



Best Practices for System Integrators

InduSoft Web Studio

The requirements of your application may require deviations from this guide, but these best practices should help streamline the application development process and assist you in creating an efficient HMI or SCADA application for your solution.

Before you begin developing your application:

- Specify the tags database (main classes and arrays).
- Determine how the screen navigation will flow (links among the screens).
- Specify the application resolution. When the application is meant to run in different resolutions, develop it in the smallest resolution that must be supported by the application.
- Specify the platform where the application will run (e.g.: Windows 7, Windows CE, etc.) as well as the license type required for the project.
- Specify the standard language for the project, as well as the additional languages, if any, available via the translation tool.
- Configure all global project settings (“Project > Settings” menu).

As you develop your application:

- Assess risks associated with the development, deployment, and operation of the application and take actions to prevent or mitigate foreseeable issues. For example, if the application saves data into history files/databases, create a method to manage the data before running out of resources (storage memory).
- The application’s documentation (description and comments), as well as the tag names, must be written in English, unless explicitly requested otherwise for the end user.
- Write an objective description for each worksheet created in the project.
- Avoid creating file names with the space character (e.g. application name, screen name, recipe name or report name).





Best Practices for System Integrators

InduSoft Web Studio

As you develop(continued):

- Avoid using accentuation (e.g. ~, ^, ` , and so forth) in the application (including file names).
- Although InduSoft Web Studio is not case sensitive, it is recommended that you follow the same standard for tags and expressions throughout the whole application (e.g. UserName, Str2Asc(), and so forth).
- Avoid adding space characters at the beginning or at the end of labels, captions or descriptions. It may cause problems when translating the application to different languages.
- Keep in mind that the background tasks are executed on the Server only (not on the Thin Client stations).
- Avoid using synchronous methods that potentially take a long time to be executed, such as the Math() function, Wait() function, or recursive loops. Keep in mind that the execution of each task (thread) is sequential. For example, when calling the Wait() function in a Math(), the whole cycle of execution for all Math sheets is “paused” until the wait() function returns.
- Avoid calling synchronous functions that display dialogs (e.g. RDFFileN() function) from background tasks (math sheets or scheduler sheets). The task that executed the function will be paused until the user closes the dialog.
- Avoid using the Changed() function. Use the Scheduler module instead. The Changed() built-in function is kept in the product for legacy reasons only.
- Use the Execution field from math sheets to enable the execution of each math sheet only when necessary. Running a math sheet continuously (Execution=1) when it is not necessary is a waste of processing time.

Applications for touch-screen devices:

- Enable the Virtual Keyboard from the Project Settings dialog.
- Do not use the “Right-click” events and avoid using the Double-click event from the Command dynamic.
- Design objects that support interactions from the user (e.g.: buttons, scroll bars, sliders) with an appropriate size to be operated from a touch-screen device, which does not offer the same precision of a mouse pointer.





Best Practices for System Integrators

InduSoft Web Studio

Tag Database:

- Create classes in the Tag Database to encapsulate properties from a specific device, object or area of equipment of the process.
- Create array tags when the same property is applied to many instances of the same device, object or area of equipment of the process.
- Add a description to all tags created in the tags database. The description must be objective and provide minimum information for future maintenance of the project.
- Avoid creating tags to hold information that is available in the tag fields (e.g. HiLimit, AlrStatus, TimeStamp, Quality, and so forth).
- The tag name should be as short as possible, but it must be long enough to be meaningful, indicating the value that the tag handles.
- Avoid using retentive values or retentive parameters for many tags. Doing so may decrease the performance of the system, because the tag values must be saved in disk whenever the tags change of value. If it is necessary to keep the value of many tags retentive, study the feasibility of using the Recipe task to save the values at a suitable rate.

Graphics:

- Use groups of screens. Information that should be available continuously to the operator should be configured in one or more screens that are always open and visible. For instance, use a Header screen, which displays the logo of the company, date, time, username, main menu and so forth.
- Do not duplicate screens that are identical or very similar. Use features such as indirect tags and indexed array tags to display different information from the same screen.
- If the application must be translated to different languages during runtime, make sure that the interface is designed to accommodate the text in all supported languages—and not only in the original language used to create the screens.





Best Practices for System Integrators

InduSoft Web Studio

Graphics (continued):

- When designing screens with many static graphics, study the feasibility of using a background graphic (instead of several graphic objects) to increase the performance during runtime. If more than one screen uses the same background, use the Shared Image option from the Screen Attributes dialog.
- Consider using Linked Symbols to any interface that is used more than once in the application (e.g.: pump icons, PID faceplates, and so forth).

After finishing the development and test of the application:

- Execute the Verify Application command (“Tools Verify Application” menu) and check for errors in the Output (Log) window. Fix the errors, and run the Verify Application command again. Repeat this procedure until all errors are solved.
- Delete the trend history files from the \HST sub-folder of the application, unless this data cannot be lost.
- Delete the alarm history files from the \Alarm sub-folder of the application, unless this data cannot be lost.
- Create a backup of the application (compact the application folder, including all files and sub-folders), and store it in the appropriate place

