production calendar periods (defined by the user).

The Production Calendar screen contains two main components, a Name/Dependency list and the Production Calendar Period details.

Production Calendar periods will always follow each other immediately. For example, a period cannot be defined as 8 am to 5 pm each day, i.e. 9 hours followed by a 15-hour gap. They will also be for an unvarying time period; it is not possible to define 3 shifts of 8, 10 and 6 hours in a day (00:00 to 08:00, 08:00 to 18:00, 18:00 to 24:00). The actual period will however vary if it contains a Daylight Saving clock change.

The Production Calendar allows only certain combinations of offset and dependency. Each aggregation period has a fixed set of options from which the offset can be selected. In the cases where dependency exists for an aggregation period, the offset cannot be modified.

The periods set in the Production Calendar are also used in the Tag Template. They are listed in the Tag Template under the heading ‘Aggregation Periods’.

4.3 Tag Configuration Tools

The process of tag configuration is based on templates. It is a means of classifying tags and function blocks in terms of their structure and properties. Defining such settings in a separate data structure reduces the necessity to repeatedly set the same values for large numbers of tags.

The first stage in configuring tags and function blocks within Exaquantum, therefore, is to identify the same structure and properties across prospective tags and/or function blocks. Tag Templates and Function Block Templates can then represent this information, respectively. When tags or function blocks are subsequently produced, they can be associated with the proper templates.

4.3.1 Tag Template

The purpose of the Tag Template is to define a structure and set of properties upon which tags can be based when they are created (both function block tags and flat tags). Tag Templates must be configured before Function Block Templates.

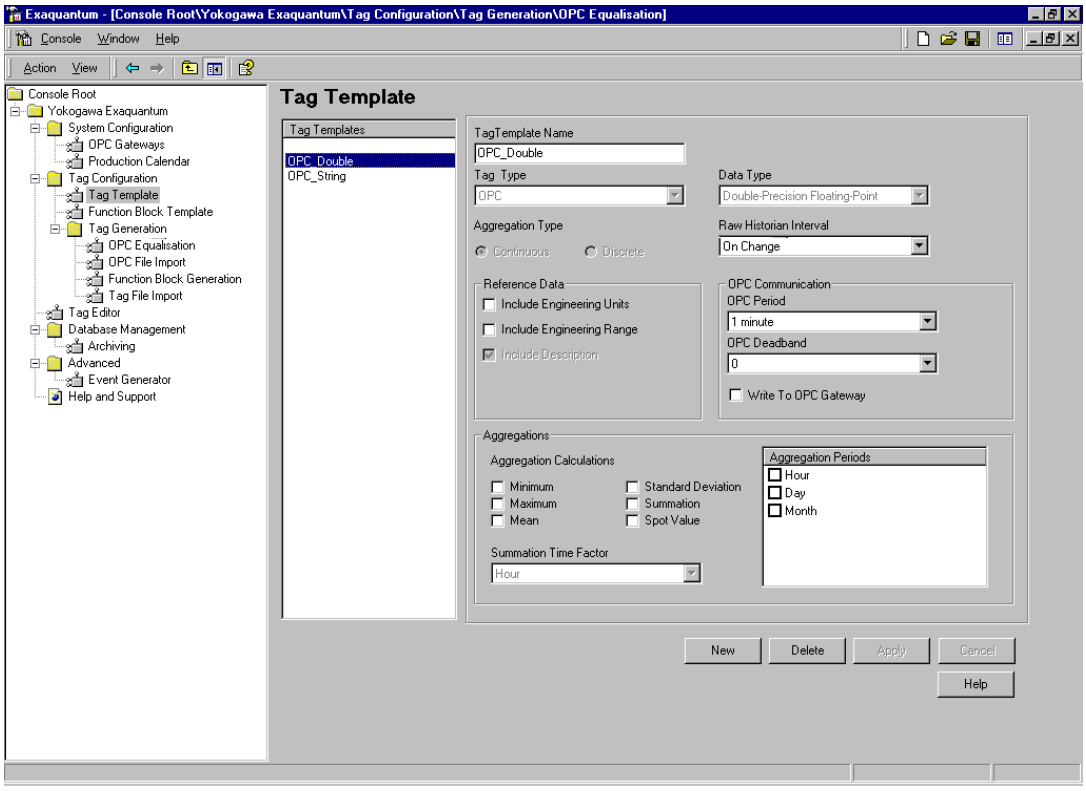


Figure: Tag Template Screen

The following items in the tag template are ‘locked’ once one or more tags are created and exist for a template:

- Tag Type
- Data Type
- Aggregation Type.

See also "When Changes Take Effect".

Changes to the Tag Template are controlled by versions. This means that there could be a number of templates with the same name, but slightly different contents. This is important because having versions allows changes to existing tags to be assessed when templates are to be updated. New tags are always created based on the latest version of the specified tag template.

4.3.2 Function Block Template

The Function Block Template is used to define which member tags are created when a particular function block is created. Manual and calculated tags also have their initial values defined by this template. For example, a manual tag has an initial 'value' and a calculated tag has an initial value defined by an 'Expression' that are used when a function block is created. Expressions can be used to update the expression of existing function block tags. This enables function block expressions to be updated easily.

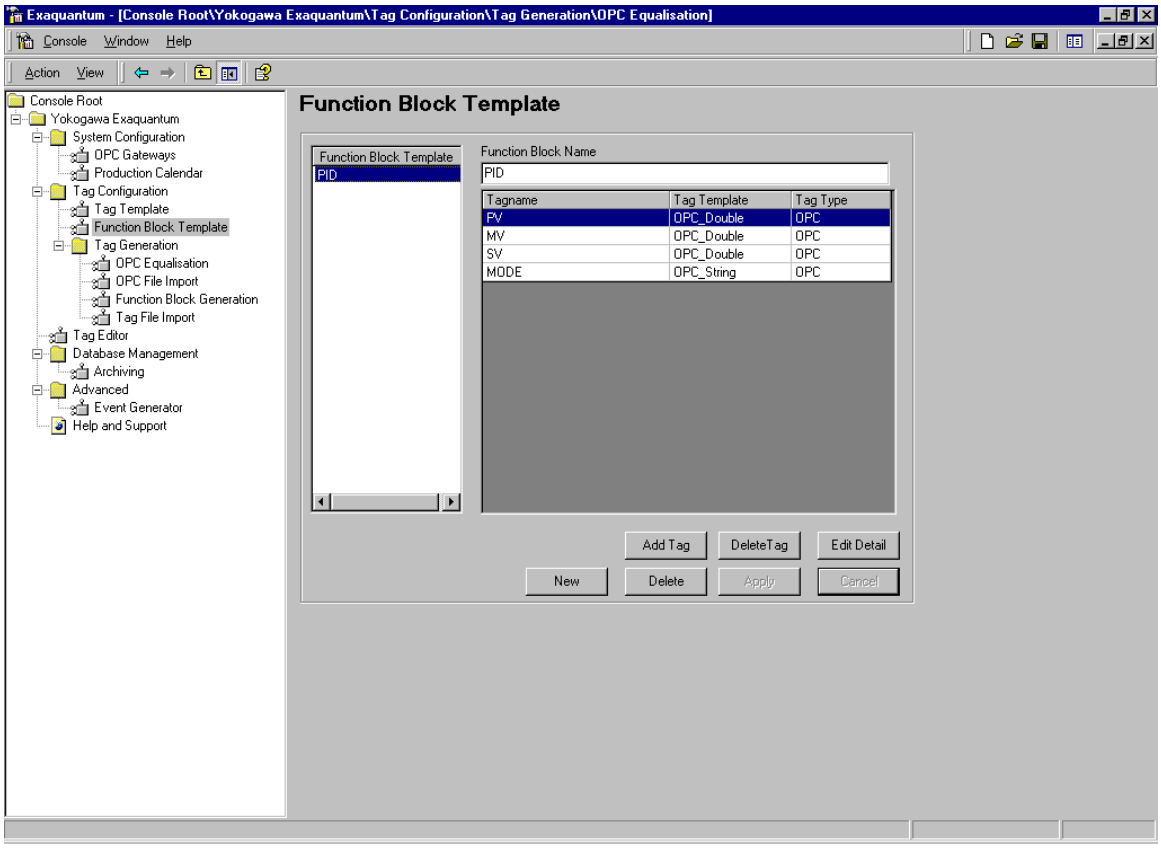


Figure: Function Block Template Screen

As with the tag template, the function block template is version-controlled. New function blocks are always created based on the latest version of the specified template.

Once one or more function blocks are created and exist based on a particular function block template, then any tag templates used within the function block template may only be modified to compatible tag templates. These templates are those with the same Tag Type, Data Type and Aggregation Type.

4.3.3 Tag Generation

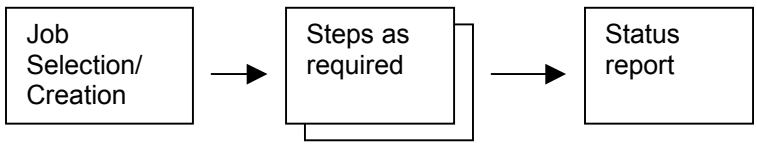
Tag Generation is a set of four tools that generate tags and/or function blocks. In each case the tool is a guided process with steps to be followed. The tools are:

- OPC Equalisation – get a PCS tag list through OPC server Browse interface
- OPC File Import – get a PCS tag list from a file
- Function Block Generation – specify function blocks directly by a file
- Tag File Import – specify tags in a file.

The tool used depends on the scenario, as follows:

- If the system has a pre-configured OPC server ready for Exaquantum access, the best choice is OPC Equalisation. It assumes a tag naming convention established in the PCS tags, so that a potentially huge number of tags can be easily filtered by tag names.
- If the PCS is not yet ready, or the naming convention does not meet the Exaquantum filtering requirements, OPC File Import is the best choice. It is almost identical to OPC Equalisation, except that user has to define a file to supply all the information (essentially a list of tag names) that the tool requires. In general, it is more flexible than OPC Equalisation. This is because there is no restriction of supported PCS systems, and it does not require a tag naming convention; the user, rather than the tool, filters the tags.
- To generate various structures of function blocks (i.e. how many tags, what type and with what name) based on various template definitions, the best choice is to use Function Block Generation tool. This requires the user to initially create the function block generation file that contains function block definitions, one for each function block.
- If the user wants to construct flat tags, then the Tag File Import tool is the best choice.

All tools start with a ‘Job’ screen that allows the user to select a job to run, or create a new one. The intermediate steps differ depending on the tool being used. At the end, all the tools display a status report screen that shows summary information.



A ‘Job’ is used to identify the task that has created the tags/function blocks. This information is useful when templates are changed. A typical scenario is as follows:

- Tags/Function blocks are created (by one of the four methods).
- When some changes are required (for example to add an aggregation period, to change the OPC update rate, or to add/remove/rename function block tags), a typical way is to find the job that created the subject tags/function blocks, and re-run it, after changing the template accordingly. Users can typically locate a job through the Tag Editor, as a particular tag/function block usually drives the required change; the Tag Editor shows the job when it is used to access such a tag/function block.
- When a job is rerun (or run), users are provided with an analysis report that shows the changes that are to take place. Users are then prompted to check the changes, and select an action: to continue, to change templates again, or even to cancel the operation.

4.3.4 Tag Editor

The Tag Editor is used to create, update and delete individual tags (both function block tags and flat tags). It allows:

- Creation of folders, function blocks, OPC tags, Manual tags and Calculated tags
- Editing of tags and function blocks by changing the associated templates (using the Template tools), and applying the templates through the Tag Editor
- Display of an analysis report whenever users attempt to make an update, deletion or creation of tags and function block
- Provision of information of a job that created the tag or function block
- OPC item ID to be specified and modified.

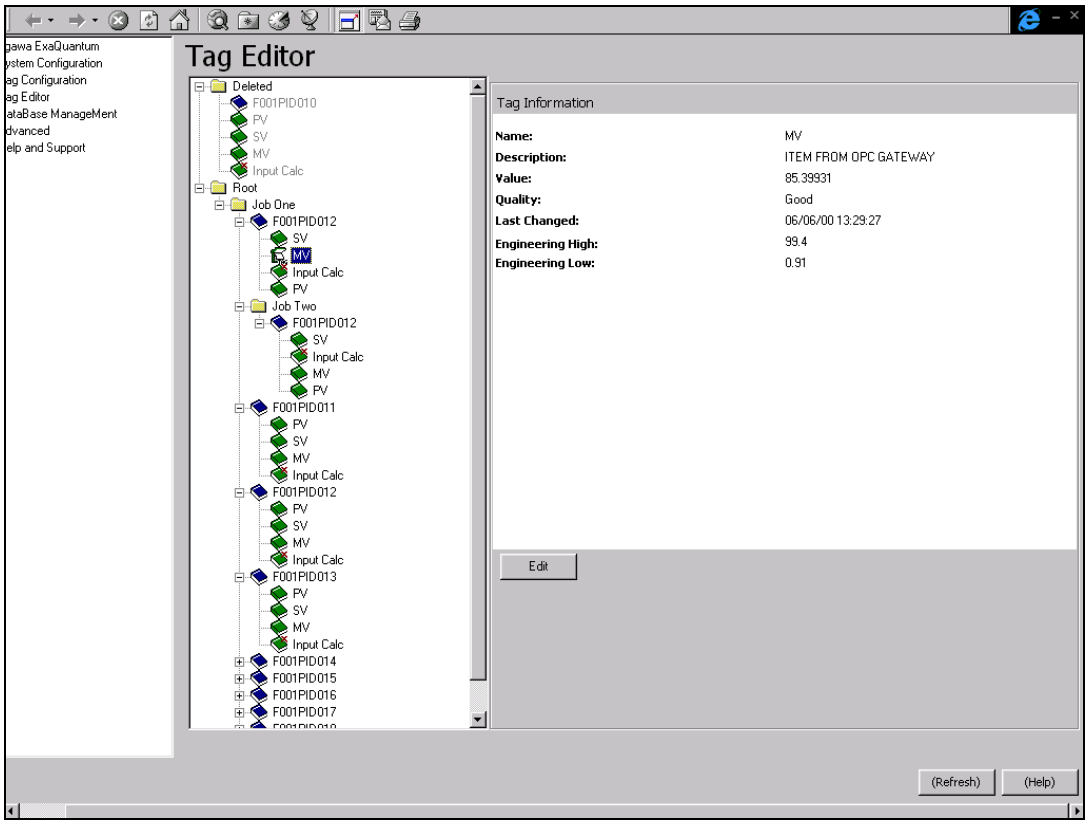


Figure: Tag Editor Screen

■ How and When to Change Tags/Function Blocks

Depending on the nature of the piece of information that defines a tag or function block, the source of the initial value and update value are different, as summarised in the following Table.

In the Table, 'Created from - Individual' means that values are given during Tag Generation work. 'Update individual' indicates that the Tag Editor or other tag management tools can change values in an interactive way.

Table: Function Block/Tag Source Details

Object	Information	Created From		Update individual
		Template	Individual	
Function Block	Function Block Template name	✓		✓
	Instance name		✓	✓
Function Block Tag	Tag name	✓		
	Tag template name(*)	✓		✓
	Reference data	✓		✓
	Value (Manual tag)	✓		✓
	Expression (Calculated tag) (*)	✓		✓
	OPC gateway (OPC tag)		✓	✓
Flat Tag	Tag name		✓	✓
	Tag template name		✓	✓
	Reference data		✓	✓
	Expression (Calculated tag)		✓	✓
	OPC gateway (OPC tag)		✓	✓
	OPC item ID (OPC tag)		✓	✓

Note: The OPC tag of a Function Block is either an instance or shortcut to an existing tag.

Note: Information marked (*) is subject to a function block lock. This is a mechanism that allows users to lock function blocks when their values (typically those mentioned here) become different from the values in the Template. Unless there is a lock, the changes will be overridden when the next generation is run. The user has to deliberately lock function blocks.

■ When Changes Take Effect

Changes can be made through various tools; the timing depends on the nature of the information to change.

● Tag/Function Block Oriented Information

Most of the changes made through the Function Block and Tag Templates take effect when the Tag Editor or Tag Generation tools are subsequently used to reconfigure function blocks and tags. However, when the Tag Editor changes certain values, the changes take effect immediately. These values are:

- Tag name or Function block name
- OPC gateway/item ID
- Reference data
- Value of Manual tags
- Expression of Calculated tags.

There are some Tag Template values that cannot be changed once tags are created. These values are:

- Tag type (OPC/Manual/Calculated)
- Data type
- Aggregation type (Continuous/Discrete).

● System-wide Information

How changes to system-wide information are implemented is detailed below.

Table: System-wide Information

Types of information	Information to change	Program or timing
Production calendar	Offset	Next re-start
	Add/Remove	When tag templates are changed and tag generation is re-run
OPC gateway	Host name	Next re-start
System wide parameter	%GOOD threshold	Next re-start

Tags created after changing these values will work with the new value immediately.

4.4 Archiving

Archiving removes a requested amount of data from the Historian Data database into an Archive database and allows the archive to be managed independently.

■ Archive Groups

In Exaquantum, items that are related by their type are grouped into a number of Archive Groups. Archiving is performed based upon a selected Archive Group, as is restoring previous Archives of that group. The Table lists the pre-defined Exaquantum Archive Groups.

Table: Archive Groups

Archive Group		Description
Type	**Sub-type**	
Raw	-	All OPC, calculated outputs and manual Items
Aggregation	Monthly	Calendar month aggregations
	Daily	Daily aggregations, Hourly aggregations and Custom aggregations
Reference	-	All reference items
Alarm & Events	-	Alarm and Event data

These groups replace Exaquantum R1.0 History Retention Groups. When R1.0 is upgraded, the History Retention Groups are used in parallel with the new Archive Groups.

■ Archive Devices

An Archive can be backed up to any physical device that is connected to the Server hosting the SQL Server database used by the Exaquantum Historian. This includes serial and random access devices such as tape, hard drives, CD or DVD. It is also possible to copy Archive Backups to network drives.

■ Maximum Size and Days

Within archiving, it is possible to specify the maximum archive size. This ensures that the archive fits the media that is used to back it up. This size restriction also allows the adequate amount of disk space required for creating and restoring an archive to be allocated. The size is specified in Megabytes (MB).

It is also possible to specify the number of whole day's worth of data that will be removed from the Historian database to create the archive. This is a requested figure; it will only be satisfied if the size of data for the number of required days is less than the value specified for the Maximum Size.

Backup Operation

Once an archive database has been created, it is possible to copy it to a backup device. Copying an archive usually takes place when it is created.

The backup operation can be performed to any backup device attached to the Exaquantum server, including directories on the server's disk, or to mapped network drives.

Multiple backups of the archive database are possible to different backup devices/media. The Backup operation does not change the state of the Archive, unless it is explicitly specified that the Archive be put offline after successful completion of the backup; it is still online and available for querying.

Where multiple Archives exist on the same backup media, its description and unique ID can identify a particular archive.

Online Operation

This operation allows previously archived (and eventually backed up) data to be re-integrated with the Historian, to allow access to the Archive's data. In order to restore an archive, there must be enough space available on the Exaquantum server. Depending upon the size and space allocated for restoring archives, many archives may be online at any one time.

Offline Operation

This operation removes an archive database from Exaquantum, both in terms of being available to the Historian for queries and physically removing it from the server's disk.

4.5 Event Handing

Exaquantum Event Handling provides the following functions:

- Execute application
- Application registration
- Deletion of registration information
- Internal events
- QEventUtility User Interface utility
- Trace function.

These are described below.

■ Execute Application

Event Handling uses Microsoft Message Queuing (MSMQ) to activate an application which then waits for MSMQ messages that it is interested in. The application will receive selected events as messages, with associated data, from MSMQ. The application can then execute according to its designed function.

■ Application Registration

Applications can be registered with Event Handling using the QEventHandler User Interface utility. Registration includes specifying the events that the application will use. Registration information can be saved to a database and used at each Exaquantum start-up.

The following conditions can be specified:

- Executing Application
  - Application path
  - MSMQ name
- Messages that are to be associated with this MSMQ and application:
  - Gateway
  - Event Category
  - Event Filter condition (up to 8)
  - Pass Data that will be passed to this application at execute (up to 8).

■ Deletion of Registration Information

Applications can also be unregistered using the QEventHandler user interface utility.

Internal Events

Event Handling can support Exaquantum internal events originating from Aggregations and Calculations, in addition to events from OPC A&E server(s):

Calculation Event

An event can be sent from a Calculation expression of a calculated tag, by using a prepared function call. At that time, it is possible to specify additional parameters to be sent to the application.

- Message: text string (maximum 256 characters)
- Tag ID: internal Tag ID of the calculated tag

  This data is prepared by the framework and does not appear in the function call used in the expression.
- Data 1 – 8: a maximum of 8 data items as text (maximum text size is 256 characters).

Aggregation Event

An aggregation sends an event to indicate that the aggregation was completed for all tags, with the following additional data:

- AggregationType : the Aggregation type – the ID of the aggregation period
- AggregationTime : the Aggregation time that was executed.

Trace Function

Event Handling also has a trace function that outputs a record of all events that it generates to a text file.

QEventUtility User Interface Utility

QEventUtility is one of the Exaquantum tools that can be accessed from the Management Console (from 'Advanced' in the tree). It provides the following functionality:

- Displays a summary of registered applications
- Registers/unregisters application information
- Displays current Event Filter condition information
- Adds/deletes/modifies Event Filter condition
- Loads Event Filter condition
- Generates Manual Events.

4.6 Help and Support

The Exaquantum Management Console provides a file-based URL link to a support page written in HTML. The support page contains hypertext links to the on-line help files shipped with Exaquantum and to the Exaquantum support web-site on the worldwide web.

4.7 System Administration Tools

The following administration tools are available to different categories of users.

4.7.1 Tools Intended for End Users

- **Start-up and Shutdown**

  Users can manually start up and shut down Exaquantum through a simple user interface program called ExaQCtrl.

  Exaquantum can be configured to start up automatically when the computer boots up. There is also a command line tool that provides a way to automatically shut down the system upon receipt of a UPS message.

- **Database Backup and Restore**

  The databases of the Historian and other configurations can be written to external media for later restoration. The task is carried out with the standard SQL Server backup and restore functions.

- **Windows Resource Backup and Restore**

  In order to restore Exaquantum-specific resources in Windows system, it is suggested that a backup be taken. The user can decide which tools to use.

4.7.2 Tools Intended for Project Engineers

- **Installation**

  The InstallShield program is used to install and uninstall Exaquantum components.

- **User / Group Configuration**

  The Exaquantum NT Users and Groups must be created before the Exaquantum system installation can begin. Guidance is given in the Installation Guide document.

- **Time Synchronisation**

  Exaquantum time should be synchronised with OPC server time. A special tool is provided to specify which OPC server Exaquantum time should be synchronised with, and how frequently the synchronisation should take place. The target OPC server does not need any dedicated software to do this.

- **Database Sizing Tool**

  This tool retrieves information that indicates how large a disk must be for it to be able to store the data planned for the Historian and other Exaquantum databases.

- **Database Wizard**

  The SQL Server Exaquantum databases can be recreated with this tool.

- **Performance Monitor**

  The Windows NT standard tool is used for the purpose. This is especially useful when the project is at an early stage, when there could be a number of performance and stability oriented issues.

- **Diagnostics**

  This tool provides two particular types of health indicator information:

  - Version check tool, which verifies the integrity of the installation
  - OPC diagnostics, which checks the health of the OPC communication.

4.7.3 Tools Intended for a Development Company

■ Event Log

The Windows NT Event Viewer displays the Exaquantum and SQL Server software execution process log files, and is intended primarily for Yokogawa engineers.

4.8 Support Tools

4.8.1 Configuration Reporting Tools

■ Cross-reference Tool

The Cross-reference tool allows users to produce reports of cross-referenced information that exist for a specified list of tags or templates in Exaquantum.

There are three types of Cross-reference checks that can be performed:

- References to specified tags in Exaquantum calculations
- All tags attached to a specified Tag Template
- All Function Blocks attached to a specified Function Block Template.

The Cross Reference tool has a flexible approach to reporting; the selection of tags or templates to search for allows selection of single tags or templates, wildcards, and folders (i.e. all tags within a folder), or a combination of all three.

■ Tag Configuration Viewer

The Tag Configuration Viewer assists configuration engineers in diagnosing problems and checking that the Exaquantum system is set up correctly.

The Tag Configuration Viewer provides:

- A viewer that displays Tag Configuration information
- A consistency checker that runs a check on the Exaquantum system and reports on various known configuration problems
- An export facility to store configuration information in CSV files.

4.8.2 Historian Management Tool

Exaquantum provides a tool to manage the Historian database. Typical cases that the tool will primarily address are when the disk is full and when time is accidentally changed to the future. These are described below.

- **When the disk gets full**

  If constant administration work is done, this situation should not arise. However, once it has happened, the best way to cope with the difficulty is to delete the old historian information, and make space for the Historian database until a permanent solution is implemented.

- **When time is accidentally changed to future**

  In a typical situation Exaquantum is time-synchronised with the OPC server. Whenever the OPC server time changes accidentally to the future (for instance one-year ahead), Exaquantum time will catch up eventually. One or more values with such future time will be stored in the Historian. When the time is corrected, all of the now ‘proper values’ received from the OPC server are regarded as Late-arriving data; these cause considerable problems to the Historian. In this case any wrong future values are simply removed from the Historian by this tool.

4.8.3 Exaquantum Versioning Tool

The Exaquantum Versioning tool checks that the versions of all Exaquantum-related files on the computer are correct. The files that are checked are those installed by Exaquantum, and common files that are installed as part of other software. The Versioning tool compares a list of all expected files, and their version numbers, with the actual files on a specific computer.

The Versioning tool also allows the user to create a log of ‘file version information’, based on what is currently installed on the computer. This can also be used to perform the version checks, rather than using the list supplied at installation. This is a useful option after installation, as it allows the user to check the ‘as installed’ software.

5. Exaquantum/Explorer

Exaquantum/Explorer is a set of analysis and reporting tools with a reporting environment, that together provide an easy-to-use method for building User Interface displays.

Exaquantum/Explorer has been designed using the standard Microsoft Multiple Document Interface (MDI) and ActiveX technology. This gives it a Microsoft look and feel so users who are familiar with Microsoft products will feel comfortable with most of its interactions and terminology.

The analysis and reporting features of Exaquantum/Explorer comprise a range of ActiveX controls. Each control specialises in a particular graphical representation of data from the Exaquantum Data Server. Meanwhile, these ActiveX controls are hosted by the Exaquantum/Explorer infrastructure that enables a number of rich features for making the best use of the ActiveX controls.

5.1 ActiveX Controls

The Exaquantum/Explorer functionality is implemented by ActiveX controls. Most of the important and useful ActiveX controls are supplied with Exaquantum/Explorer. This section describes some of commonly used ActiveX controls.

5.1.1 Trends

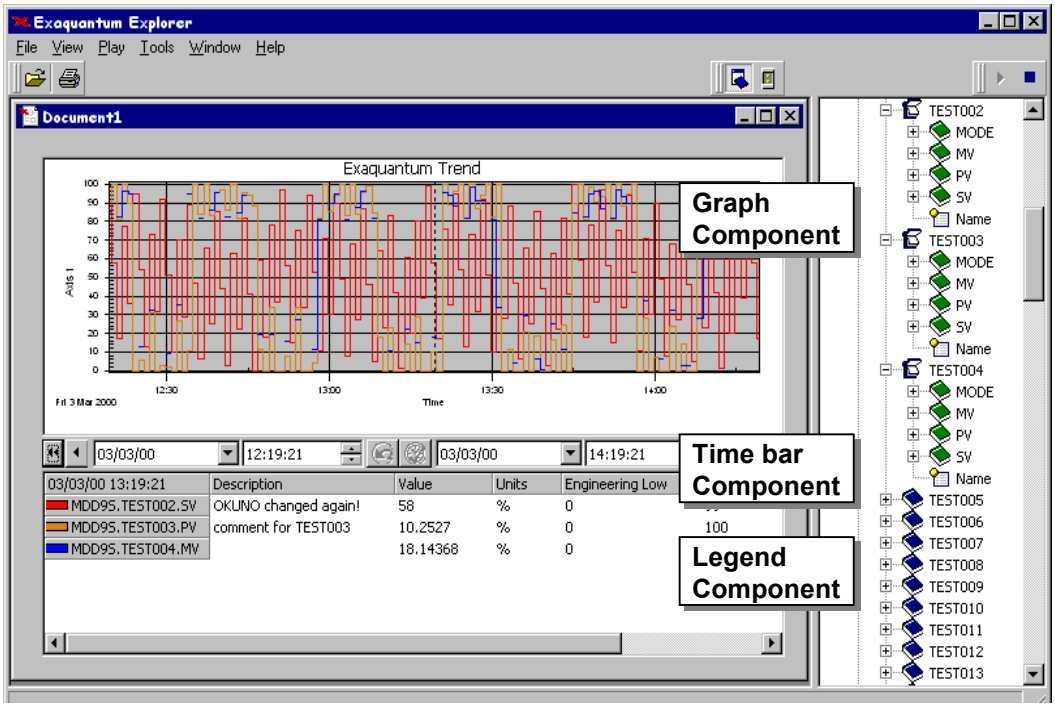


Figure: Trend Control Components

The greatest value of a PIMS system is in the long-term history, and the first tool for examining history is the Trend control. There are a variety of scenarios in which a user will examine trends, for example, when the user wants to:

- Instantly view a tag trend, without having a pre-configured application
- Select “personal” trends, for rapid and frequent access to particular views of history
- Find pre-configured trends in a structured User Interface, using the interface structure to find particular groups of tags for particular times
- View live updating or static history trends.

The Trend control supports all these scenarios.

The main features of the Trend control that support this range of uses are:

- Tags can be assigned to a Trend in the “drag and drop” fashion. Selection of the tags for assignment is made simple by the use of the Data Selector. The Data Selector presents all the tags in a structured tree. Typically, the structure will mimic the physical layout of the site.
- Tags can be assigned to the Trend control whilst the control is running.
- The Trend control and Data Selector can easily be built into applications, configured with tags, and saved as personal files. A Trend is then instanced by simply clicking on a file.

- The Trend control can be built into many pages of a single application and configured with tags relevant to the page. The navigation facilities of the application can then lead the user to the page and hence the trend of interest.
- The Trend control has a wide range of possible settings, including pen colours, line styles, live/historical, update rate. To ease the configuration process there are a range of typical templates for the most frequently used settings.

Once a user is viewing an instance of the Trend control configured with the tags of interest, he may want to fine-tune the plot range of the time and amplitude axis, to further investigate the tag history. Features to aid this are:

- Full control of the time-scale axis, start and end times, by direct setting
- Full control of the Y-axis, by direct setting. Up to six axes can be configured; they can appear either stacked or grouped (overlapped).
- Zoom facility. The user can use the mouse to select a rectangle on the Trend, and the rectangle will expand to cover the whole trend area, (the axis scaling changes automatically).
- Choice of point representation. Points on a trend can be connected by a line, or extended step wise, or be left unconnected.
- The Trend cursor can be positioned at any time. Details of tag values and the selected time will then appear in the legend area.

5.1.2 Data Entry Grid

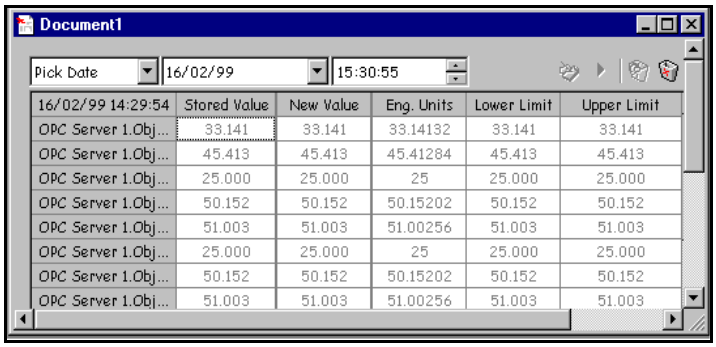


Figure: Typical Data Entry Grid

The Data Entry Grid is a control that allows the user to view information for a selection of tags; the information is displayed in a tabular form for a selected time.

Some information is best presented to the user in the form of a Data Grid, for example, several laboratory analysis results belonging to a single sample of product.

In addition to being able to display information about tags, the Data Entry Grid allows data values and data qualities for tags to be changed and written back to the database by the user, for a selected time.

Tags can be easily configured to the Data Entry Grid with the aid of the Data Selector.

5.1.3 Alarm and Events

Alarms and Events are key types of information stored by Exaquantum (where supported by the data source).

Alarms are abnormal conditions that are defined for objects within the PCS. Each condition may include sub-conditions that enable the user to accurately identify the cause of the alarm. For example, the LevelAlarm condition may include sub-conditions such as HighAlarm, HighHighAlarm, LowAlarm, and LowLowAlarm.

Events are any changes to the PCS settings that may have an impact on site processes. For example, operator actions or system configuration changes.

Exaquantum/Explorer provides the ability to display current, or historical, alarm and event information through an ActiveX control.

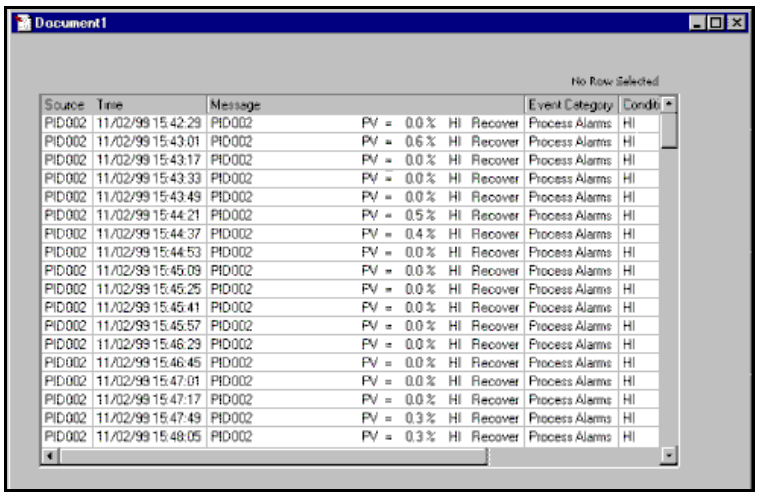


Figure: Typical Event Summary Control

By default, the Event Summary Control is displayed as a table of events in ascending chronological order. The table can be re-ordered by any column. Double-clicking on any event in the list will display a Detail window.

5.1.4 Web Browser

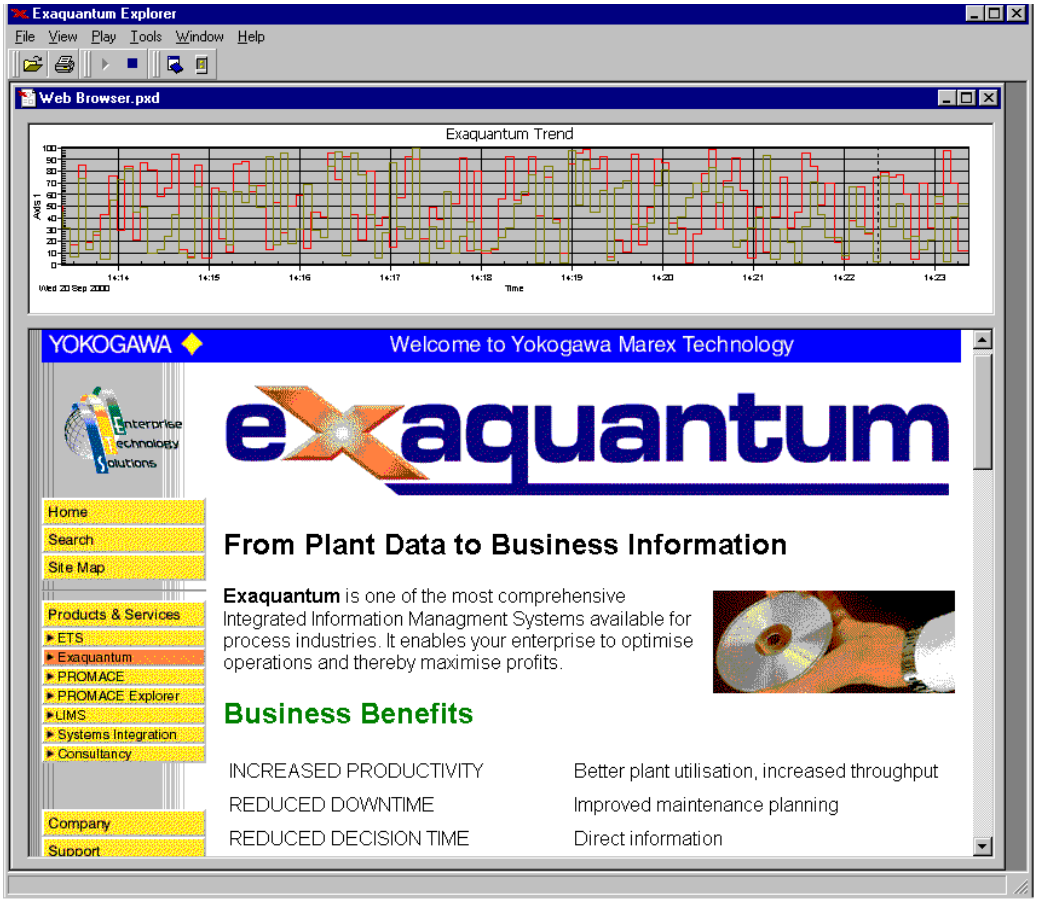


Figure: Example Web Browser Control

The Web Browser Control allows pages on a business Intranet, or Microsoft's Internet Explorer, to be displayed in a host application, alongside other ActiveX controls.

5.1.5 Excel Viewer

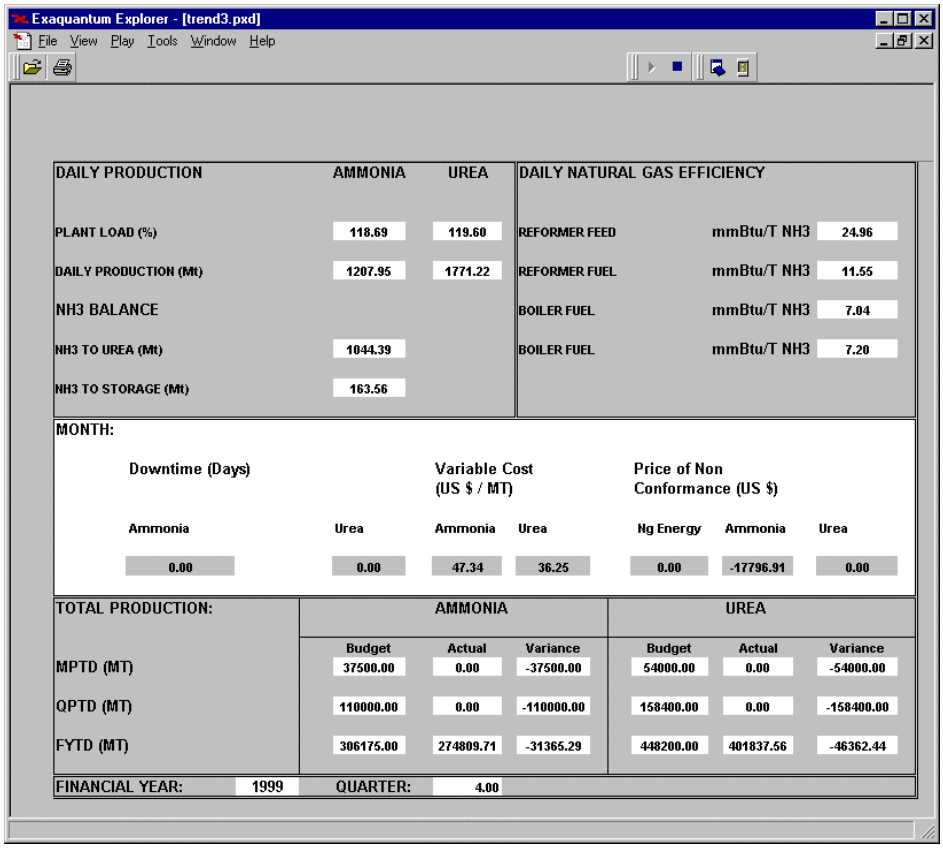


Figure: Typical Excel Viewer

Excel is an invaluable tool for process data analysis and reporting. Excel has easy access to Exaquantum data, data in other Excel files, and any data available through an ODBC link. The user may have a reporting requirement that integrates data from one or more of these sources, and performs summary calculations.

The Excel Viewer allows an Excel file to be instantiated within an application, and hence viewed from the application.

5.2 Exaquantum/Explorer Infrastructure

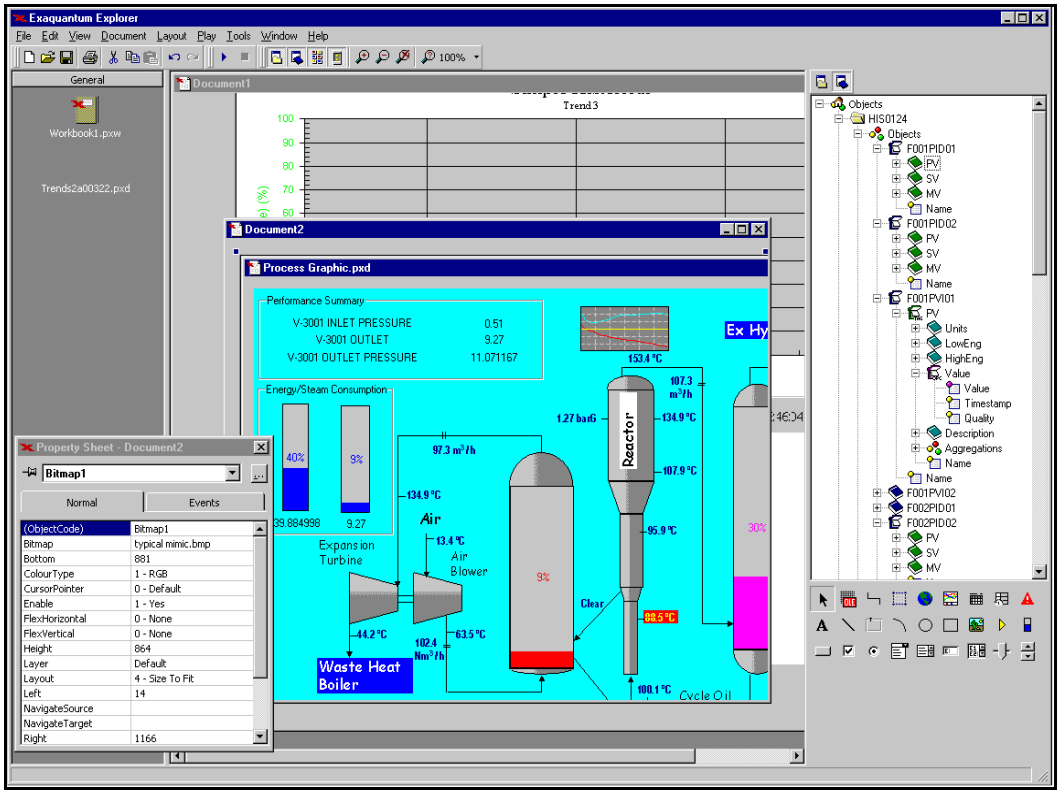


Figure: Exaquantum/Explorer – Design Mode

Exaquantum/Explorer provides the infrastructure to build and link ActiveX controls into cohesive, industry applicable applications and reports.

The Exaquantum/Explorer infrastructure comprises these features:

- Native Objects
- Host Document for Controls
- Multiple Document Interface
- General Reporting Environment
- Tag Configuration
- Report Times
- Data Parameterisation
- Scripting.

Each of these subjects is expanded in the following sections.

5.2.1 Native Objects

Native Objects are similar to ActiveX controls, but are available only in the Exaquantum/Explorer environment. Examples of Native Objects are: textbox, editbox, listbox. (In the text that follows the word “controls” will be used for both Native Objects and ActiveX controls.)

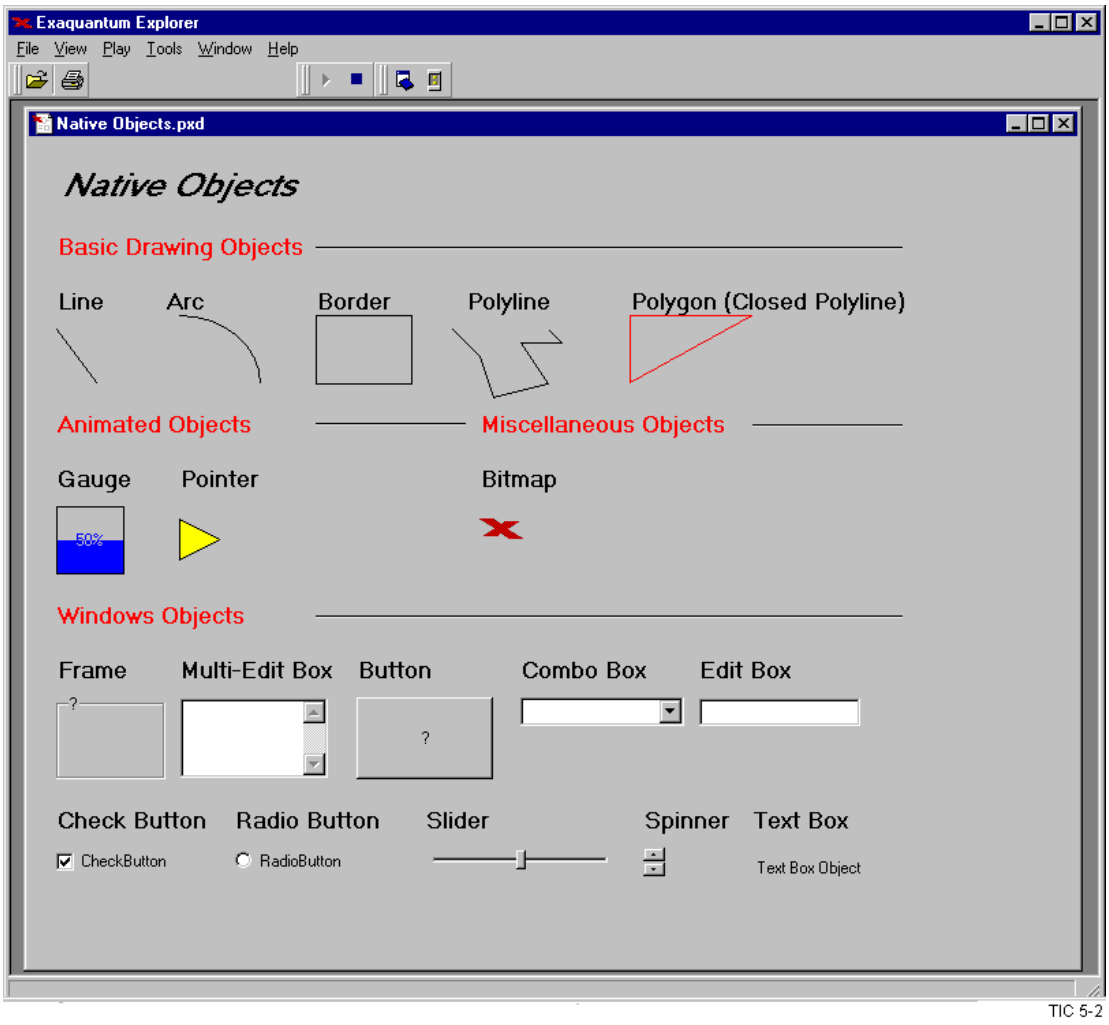


Figure: Example of a Document with Native Objects

Native Objects can be categorised as:

Drawing objects: e.g. line, circle, rectangle. These objects can be used for drawing the static part of the document.

Animated objects: e.g. gauge, pointer. These objects are used for dynamic representation of Exaquantum server data, i.e. tank fill.

Miscellaneous object: e.g. bitmap. This object is used to populate the document with a bitmap image.

Window objects: e.g. listbox, textbox. These objects are used to create an application user interface.

The most commonly used object is the textbox. A textbox can be configured to show a tag value by simply using “drag and drop” from the Data Selector.

5.2.2 Host Document for Controls

Controls are hosted by the Exaquantum/Explorer container. This container has the appearance of a document. (In the text that follows the word “Document” will mean a container configured with ActiveX controls.)

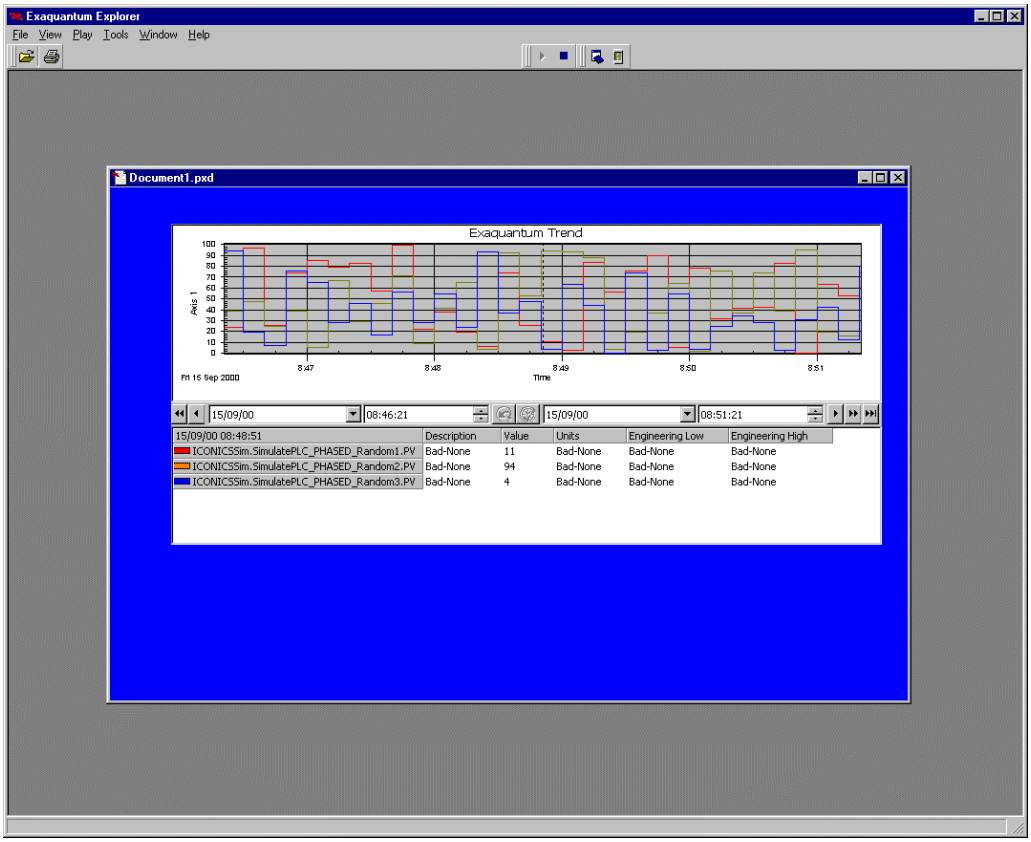


Figure: Example Document

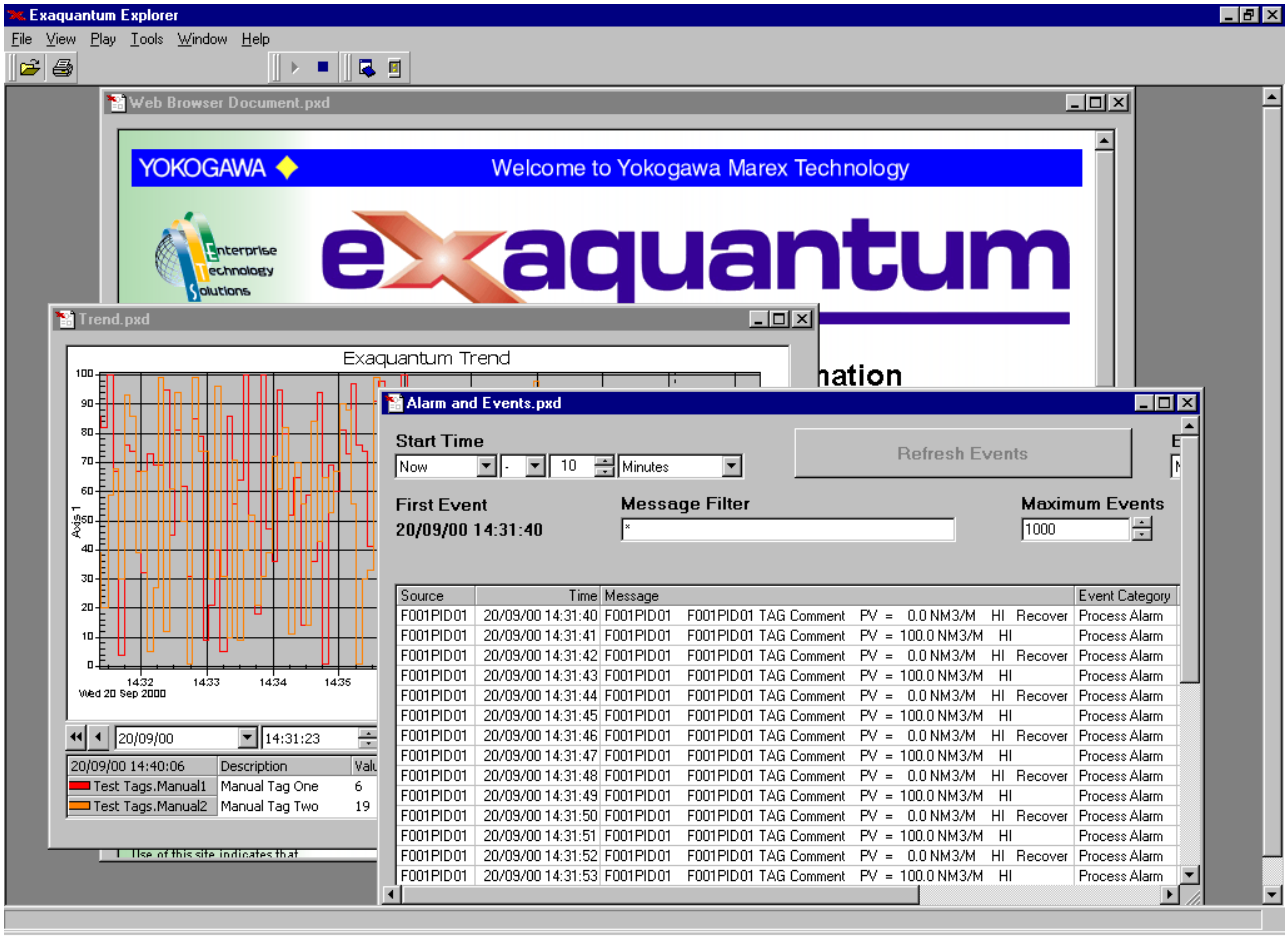
Using the Exaquantum/Explorer in Design mode, a document can be populated with controls, and the controls then configured with tags. Configuration is supported by drag and drop. Customisation of the controls is achieved using the Property Sheet/Property Binder. The document supports a powerful scripting language, for more information, see Scripting.

The normal range of edit actions is supported, i.e. resize, move, cut, copy and paste. Drawing operations are supported with a configurable grid and snap-to-grid behaviour.

Exaquantum/Explorer documents may also be saved in HTML format and viewed in a business Intranet through Microsoft’s Internet Explorer. Exaquantum/Explorer must be installed on the client computer to do this, and that client computer must be able to access the Exaquantum/Explorer documents.

5.2.3 Multiple Document Interface

Multiple documents can be active at the same time, and selected with the mouse. Alternatively, navigation between documents can be achieved with Button control navigation. This allows a hierarchical User Interface to be built.



TIC 5-3

Figure: Example Multiple Documents Interface

A document can be saved as a file. This file can be activated by double-clicking it.

Multiple documents can be activated simultaneously, with mouse selection or use of the Window task bar determining the foremost document.

In general, the number of documents of interest will be sufficiently large that selection by mouse or taskbar is impractical. In this case, navigation facilities can be added to documents, so that from one active document it is possible to “navigate to” or “select” another document. It is also possible to load the new document into a specified area on the current document.

By use of navigation, a large set of documents can be linked together in a structured hierarchy that matches the site structure. For example, the initial document may have a navigation button to Utilities. Utilities may have navigation to Water. Water may navigate to Tank3. The Tank3 document may have pressure, level, and flow tags for Tank3.

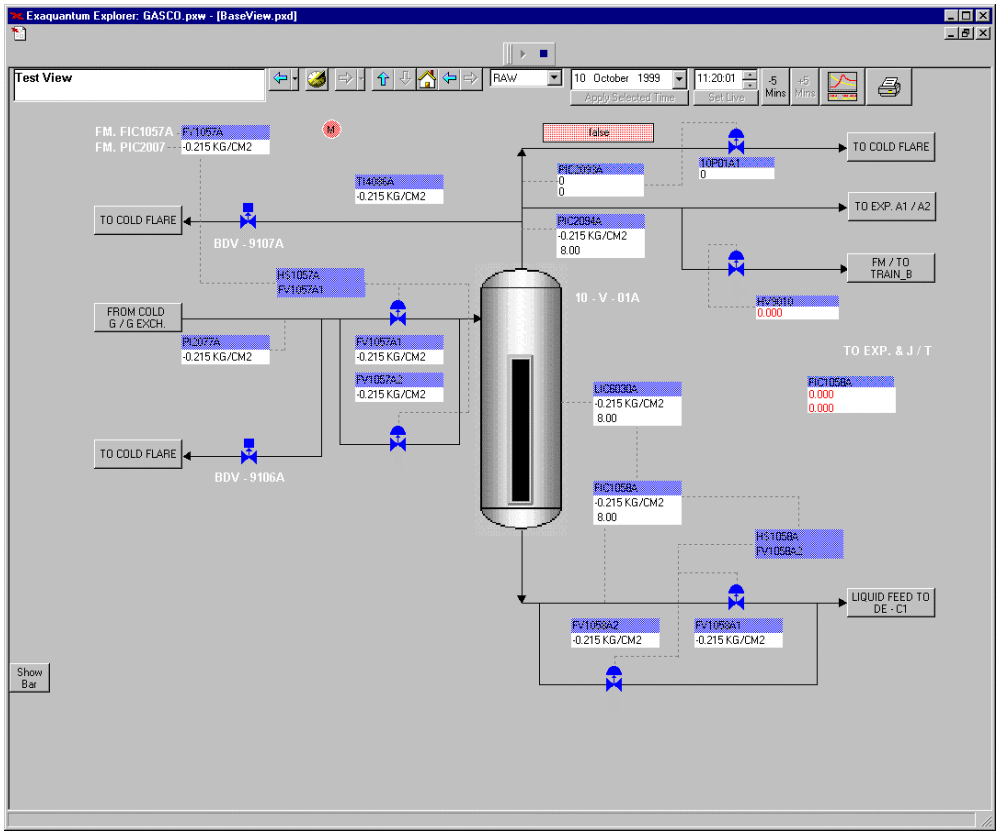
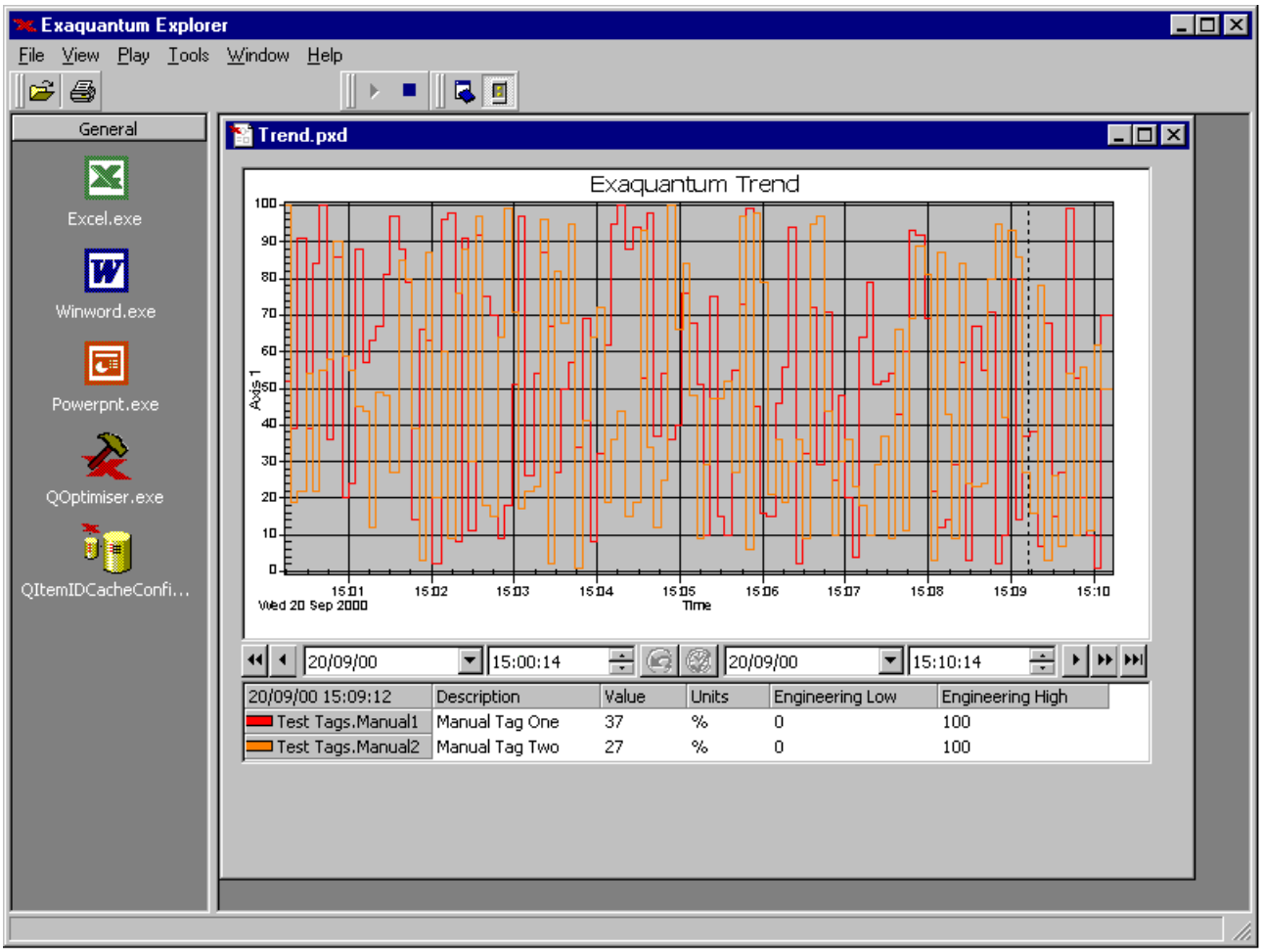


Figure: Example of Navigation Buttons on a Document

Exaquantum/Explorer further supports collecting documents into a single entity called a Workbook. The Workbook defines the Run mode environment. It sets the initial document, the Window style, and whether the Data Selector, Application Launcher and menus are available or disabled.

5.2.4 General Reporting Environment

Exaquantum/Explorer can be the single point of focus for additional computer functions, besides the implicit Exaquantum/Explorer functions. This is achieved through the Application Launcher. For example, from the Application Launcher it would be possible to launch Microsoft Word.



TIC 5-4

Figure: Example of Application Launcher

Some groups of users will expect Exaquantum/Explorer to be their single point-of-access for several applications. The applications may be general Office applications, such as Microsoft Word, and Excel. Alternatively, the applications may be specific site applications, such as a Reconciliation package or Operations Logbook.

The Application Launcher can be used to open any external application.

5.2.5 Tag Configuration

Tag configuration means the binding of Exaquantum server tags to controls. Exaquantum/Explorer supports “drag and drop” to maximise the ease and speed of tag configuration. Tags are “dragged” from a Data Selector, and “dropped” directly on the control.

■ Data Selector

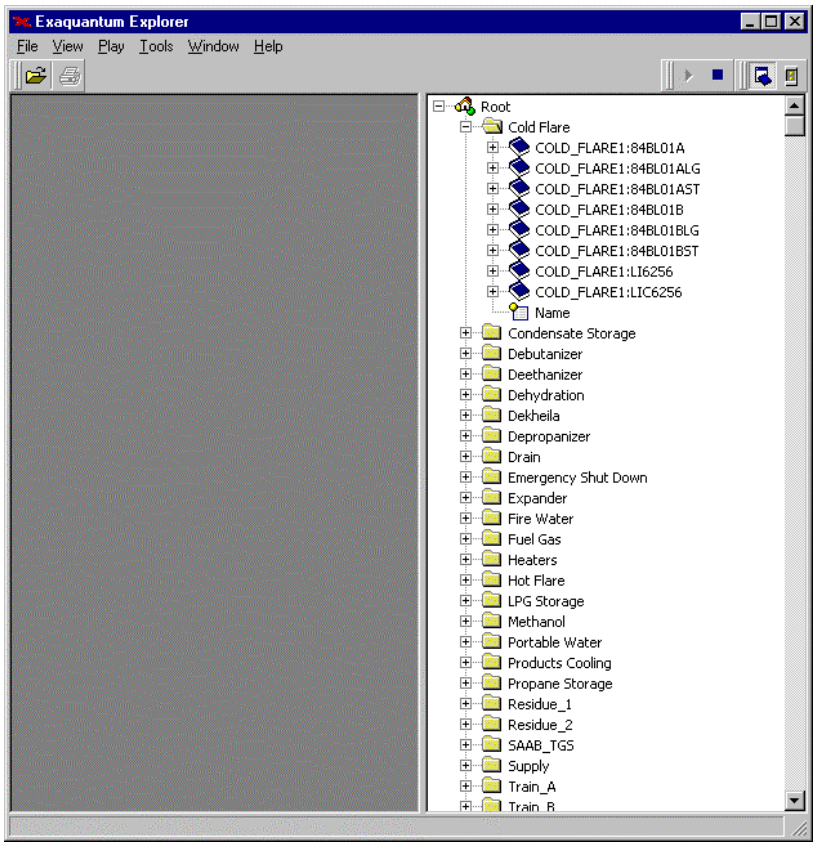


Figure: Example of a Data Selector

The Data Selector shows the hierarchical structure that represents the current Exaquantum process data. The structure of this display will represent the Plant Model. The Plant Model is a list of folders, function blocks, tags and items from the Exaquantum data server. The hierarchical structure can list, for example, all the plants, the unit in these plants, the equipment in each unit and the tags that measure the equipment performance.

The Data Selector is used to search for and select specific data. Data can be attached to the controls in a document by dragging the applicable item or tag from the Data Selector onto the control. This function binds the data to the Control. When the document is run, the data is displayed on the document.

5.2.6 Report Times

The data in a document will be associated with a time, or period. For example, a single primary value will be for a single time, an aggregation value will be for a Calendar Period, and a trend will have time period. Exaquantum/Explorer supports the association of all types of data with times and periods relevant to the Exaquantum server data.

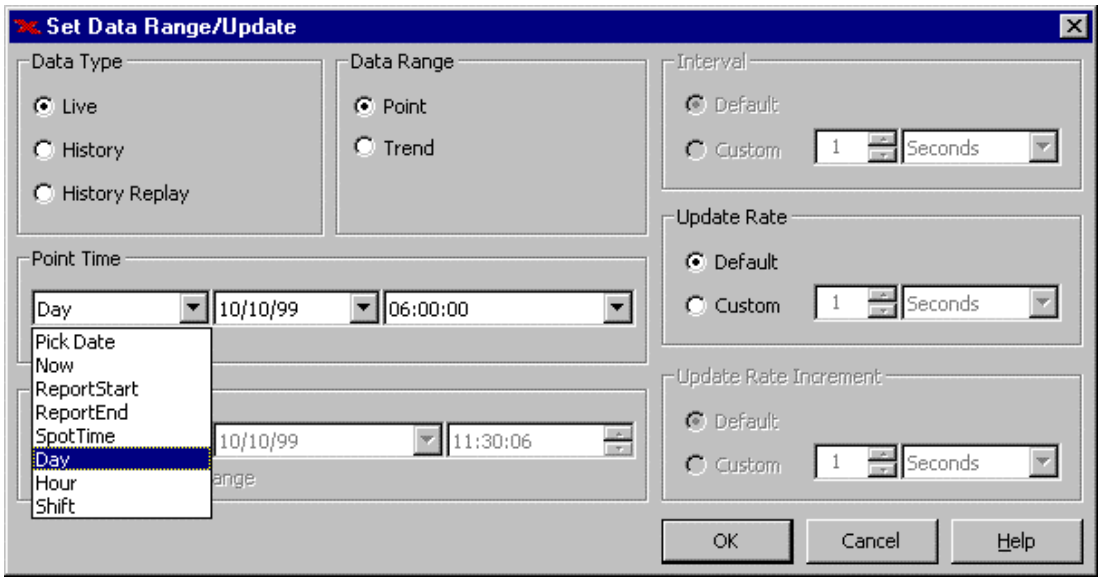


Figure: Set Data Range/Update Window

Users will expect to select a time of interest (perhaps when a plant shutdown has occurred) and navigate through many documents, with an unchanging time frame.

Exaquantum/Explorer supports three global time variables, SpotTime, ReportStart, and ReportEnd. Generally, Spot Time is used where data is for a single point in time, and ReportStart/ReportEnd are used where data is for a period. The values of these variables can be preserved through all documents.

Tag data can be bound to any of these variables when the tags are selected from the Data Selector for dragging to the document.

Other types of time binding are also possible, data can be bound to:

- NOW, meaning the data is Live
- A selected, unchanging, time
- A Calendar Period boundary.

Exaquantum/Explorer provides a comprehensive time selector. The Set Data Range/Update window allows the user to set any of the three global time variables.

5.2.7 Data Parameterisation

The purpose of data parameterisation is to introduce a level of indirect binding of control properties within documents to external data. A single document will be able to display different data, depending on the setting of parameterisation variables.

When a tag is bound to a control, it is the whole tag path that is associated with the control, for instance “Root.Boiler1.10PI001.PV.Value”. Data Parameterisation allows some part of the path to be dynamically assigned when the document is running. This is particularly useful for switching the view of the data from primary value to an aggregation value.

For example, assume a text box is bound to the tag:

```
"Root. Boiler1.10PI001.PV."+<AggregationMEAN>+"Value:Value"
```

When the AggregationMEAN mean variable has the value NULL, then the text box will display the value of tag 10PI001.PV.

When the AggregationMEAN mean variable has the value “Aggregations.Day.Mean”, then the text box will display the average Day value of tag 10PI001.PV.

The AggregationMEAN mean variable is a simple, locally held variable in the document, and can easily be changed with scripting.

5.2.8 Scripting

Scripting provides the ability to create programmatic interaction between components in a document, and between documents.

Scripting introduces programming flexibility into the whole of the Exaquantum/Explorer environment it can be divided into three areas:

- Control to control interaction
- Tag data to control interaction
- Document to document interaction.

■ Control to Control Interaction

A control has events and properties. One control can interact with a second control by use of scripting configured in an event, whereby the script in the first control manipulates the property of the second control.

A simple example is:

- A button and a textbox are configured on a document
- The button eventclick event is configured with *text1.text="hello world"*
- Therefore, when the document is run and the button is clicked, the text box will display *"hello world".*

This type of scripting is applicable to Exaquantum/Explorer as an Application. It offers the possibility to create forms style user interfaces.

Scripting is achieved with the Script Editor or the Script Wizard, these are describe later in this document.

■ Tag Data to Control Interaction

A tag can be bound to a control by simply dropping the tag onto the control. This is a quick way to configure the display of a tag value. There may be further requirements on the display of a tag value, beyond the basic value.

For example, there may be a requirement to indicate the tag quality value by the colour of the tag value. Good quality could be represented by a white Tag value, and Bad quality by a red Tag value. This gives immediate feedback to the user of the "worth" of the data being examined.

Another requirement may be to automatically format the number of decimal places displayed as a function of the magnitude of the Tag value. For example, if the Tag value is between 1 and 10, show 2 decimal places, if the tag value is between 10 and 100, show 1 decimal place.

These requirements can be achieved by binding the Tag to the control with scripting. This is a very powerful feature of Exaquantum/Explorer.

Document to Document interaction

Document interaction is controlled by the Manager object. A document can call on the methods/properties of the Manager.

It is possible to have two active documents, and for one document to interact with the other document.

Our previous example could be re-used in this context. A button on Document1 could have the eventclick event configured as:

```
Dim Doc As Object
Set Doc = Manager.Documents.Item("Document2.pxd")
Doc.text1.text=”Hello World”
Set Doc=Nothing
```

The result being that when the button on a document is clicked, “Hello World” appears in the text box on Document2.

This feature may be useful in a User Interface that is required to manage the whole document structure in some manner that is different to the standard Exaquantum/Explorer approach.

Script Editor

The Script Editor allows the editing of existing or the creation of new script files. It is also used to define functions and subroutines, which may be used in more than one script. This creates a library of shared routines for a particular document.

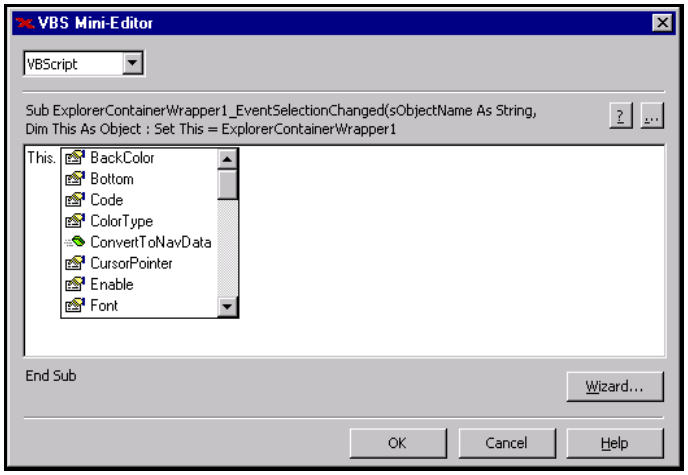


Figure: Script Editor Window

■ Script Wizard

The Script Wizard assists in identifying objects, properties and methods for inclusion in script. A list of the properties and methods for each currently configured Control can be obtained by using the drop-down lists at the top of the wizard. If a user needs to access a property or call a method, clicking on the **Insert** button will insert the required code into the edit window at the current location of the cursor.

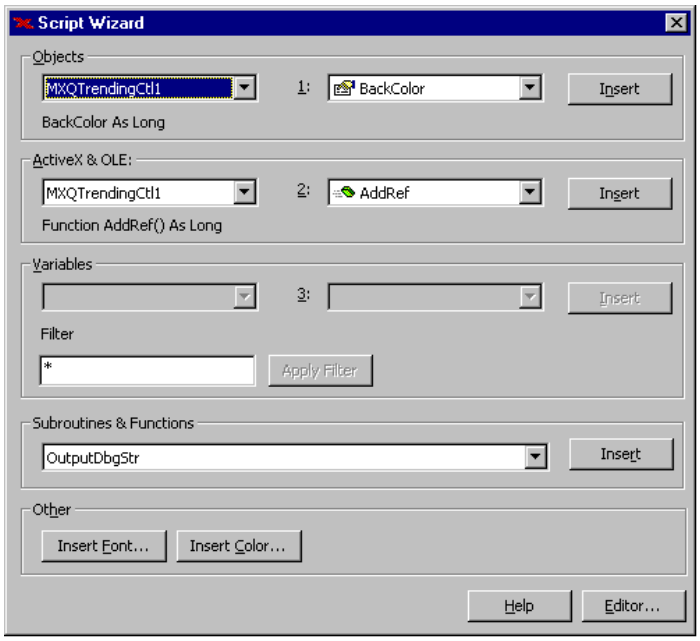


Figure: Script Wizard Window

5.3 Excel Add-In

The Exaquantum Excel Add-In extends the standard Microsoft Excel capabilities to provide control and display of real-time information. The Excel Add-In:

- Makes live and historical Exaquantum data available within Microsoft Excel
- Provides simple 'drag and drop' of Exaquantum data into spreadsheets
- Uses the standard Exaquantum Data Selector
- Integrates with the Exaquantum Production Calendar for setting report start and end times.

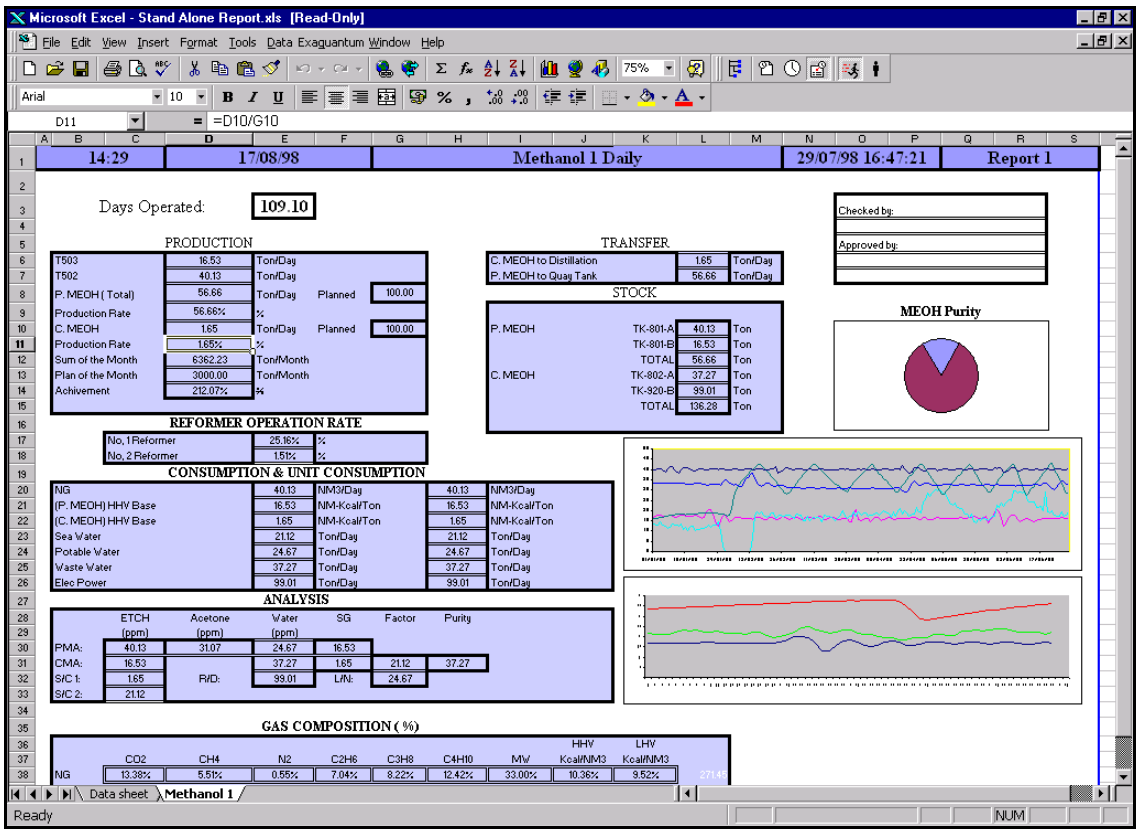


Figure: Example of an Excel Add-In Display

Blank Page

6. OLE DB Client

The Exaquantum OLE DB Client application allows the user to view a 'snapshot' of Exaquantum data using OLE DB/ODBC compliant reporting and analysis tools, for example Excel and Seagate Crystal Reports.

OLE DB allows generic access of information in various forms. Exaquantum takes advantage of the technology in such a way that it defines several useful forms of access methods that primarily expose the trend information.

In order for users to make best use of the technology in a comprehensive way, Exaquantum provides a simple wizard tool that allows users to retrieve information without any technical knowledge of OLE DB/ODBC.

6.1 OLE DB Client Tools

The OLE DB tools consist of:

- A Query Wizard for the production of QOLEDB queries
- An Item Selector for the selection of items.

■ Query Wizard

Users can construct query commands in an intuitive and comprehensive way, Exaquantum provides a wizard designed for Microsoft Excel. Using the Query Wizard, users are allowed to take advantage of OLE DB Client without knowing query parameters detail.

The Query Wizard prompts users for appropriate options. The first step is to select a query type, e.g., QTrendData, and a set of query specific selections will follow. In the case of QTrendData, these are:

- Select Times
- Select Options “IncludeEdge” or “IncludeBounding”
- Select Items
- Select Columns
- Finished.

■ Item Selector

The Item Selector is a standalone program accessible through the standard Window Programs menu. It provides the ability to select items to a list. Items can be dragged from the Data Selector to the selected list. The list of tags can then be copied to the clipboard as a comma-separated list. The list can then be pasted into the Crystal Reports entry box when prompted for the Items in a QData or QTrendData request.

6.2 Query Types

OLE DB Client has the following query types available:

- QBrowse, allows users to search for a list of Exaquantum tags (Name Space Browsing)
- QData, retrieves tag data for a determined time (Tag Spot Data)
- QAEData, retrieves Exaquantum Alarm and Event data for a determined time
- QTrendData, retrieves tag data for a specific time and updates the data periodically.

The two of most commonly used commands (QData and QTrendData) are explained further.

■ QData: Tag Spot Data

This query will return information for specified items at a particular point in time.

The command consists of the following parameters:

Columns — One or more unique column names such as Path name, Aggregation period, Value, Secondary Quality and Timestamp, or the '*' character that is shorthand for requesting all columns.

SpotTime — A locale formatted date/time string, the minimum time resolution is 1 second. Pre-defined values, such as NOW, are available, as well as calculations like NOW – 10 minutes.

Attributes — IncludeEdge or IncludeBounding constants.

- IncludeEdge — The point at the specified time is returned, or if a point does not exist then an interpolated value is returned from the last point in history.
- IncludeBounding — The point at the specified time is returned or if the point does not exist then the previous point in history is returned.

Items — A list of fully qualified comma separated paths to the requested items. All paths start from 'Root'.

● Example

The command,

```
QData< "*","24/05/00 11:36:00",
"IncludeEdge","Root.Server1.F001PID01.PV.Value ">
```

will return a spot value populated into Excel spreadsheet as follows:

Name	Aggregation	Value	Quality	SecondaryQuality	Timestamp
Root.Server1.F001PID01.PV.Value		12	Good	None	24/05/00 11:36:00

■ QA&EData: Alarm and Event Data

This query retrieves Exaquantum Alarm and Event data for a determined time.

● Query Parameters

The command consists of the following parameters:

Columns	One or more unique column names defined by the QAEData result set, or the '*' character. Multiple column names are comma separated. The '*' character is shorthand for requesting all columns.
StartTime	Inclusive start time of query for alarm and event data. A locale formatted date/time string; the minimum time resolution is 1 second.
EndTime	Inclusive end time of query for alarm and event data. A locale formatted date/time string; the minimum time resolution is 1 second.
MessageFilter	If a wildcard string is specified then alarms and/or events are only returned if the message matches the MessageFilter. The default is *.
MaximumEvents	Specifies the maximum number of alarm and/or events that can be returned, The default is to return all alarms and events.
OPCServers	A list of comma separated OPC server names for which alarm and event data is required. If no server is specified, data for all servers is returned.

- **Example**

  This example displays alarm and event data from Server 1 between the specified Start and End times, all columns will be displayed.

```
QAEData( "*" , "24/05/00 11:26:00" , "24/05/00 11:36:00" , ,,
"Server 1")
```

Server	Source	Timestamp	Message
Server 1	F002PID01	24/05/00 11:26:00	F002PID01 1234567890XXX1234567890 PV = 27.5 NM3/M HI Recover
Server 1	F002PID01	24/05/00 11:26:00	F002PID01 1234567890XXX1234567890 PV = 27.5 NM3/M LO
Server 1	F002PID01	24/05/00 11:26:01	F002PID01 1234567890XXX1234567890 PV = 47.7 NM3/M LO Recover
Server 1	F002PID01	24/05/00 11:26:02	F002PID01 1234567890XXX1234567890 PV = 30.7 NM3/M LO
Server 1	F002PID01	24/05/00 11:26:03	F002PID01 1234567890XXX1234567890 PV = 45.4 NM3/M LO Recover

Category	Severity	Condition	No of Attributes	CENTUM Message ID	Station Name	Tag Name	Recipe Group Number	Recipe Name
102	500	HI	10	4354	FCS0101	F002PID02	0	
102	500	LO	10	4353	FCS0101	F002PID02	0	
102	500	LO	10	4354	FCS0101	F002PID02	0	
102	500	LO	10	4353	FCS0101	F002PID02	0	
102	500	LO	10	4354	FCS0101	F002PID02	0	

Batch ID	Unit Recipe Number	Engineering Units	Data Value	Item Name
	0	NM3/M	27.5458030700684	PV
	0	NM3/M	27.5458030700684	PV
	0	NM3/M	47.6892623901367	PV
	0	NM3/M	30.7076587677002	PV
	0	NM3/M	45.412841796875	PV

■ QTrendData: Trend Data

This query will retrieve large amounts of trend data at a time. It allows multiple items and/or span of time within a single query.

● Query Parameters

The command consists of the following parameters:

Columns One or more unique column names such as Timestamp, Value and Secondary Quality, or the '*' character that is shorthand for requesting all columns.

StartTime Inclusive start time of query for trend data. A locale formatted date/time string; the minimum time resolution is 1 second.

EndTime Inclusive end time of query for trend data. A locale formatted date/time string; the minimum time resolution is 1 second.

Interval If Interval is specified as other than "00:00:00" then data is returned exactly once per Interval rather than for each data point in history.

Attributes IncludeEdge or IncludeBounding constants.

- IncludeEdge — The point at the specified time is returned, or if a point does not exist then an interpolated value is returned from the last point in history.
- IncludeBounding — The point at the specified time is returned or if the point does not exist then the previous point in history is returned.

Items A list of fully qualified comma separated paths to the requested items. All paths start from 'Root'.

● Examples

The first example query fetches the value and quality data of two Tags' values over a period of time with a re-sample interval of 1 hour. As a re-sample interval has been specified this dictates that the result set will have a shared Timestamp column for all items specified.

```
QTrendData< "*","24/05/00 07:36:00" , "24/05/00 11:36:00" ,
"01:00:00" ,  "IncludeEdge",
"Root.Server1.F001PID01.SV.Value,Root.Server1.F001PID01.MV.Value
">
```

Represents the string <Root.Server1.F001PID01>

Timestamp	#.SV.Value:Value	#.SV.Quality: Value	#.SV.Secondary Quality:Value	#.MV.Value: Value	#.MV.Quality: Value	#.MV.Secondary Quality:Value
24/05/00 07:36:00	175	Good	None	100	Good	None
24/05/00 08:36:00	525	Good	None	100	Good	None
24/05/00 09:36:00	378	Good	None	100	Good	None
24/05/00 10:36:00	231	Good	None	100	Good	None
24/05/00 11:36:00	588	Good	None	100	Good	None

The second example query fetches the Timestamp and Value columns for two Tag value attributes. The re-sample interval is not specified so the raw values are returned and hence a separate Timestamp column exists for each requested item.

```
QTrendData< “Timestamp,Value”, "24/05/00 11:26:00" , "24/05/00
11:36:00" , "00:00:00" , "IncludeEdge",
"Root.Server1.F001PID01.SV.Value,Root.Server1.F001PID01.MV.Value
">
```

SV.Timestamp	Root.Server1.F001PID01.SV.Value:Value	MV.Timestamp	Root.Server1.F001PID01.MV.Value:Value
24/05/00 11:26:00	364	24/05/00 11:26:00	100
24/05/00 11:26:20	287	24/05/00 11:36:00	100
24/05/00 11:27:20	210		
24/05/00 11:28:20	133		
24/05/00 11:29:20	553		
24/05/00 11:30:21	476		
24/05/00 11:31:21	399		
24/05/00 11:32:21	322		
24/05/00 11:33:21	245		
24/05/00 11:34:21	168		
24/05/00 11:35:21	588		
24/05/00 11:36:20	588		

6.3 OLE DB as a Programming Interface

Whenever an application program requires access to the Exaquantum data, the application program can use the OLE DB interface as an alternative to the DCOM based API.

The OLE DB interface is used in a similar way to SQL commands. Users familiar with SQL technology will understand the commands used in the OLE DB interface.

In summary, the significance of these two methods is as follows:

By using the OLE DB interface together with a commercially available application, e.g. FlexGrid, retrieving and populating information in a tabular form can be achieved with almost no programming effort. Writing back is also possible without writing any complex code. The application will not be limited to the tabular representation. There are, and will be, many OLE DB compliant toolkits in the market place, all of which are potential functional and representation programs of Exaquantum information through the Exaquantum OLE DB interface.

An application that requires asynchronous access to Exaquantum data should take advantage of the DCOM based API. With the OLE DB interface the application should access the data periodically by its own effort, while the DCOM API prompts the application for any changes of information.

7. Deployment Options

The Exaquantum system can be configured in a wide range of ways to meet the differing needs of the Process Industries. The configuration options are detailed in this chapter.

7.1 Supported OPC Servers

Exaquantum supports the following PCSs via associated OPC servers.

Table: Supported OPC Servers

PCS	OPC Server	Description
Yokogawa CENTUM CS/CS 3000/CS 1000	Yokogawa Exaopc	CS cassette Exaopc version R1.01, R1.10, R1.20
Yokogawa CENTUM-XL	Yokogawa Exaopc	XL cassette Exaopc version R1.20
Yokogawa µXL	Yokogawa Exaopc	µXL cassette Exaopc version R1.20
Honeywell TDC3000	Matrikon	Version 3.0.0
Foxboro IA series	Matrikon	Version 2.0.0
Modbus	Matrikon	Version 3.1.0

7.2 Platform Availability

Exaquantum software components are available for following platform environments.

Table: Platform Availability

Component	Platform	Description
Exaquantum/PIMS	Windows NT 4.0 Server with SP6A	Server (RTDB and Historian) and Configuration Tools
Exaquantum/Explorer	Windows NT 4.0 Server with SP6A	Client, includes OLE DB interface
"	Windows NT 4.0 Workstation with SP6A	
"	Windows 95 SR2	
"	Windows 98 Second Edition	

Exaquantum is available in both English and Japanese. Language support includes:

- Separate CD-ROM for different language version
- Default database settings in the language string
- Online manual in the language version
- Help in the language version.

7.3 Exaquantum Deployment and Network Configuration

For network configuration purposes, Exaquantum consists of three components:

- Exaquantum/PIMS server (comprising the Real-time Database (RTDB) and Historian)
- Configuration tools (comprising the UI tools set)
- Exaquantum/Explorer (including OLE DB based clients).

These components can be deployed in various combinations; the physical configuration differs depending on network requirements.

7.3.1 Deployment Examples

In a typical system, a server PC has the server and configuration tools components installed, whilst as many client PCs as are required can install the Exaquantum/Explorer component.

A variation of this is the Mid-scale system, where some (or even all) client PCs install not only the Exaquantum/Explorer component, but also the configuration tools.

In a small system, where there are few PCs, a server PC may have Exaquantum/Explorer installed as well as the other two components.

Configuration	Exaquantum/PIMS server	Configuration tools	Exaquantum/Explorer
Typical system			
Mid-scale system			
Small-scale system			

Installed on the server PC

Installed on one or more PC(s)

The Configuration tools must always be installed on the Exaquantum/PIMS server PC, because some of the functions (such as tag generation and archiving/restoring) are only available on the server.

7.3.2 Co-existing with Other Software Packages

In a typical system, a server PC has the server and configuration tools components installed, whilst as many client PCs as are required can install the Exaquantum/Explorer component. The exception to this policy is the Exaopc OPC server.

Exaquantum/Explorer is also resource-consuming software, especially when it runs the Trends component. Because of Exaquantum/Explorer's 'on-demand' nature, other software packages may run on the same PC. However, users should be reminded that Yokogawa does not guarantee the outcome of any concurrent usage of other packages with Exaquantum/Explorer.

7.3.3 Configurations

One of the disadvantages of Windows systems is the complex work involved in network configuration. This requires knowledge of Windows Domain, Windows Workgroup, and to some extent security mechanisms.

Exaquantum/Explorer client software will run on Windows 95/98 operating system. In this case, due to limited implementation of Windows 95/98 DCOM software, such clients must be members of the Exaquantum server domain.

When a client connects a server via PPP protocol (which is the case where a modem or ISDN is the path between them), it is recommended that some investigation be done to assess the feasibility of the connectivity, in terms of performance and reliability.

When clients need to reside beyond the firewall, a special configuration of the firewall software is required. This is a consequence of employing DCOM communication protocol between the client and server. DCOM default setting uses a wide range of 'ports', whereas the firewall allows only a limited number of open ports.

Examples of three configurations are illustrated below. For further information about actual network configurations, see the Exaquantum Document Set.

Typical Configuration

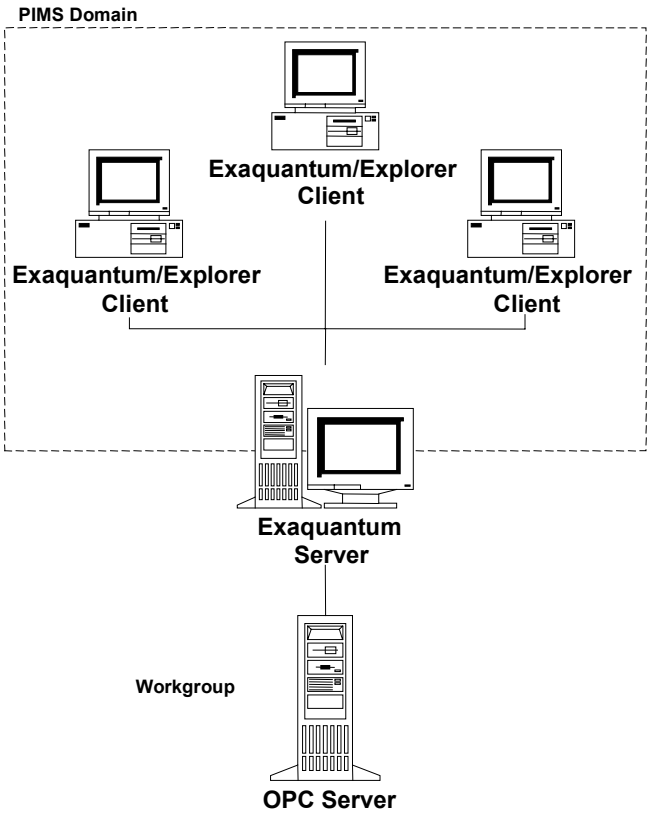


Figure: Typical Configuration

The Control system and PIMS domain (in many cases part of Office systems) are physically separated by the Exaquantum server, which has two network cards and acts as a gateway.

The Exaquantum server and clients are part of the PIMS domain, so the administration work is reasonably easy.

The OPC server and other control system components are configured as a Workgroup, as administration work is less intensive in this category.

Exaquantum in an Existing Mid-scale to Large Network

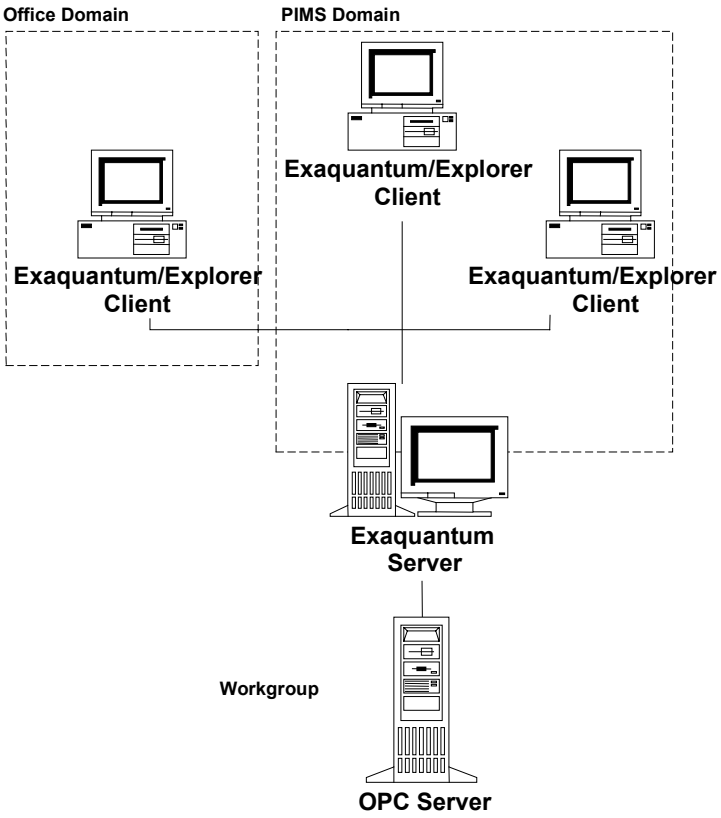


Figure: Mid-scale to Large Network

The PIMS system runs in a dedicated domain so administration work is kept separate from Office domain work.

Exaquantum/Explorer resides in both the Office domain and PIMS domain.

- **Newly Installed Small-scale System**

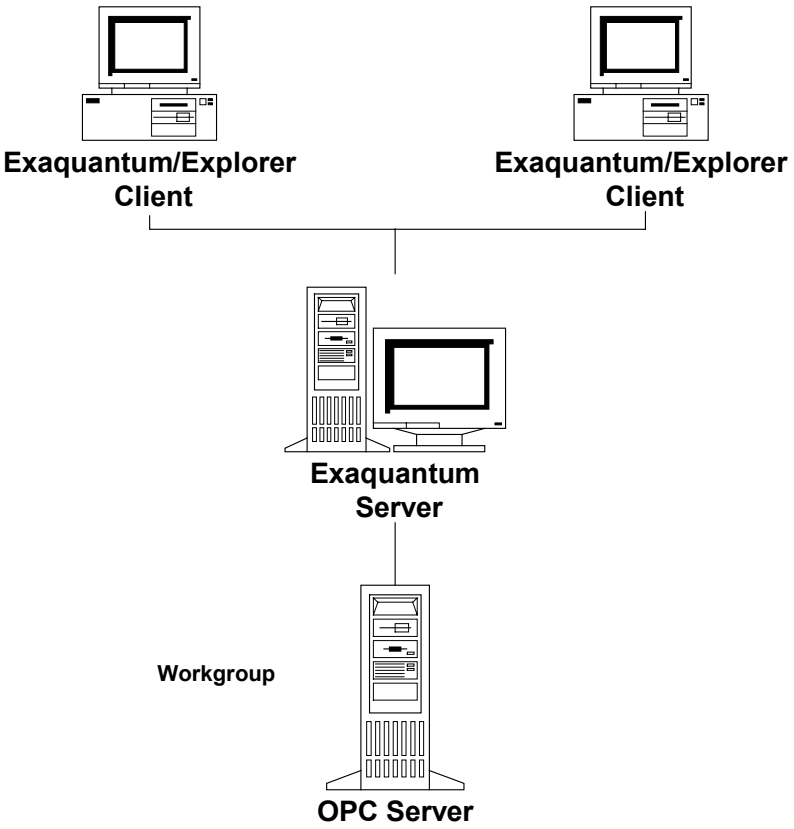


Figure: Newly Installed Small-scale System

In a small-scale system, the work of administrating the network is not so extensive. All the components are therefore put into the Workgroup configuration, rather than Domain.

The Domain configuration is useful when several PCs are involved, and network administration is not a negligible task. In small-scale systems, however, the cost-effective way is to put all the PCs into the Workgroup configuration. This is because the Domain configuration requires Primary Domain Controllers (PDCs) and Backup Domain Controllers (BDCs), which all cost more at the initial stage.

Blank Page

Exaquantum Glossary

TI 36J04A10-10E

This section contains a list of terms used in the Exaquantum document set. Each term is listed below in alphabetical order. Text shown in *Italic* form represents other terms defined in the Glossary.

A

- **ActiveX**
  Microsoft 'Container' technology for embedding software items from one application into another.

- **ActiveX Control**
  An ActiveX control is a standalone software component that performs a particular task in a predictable way. It can be used in many different applications, and the user will always know how to use it and how it is going to behave. ActiveX controls were formerly called *OLE* controls.

- **Add-in**
  Supplemental programs that extend the capabilities of any program, e.g. Exaquantum/Explorer, by adding custom commands and specialised features. Where Exaquantum/Explorer is installed, the Exaquantum Excel Add-in is available that allows Excel to access Exaquantum data.

- **Administration Tools**
  Also known as *Configuration Tools*.

- **A&E**
  Alarms and Events. Used to call event messages generated by *OPC servers*.

- **Aggregation**
  Calculation of summary values over a period of time. Built-in calculation for closing data.

- **Aggregation Period**
  This is a period of time over which aggregations can be made, e.g. hour or month.

- **Application Launcher**
  The Application Launcher is used to enhance the user interface. It can be used in a similar way to the Microsoft Outlook Task Bar, and can contain shortcuts to any application, in the same way as a User Desktop. This means that business or personal applications can be launched without having to minimise the user interface. As the Application Launcher is available in Run mode, it can be used as a navigation tool as well as a replacement for user desktop. In effect the Explorer user interface can be used as the user's default desktop (or as a Digital Dashboard), with live process data in the documents and access to all other business applications via the Application Launcher. This makes the Application Launcher a very powerful part of the user interface.

- **API**
  Application Programming Interface that allows programmatic access by any application program to Exaquantum information.

- **Archiving (of historised data)**
  The process of moving historised data into separate database than Historian database so that the portion can be managed independently; i.e. backing up onto an external medium, putting into offline state or back into online state.

B

- **BDC**
  Backup Domain Controller. Windows network component to back up the *PDC* whenever the *PDC* fails.

- **BLOB**
  Binary Large Object. A user-defined binary data structure which is set and interpreted by the application program; it is stored in the RTDB table as a single item.

C

- **Closing**
  To finalise and store aggregations at the end of an aggregation period.

- **COM**
  Component Object Model. A standardised way of linking software components, from possibly different vendors.

- **Configuration Tools**
  The tools that comprise Exaquantum.

- **Console**
  Also referred to as the Management Console (based on the Microsoft Management Console (MMC)). A Console is a user interface that provides an environment for running system management and administration applications. (Part of the Management and Configuration Tools.)

- **Control System**
  PC or computer based system that controls manufacture plant operations. They are real-time, required high reliability and availability. From Exaquantum viewpoint, they are primary data source, and additionally data sink when data write function is concerned.

■ D

- **Data Selector**
  A tree-view of all available data in the Exaquantum system used for selecting data items by drag-and-drop in Explorer and Excel Add-In.

- **DCOM**
  Distributed Component Object Model. Software components that reside on physically distributed computers. See *Control System*.

- **DCS**
  Distributed Control System. A *Process Control System* (*PCS*) that employs multiple computer units at different locations in the plant.

- **Design Mode**
  Documents can be created in Design Mode using data received from the *PCS,* etc. Facilities such as layout editing and scripting are available in this mode.

- **DST**
  Daylight Saving Time.

- **Document**
  A document is a user-interface screen developed in Exaquantum/Explorer.

- **Domain Configuration**
  A Windows Domain network configuration, which requires some Exaquantum configuration. Domain is used to simplify the management work of user accounts in mid-size to large network systems.

- **Dynamic**
  Continuously changing or updated (i.e. ‘live’).

■ E

- **Embed/Nest**
  Insert an object created in another application that supports *OLE* (such as Microsoft Word documents).

- **Embedding**
  A form of copying in which the copied object resides in the destination file only, with no link to the source file, but can be edited using the same tools available in the source file. Changes made to an embedded object exist only within the destination file and do not change the source file from which the object was copied. Likewise, changes made to the source file are not reflected in the embedded object.

- **End User**
  The customer for, or user of, the completed, configured Exaquantum system.

- **Equalisation**
  It is used to generate tags/function blocks data structure by way of OPC browse interface by which a candidate tag list is retrieved from the *OPC server*.

- **Exaopc**
  Yokogawa's OPC server product.

- **Event**
  An asynchronous generation of a message caused by the change of state of some resource or process.

- **Event Message**
  A message generated typically by the *PCS*, to inform users of an abnormal condition or of a significant process occurrence.

■ F

- **Folder**
  A folder is used to hold or contain *tags*, *function blocks*, or other folders. It provides a hierarchical naming service to locate *item data*.

- **Flat Tag**
  A tag structure that has no relationship with other tags, and is located below a folder.

- **Function Block**
  A function block is the generic term used to refer to the two-level structures used in *Exaquantum* to provide a meaningful function as a whole.

  In some contexts, it may refer to a Yokogawa CS data structure that is meant to provide a control and monitoring function. This is also two-level structure.

- **Function Block Template**
  A set of information to describe the structure and properties of function block. It consists of name, member tags and their definition and tags default values.

■ G

- **GUI**
  Graphical User Interface; user interface software using graphical techniques. (Now commonly known as the *User Interface* (UI).)

H

- **Historian**
  The Exaquantum component responsible for long term data storage.

- **History Stream**
  The total history of a given history type which is available for *item data*; the stream is represented by RTDB column.

- **Historised Values**
  Data that has been stored in the *Historian*.

- **HMI**
  Human Machine Interface.

- **Hot Spot**
  An area that can be activated by a mouse click.

- **Hyperlink**
  A *hotspot* that allows the user to jump to another location. The location can include another file on the hard disk or company's network, an Internet address, or a location such as a bookmark or slide. The *hotspot* includes display text (which is often blue and underlined) that the user clicks to jump to the specified location.

I

- **IIS**
  Integrated Information System.

- **Instance**
  Used to address a piece of information in contrast to class. It has a name linked to a physically existing object, and all of values that compose the object as a whole.

- **Intranet**
  Internet technology used on a Local Area Network that provides similar services within an organisation to those that are provided by the Internet outside the organisation.

- **Item**
  The lowest data level in the Exaquantum system. All items have a *value*, *timestamp* and *quality* code.

- **Item Data**
  All data stored for an *item*.

J

K

- **Key Performance Indicator (KPI)**
  Calculated values related to the performance of the plant or equipment.

L

- **LIMS**
  Laboratory Information Management System. An external system that manages all aspects of process laboratory work.

M

- **Multiple Document Interface (MDI) Functionality**
  A Windows application that allows many documents to be open at the same time. Exaquantum/Explorer inherently provides an MDI.

- **Mimic**
  A screen display that indicates the layout of a plant or process.

- **MMC**
  Microsoft Management Console.

- **MV**
  Manipulated Variable.

N

- **NLS**
  Native Language Support.

■ O

- **Offline (of history data)**
  History data that has been removed from Historian database, and hence not accessible by the programs.

- **OLE**
  Object Linking and Embedding. See *ActiveX controls*.

- **OLE DB**
  Microsoft-defined interface protocol and standard, which allows access to various types of source data in the form of structural two-dimensional array.

- **Online storage (of history data)**
  History data that are part of Historian database and accessible by programs.

- **OPC**
  OLE for Process Control; a defined set of interfaces, based on *OLE/COM* and *DCOM* technology.

- **OPC Server**
  OPC servers implement OPC COM objects and their interfaces. An *OPC* client can configure the rate at which an OPC server should provide the data changes.

■ P

- **Palette**
  A list of available colours, drawing and user-interactive controls etc for use in an Exaquantum/Explorer *document*.

- **PCS**
  Process Control System. See Control System.

- **PDC**
  Primary Domain Controller; a Windows network component that manages user account information in a centralised way.

- **PID**
  Proportional Integral and Derivative Controller. Most often used control scheme within control system. It provides three types of feedback control capability as the name suggests.

- **PIMS**
  Plant Information Management System. A computer system for managing information within a plant.

- **PLC**
  Programmable Logic Controller. A *PCS* using programmable logic devices.

- **Production Calendar**
  The business and operationally significant time periods which Exaquantum must be aware of in order to perform sensible calculation of *aggregation* values at the appropriate times.

- **Product Tools**
  Also known as *Configuration Tools*.

- **Property**
  A single *attribute* or parameter of an object.

- **Property (of an item)**
  The aspects of an *item* that can be changed.

- **Property Sheet**
  Displays the properties and events of the currently selected control or document. Allows the configuration of any parameters for a control.

- **PV**
  Process Variable.

- **PVI**
  Process Variable Input. A data and functional unit designed to monitor process status. Used in Control System.

■ Q

- **Quality Code**
  A 16-bit chunk of information (see *VTQ*). The lower 8 bits represent *OPC* status (Primary Quality). The higher 8 bits represent a sub-status (Secondary Quality).

R

- **Raw History**
  A history type used to address source of data within the RTDB.

- **Reference Data and Reference Items**
  These are static and rarely change. They are the fixed attributes of a *tag*, e.g. the Description, Engineering Units or Engineering Range.

- **Report**
  Any textual or graphical way of representing Exaquantum data.

- **Run Mode**
  Documents created in *Design Mode* can be run and interacted with in Run Mode.

S

- **SCADA**
  Supervisory Control and Data Acquisition (usually with *PLCs*).

- **Scripting Language**
  A programming-type language that allows Exaquantum/PIMS and Exaquantum/Explorer's capability to be customised and expanded.

- **Server**
  Computer holding master files and data for distribution to other computers.

- **Snap-to-grid**
  Enables objects in an Exaquantum/Explorer *document* to be aligned with a grid when building the *document* (in Exaquantum/Explorer *Design Mode*).

- **SQL**
  Structured Query Language; a universally used standard language developed by IBM for issuing queries/interacting with a database.

- **Support Tools**
  A group of tools developed for advanced users of Exaquantum.

- **System Log**
  A log file used to store all system activities.

- **SV**
  Setpoint Value.

■ T

- **Tag**
  A basic set of information within Exaquantum that is made up of *items*. There are three types of tags – OPC, Manual and Calculated.

- **Tag Generation**
  This is a set of four programs that create and update large numbers of *tags* at one time. The program used to perform the task is determined by the type of source information.

- **Tag Template**
  A set of information that describes structure and properties of *tags*.

- **Timestamp**
  The time-of-creation of an *item*.

- **Trends**
  Changes in values of measured parameters over a period of time, usually displayed in graph format.

■ U

- **Update Rate**
  Rate at which data is updated (usually in seconds).

- **UTC**
  Universal Time Co-ordinated. The same as Greenwich Mean Time (GMT).

- **User Interface**
  Programs that allow user interactions such as showing data, updating data and triggering functions to be performed.

■ V

- **Value**
  The part of the *VTQ* that represents the value of an *item*.

- **VTQ**
  Value, Timestamp, Quality; a triplet that comprises an item.

■ W

- **Wizard**
  A small program that assists the user in performing an operation, usually with pre-defined default values that can be changed if necessary.

- **Workbook**
  Workbooks are used to specify the active document, and to control the start up and configuration options of documents in Run mode.

- **Workbook Viewer**
  Shows the hierarchy of folders and files within the current workbook directory and in shared directories.

- **Workgroup Configuration**
  A type of Windows network configuration where user account information and access security control is configured in each of composing PCs.

- **WYSIWYG**
  What You See Is What You Get.

■ X

■ Y

■ Z

- **Zoom Feature**
  Enables the user to select an area of interest and enlarge it.

Blank Page

Revision Information

Title : Exaquantum Plant Information Management System Overview
Manual No. : TI 36J04A10-01E

Nov. 2000/2nd Edition/R1.01
All pages revised.

Written by	Yokogawa Marex Technology Limited
Published by	Yokogawa Electric Corporation 2-9-32 Nakacho, Musashino-shi, Tokyo 180-8750, Japan