

# Where no SCADA has gone before

## Real-time database 4.0

A key feature of **FactoryStudio** platform is its ability to be applied in hard Process Control (with real-time, in-memory information) and on Level 2 and Level 3 applications (where the information is stored on SQL databases and is exchanged with external applications).

**FactoryStudio's** real time core is an in-memory, event-driven database, which is the result of more than 25 years of continuous learning, and it is a completely new design that leverages current technologies with past experiences.



Real-time Information Management, IT, and SCADA combined

The function of a real-time database is to allow the system's modularity, creating a layer that allows you to isolate tasks that can communicate with relational databases, other systems, the field, user interfaces, and calculation and optimization systems. A real-time database's structure enables processes synchronization of real-time independent values and event notifications, and then updates the information table"

This manager uses components designed specifically for the .NET platform with infrastructure for tracking events. The following key differences are present when a comparison is made with a real-time level 1 supervisory:

HMI/SCADA databases	FactoryStudio real-time database
Basic Tags Types, such as boolean, int, float (real) and texts	Basic tag types plus the management of events, dates, tables, queries, and structured data
Server-centric and centralized module processing	Distributed processing, multi-core and multi-processes, as well as client and advanced control stations
Proprietary access Interfaces	Access via Interfaces classes.NET or web services
Definition of variables in databases on proprietary systems	A definition of variables in standard databases with support for SQL and ADO.NET
No hot swap (swapping the project configuration without stopping the application)	Allows hot updates with built-in version management

## Object Model and Namespaces

Most systems require creating a Tag or Variable for all internal properties as well as custom logic for your projects. **FactoryStudio** allows your application to directly access all business objects created in your project, i.e. user-created temporary tags are not required to manage PLC network nodes status, the total number of alarms in a group, or the number of rows in a dataset. Now you can access runtime and business objects (representing a network node), an alarm group or dataset, and then display the required information or take action directly through the object's built-in properties.

**FactoryStudio** has an underlying .NET object model with a 100% managed code, specifically targeting the development of Real-Time data management applications. The hierarchical object model includes the following top-level objects, which correspond to the main modules in **FactoryStudio**:

Tags	Dataset
Historian	Script
Security	Server
Alarm	Client
Device	Info

The top-level hierarchy is implemented as .NET Namespaces. Each Namespace has the .NET classes and objects that are created when building a project. These objects have runtime properties, methods, statuses, and configuration settings.

For instance, the Tag namespace contains all the tags that are in the application and each of the tags have built-in field properties, such as Quality, Timestamp, Min, Max, Units, etc.

### Examples

**Tag.tagName1.bit0, tag.tagName2.timestamp**

The same concept of the tag fields applies to all namespaces, for instance:

**Alarm.TotalCount, Alarm.Group.Warning.Disable:**

**FactoryStudio** has IntelliSense auto-completion for when you build a project, fill in input fields, or create scripts. This guides you to the any existing properties that are allowed for the object you are editing and allows you to easily "drill down" to a specific property.

When accessing a project object in the .NET Script Editor, it is necessary to prefix the namespace with an "@" symbol in order to avoid conflict with the names of the .NET local variables.

### Examples

In **Script > Tasks** and **CodeBehind**, use:

**@Tag.Analog1**

**@Device.Node.Node1.Status**

The **at sign** ("@" ) is not necessary on Grids and Dialogs. Some input fields may require objects of only one type, such as Tag or Display. For these, IntelliSense will automatically guide you to the required objects.

These concepts may seem abstract for users that do not have experience in .NET or similar object-oriented systems. However, the power of these concepts will become clear when users learn the engineering configuration tools and the **FactoryStudio** modules. When users get used to working with object models and IntelliSense, the users realize that there is a huge increase in productivity so they no longer want to work with systems that lack these features.

## SQL to the Core

### Built-in Embedded SQL

Every **FactoryStudio** system includes a full-featured, embedded SQL engine which provides the following advantages:

- It is a safe and secure location for your entire project configuration.
- It can be used as the historian database to log tags, alarms, and events on small to medium systems (up to 10GB of data).
- On large systems, it can be used as a local Store and Forward location for when the remote database is not available.
- It provides an ideal system to store local runtime settings, retentive information, local recipes, schedules and tables, and queries when preparing reports.

### Multiple Database Connections

**FactoryStudio** provides seamless integration with any third-party database, including **Microsoft SQL Server**, **Oracle**, **MySQL**, **Informix**, **SQL Lite**, **PostgreSQL**, etc.

### Real-time Queries and Tables

Any data source that supports ODBC, OLE-DB, or ADO. NET can be connected with the application; an easy syntax allows you to create or customize any query or table search with real-time tags.

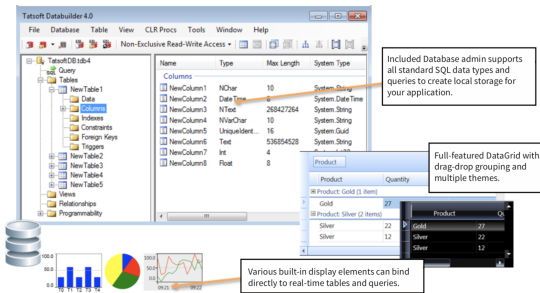
### Advanced DataGrid

**FactoryStudio** provides a fully featured DataGrid object. The DataGrid object presents tables and queries from databases as well as shows the contents of any tag, asset, or real-time object. Drop the table, query, or tag into the Grid Data Source to create front-end visualization, edit any real-time object, or database.



### The Convergence of SQL database, HMI software and .NET

Scheduling, Process Recipes, and real-time data consolidation made easy.



## Client-Server Architecture

Real-time queries can be processed either from the server or from the client computer, asynchronously or synchronously. To achieve better performance, multiple requests from distributed clients are cached and synchronized from the server.

## Data Gateway

Connecting client queries through Firewall protected security zones, such as moving data between the Automation Network and the IT network, is no longer an issue. **FactoryStudio** provides a built-in firewall friendly data gateway. Data queries from clients are securely routed through any **FactoryStudio** system.

---

## .NET to the Core

### Built-in Code Editor

**FactoryStudio** includes an integrated script editor for developers to create custom functionality for an application. The editor provides a powerful set of tools that help you test and evaluate your scripts. Debugging tools include assigning breakpoints, stepping into code, stepping over code, executing line by line, and watching values of objects change with each step.

Scripts are executed natively as managed code within the .NET framework, meaning you cannot create a script that would inadvertently cause the system to shut down. This provides a greater level of security and up-time for your application.

### Server and Client Domains

Using the **FactoryStudio** script editor allows you to create scripts that are executed on the server for a global reach, or they can be executed on the client side for a local reach

### Tasks, Classes and Expressions

You can create Tasks, .NET classes, and function libraries. In some cases, it may be more efficient or desirable to create one-line expressions, rather than full methods. For that purpose, FactoryStudio provides an expression editor which allows access to all .NET operands and classes.

## Object Orientation, Project Elements

All project elements (Tags, Datasets, Alarms, devices, and status of communications) are immediately accessible via IntelliSense. As these are native .NET objects, no temporary tags calls are required. With a single move, you can transfer data from Tags to .NET external data tables.

## C#, VB.NET and Code Translation

**FactoryStudio** includes industry standard languages, C# and VB.NET, so engineers do not struggle to use old proprietary, single-threaded, or interpreted scripting editors anymore.

Translate your code between C# and VB.NET anytime to better leverage your expertise.

## Events and Scheduling

Tasks and expressions can be triggered to execute by date, time, condition, calendar, tag change, or interval. Execution is distributed among processes. For maximum system security and performance, each process runs in its own application domain and is isolated from the real-time database.



### Complete debugging tools

You can do online project changes and configurations while running and debugging.