

## FAQ for InduSoft Web Studio

### Database & Security System

❑ **What does the Shared Tags folder store?**

📖 The *Shared Tags* folder stores the tags imported from the PC Based Control linked to the Studio application. The PC Based Control must be linked to the Studio application during the *New Project* wizard.

❑ **How to count how many tags are configured in the application database?**

📖 The amount of tags configured in the application is calculated in the **Information** tab from the **Project** → **Status** menu. Each array position and each class member of the tags configured in the Studio's tag database are counted.

❑ **How to visualize the application “Users” created with the function CreateUser() ?**

📖 Execute the following command: “<Studio Path>\BIN\Studio Manager.exe” “<Studio Path>\BIN\ExtUser.dll”. For example: “E:\Program Files\Studio\BIN\Studio Manager.exe” “E:\Program Files\Studio\BIN\ExtUser.dll”. This command will launch a dialog window from where you can visualize the users created by the **CreateUser()** function in order to create new ones or delete them.

❑ **How to shutdown the operating system automatically after closing the application?**

📖 Use the **ExitWindow(<numExitCode>)** function. According to the <numExitCode> value, the function executes different actions: 0=Restart; 1=LogOff and 2=ShutDown.

## Graphics

### ⇒ Active-X objects

#### □ How to insert and configure an Active-X object in a Studio application?

📖 To insert an Active-X object in a Studio application:

- Select the menu option **Insert** → **ActiveX object** (or press the **ActiveX Control** button from the **Active Objects** toolbar);
- Select the ActiveX object to be inserted in the application from the list box and press the **OK** button. The ActiveX object will then appear on the screen. (Unregistered ActiveX objects will not be available on this list box);
- Double-click on the ActiveX object and assign a **Name** to it (fill in the **Name** field). The dynamic Properties and Methods list can be viewed by selecting the **Methods** button. The static properties can be set by pressing the **Properties** button. A detailed description about the objects properties can be found in the component documentation, provided by the component developer.
- There are three functions to access the ActiveX component during the runtime:
  - **XGet(strName, strProperties)**: Returns the value of the properties *<strProperties>* from the object *<strName>*. The list of properties that can be read from the object are listed in the **Methods** dialog from the object, with the syntax *<Properties Name> (PropGet)* – for example: *Color (PropGet)*.
  - **XSet(strName, strProperties, Value)**: Writes the value *<Value>* to the properties *<strProperties>* of the object *<strName>*. The list of properties that can be set to the object are listed in the *Methods* dialog from the object, with the syntax *<Properties Name> (PropPut)* – for example: *Color (PropPu)*.
  - **XRun(strName, strMethod, Parameter1, Parameter2, ... , ParameterN)**: Executes the method *<strMethod>* from the object *<strName>*, according to the parameters *<Parameter1>*, *<Parameter2>*, ..., *<ParameterN>*. The list of methods available in the object is listed in the **Methods** dialog from the object, with the syntax *<Method Name> (Method)* – for example: *OpenFile (Method)*.

📌 **Tips:** Before inserting an Active-X component (usually an OCX file) into the Studio application, make sure it has been properly registered in the computer. You can register an Active-X object from Studio. Select the menu option **Tools** → **Register Controls**, press the **Register** button and select the Active-X file (usually an OCX file) which must be registered.

📌 **Note:** The amount of parameters set in the **XRun()** function can vary from 0 to 255, depending on each ActiveX component. You can use tags to set the parameters; however, the tag's types must match the component parameters type (Boolean, integer, string or real).

#### □ How to set a startup screen?

📖 Open the **Project Settings** dialog window from the **Project** → **Settings** menu and select the **Runtime Desktop** tab. Type the startup screen name in the **Startup screen** field.

#### □ How to insert a background picture on the screen?

📖 Right click on the screen and select the **Screen Attributes** option from the popup menu. Enable the **Enable Background** check-box and choose the adjacent picture format in the combo-box. Copy the picture file to the \Screen folder of the application and rename it to the same name as the screen (*<ScreenName>.SCR* file). Using the **Shared image** option, you can copy a bitmap file to the \Screen folder and share this picture with more than one screen. In this case, you must type the bitmap name in the **Share image** field.

## Tasks

### ⇒ Trend

#### ❑ How to convert the history trend to ASCII file?

📖 To convert a History trend file to ASCII format:

- Copy the file "<StudioPath>\BIN\HST2TXT.EXE" to the path "<ApplicationPath>\HST\".
- Execute the "<ApplicationPath>\HST\HST2TXT.EXE <FileName>.HST" program to convert the <FileName>.HST history file to ASCII format (for example: C:\MyApp\HST\HST2TXT.EXE 01010425.HST).

🚩 **Note:** After executing this command, two ASCII files will be generated:

- <FileName>.HDR = List of tags from the History file;
- <FileName>.TXT = Contents of the History file (value of the tags, date, time, etc).

📌 **Tips:** The **WinExec()** function can be used to execute the HST2TXT.EXE program from your application [for example: WinExec(InfoAppDir()+"\HST\HST2TXT.EXE "+TagFileName)].

### ⇒ ODBC

#### ❑ How to exchange data with FOX Pro by ODBC protocol?

📖 When exchanging data with a FOX Pro database, you must set the UseQuote=0 parameter from the [ODBC] section in the <ApplicationName>.APP file.

#### ❑ How to exchange data with Informix by ODBC protocol?

📖 When exchanging data with an Informix database, you must set the UseQuote=1 parameter from the [ODBC] section in the <ApplicationName>.APP file.

#### ❑ How to set a DATE field for ODBC interface with Oracle package?

📖 Configure the "Column" cells in the Studio ODBC worksheet with the syntax <ColumnName>.ts (for example: MyDate.ts).

### ⇒ Math sheets and Studio Language

#### ❑ How to execute a math sheet during the startup and other math sheet during the application shutdown?

📖 **Startup:** Execute a math sheet during the startup: Create a math sheet and fill its **Execution** field with the expression <TagName>=0 (for example: StartTag=0). In the last line of the math sheet, set the value 1 to the <TagName> tag. The <TagName> tag type should be Boolean.

📖 **Shutdown:** Instead of executing the **ShutDown()** function directly, execute one math sheet and configure the ShutDown() function in the last line of this math sheet.

## Communication

### ⇒ Drivers

#### □ How to insert a driver in the Studio's application?

- 📖 Right-click on the *Drivers* folder from the **Comm** tab and select the **Add/Remove Drivers** option. Select one driver from the list and press the **Select >>** button to insert the driver on the project.

📌 **Tip:** Each driver has its own documentation, which can be viewed by right clicking on the *Driver Name* folder from the **Comm** tab of Studio and selecting the **Help** option.

📌 **Note:** All drivers provided by Studio are automatically installed during Studio's installation. Each driver is composed of three files: DLL, MSG and INI. To insert a new driver in Studio (in case it has been created after installing Studio), just copy these files in the `\<Studio Directory>\DRV` folder.

#### □ How to set a “communication error” alarm?

- 📖 Configure a tag in the “Write Status” or “Read Status” fields of the driver worksheets and configure an alarm whenever this tag is different than 0 (zero).

#### □ How to communicate with a Siemens S7-200 PLC without using Prosave software?

- 📖 Siemens S7-200 PLC has a Freeport that can implement any protocol via PLC programming. There is PLC free software distributed by Siemens that implements Modbus protocol in the PLC Freeport (for further details contact Siemens support). You can use this software in the PLC and Studio Modbus driver (**MODBU**) to exchange information between them.

#### □ How to start and stop communication drivers during the runtime?

- 📖 There are three functions available to handle the execution of the communication drivers during the runtime:

- **Start all drivers configured in the application:**

Syntax: StartTask(“Driver”)  
Example: StartTask(“Driver”)

- **Start a specific driver configured in the application:**


Syntax: WinExec(“<StudioPath>\BIN\Studio Manager.exe”+ “+”<StudioPath>\BIN\Driver.dll”+ “+ “<DriverName>”)


For example: WinExec(Asc2Str(34)+“C:\Program Files\InduSoft Web Studio\BIN\Studio Manager.exe+ Asc2Str(34)”+ “+Asc2Str(34)+ “C:\Program Files\InduSoft Web Studio\BIN\Driver.dll”+ Asc2Str(34)”+ “+Asc2Str(34)+ “MODBU”+Asc2Str(34))

📌 **Note:** The **Asc2Str(34)** function is used to concatenate quotation marks for paths where there are space chars.

- **Stop a specific driver configured in the application:**

Syntax: EndTask(“Driver<DriverName>”)  
Example: EndTask(“DriverMODBU”)

 **Tip:** You can start or stop other tasks using the **StartTask(<TaskName>)** and **EndTask(<TaskName>)** functions.  
Example: **StartTaks("Viewer")** , **Endtask("Viewer")**.

 **Caution:** The drivers and tasks cannot be started or stopped during the runtime when running CEView (under the WinCE operating system).


## ⇒ DDE

### What are the parameters of Studio DDE Server?


 The Studio DDE Server and NetDDE Server parameters are shown in the table below:


Comm. Type	Application	Topic	Item
Network DDE	//<Computer Name>\NDDE\$	UNISOFT\$	<TagName>
Local DDE	UNIDDE	DB	<TagName>

### How to exchange data with Excel by using NetDDE?

 NetDDE can be used to exchange data, via the DDE protocol, between networked stations:


- Start the DDEServer module from Studio (**Project** → **Status** → **DDE Server**);
- Run Excel in the remote station;
- Open a Excel worksheet and fill the cells which must exchange data with Studio using the following syntax:  
=`"\<computer name>\NDDE$!"UNISOFT$!"<tagname>`. (For example: =`"\PC\NDDE$!"UNISOFT$!"second`).

 **Note:** When running under Windows NT or Windows2000, it is necessary to be sure that the services **Network DDE** and **Network DDE DSDM** are started. (Use the **Services** shortcut from the Control Panel to start these services).

 **Note:** When running under Windows 98, it is necessary to run the program `<WindowsPath>\netdde.exe` in both computers (for example: `c:\Windows\netdde.exe`).

## ⇒ OPC

### Is Studio OPC interface compliant with OPC specification v1.0a or v2.0?

 Studio OPC Client and OP Server modules are compliant with both OPC specification v1.0a and v2.0.

## General

### ❑ How to Start Studio automatically when the computer is powered on?

📖 Create a shortcut to the ...\<<Studio Path>\BIN\RunStartUp.exe in the *Startup* folder from the **Programs** group (... \WINNT\Profiles\All Users\Start Menu\Programs\Startup).

### ❑ How to disable Dr. Watson?

📖 To disable Dr. Watson under Windows NT:

- Execute the program <WindowsPath>\RegEdit.exe (for example: C:\WinNT\Regedit.exe);
- Select the path HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AeDebug;
- Set the value 0 (zero) to the "Auto" parameter from the path selected;
- Close the Registry Editor window.

⚠ **Caution:** You must take special caution when editing parameters in the **Registry Editor** program because some of them can modify the overall behavior of the operating system.

### ❑ How to modify the DATE format in Studio?

📖 There are two ways to modify the data format in Studio:

- Off-Line: Set the Order=<DateFormat> parameters (for example: DMY or DMY) and Separator=<Separator> (for example: / or .) from the [International] section from the \<StudioPath>\BIN\Program Settings.ini file;
- On-Line: Use the **SetDateFormat(<strSeparator>, <strDateFormat>)** function. For example:  
**SetDateFormat("/", "MDY").**

### ❑ What features are not supported by CEView?

📖 The main features not supported by CEView are: DDE, NetDDE and ODBC.

### ❑ What features are not supported by Web thin Clients?

📖 Web thin Clients provide a graphical interface to the users and allow them to monitor and set values of the process online. However, the Web thin client does not run any task or communication driver locally (Math, Scheduler, Report, ODBC, Drivers, OPC and DDE).


### ❑ What are the main steps to create a web-based application?

📖 Perform the following steps to create a web-based application:


- Develop the application locally. Don't use features that are not supported by Web Thin Clients (like Group of Screens) for the screens which will be saved as HTML format;
- After saving the screens in the standard format (**File** → **Save**), save the screens which must be available for the Web Thin Client in HTML format from the **File** → **Save as HTML** menu;
- Open the *Project Settings* dialog window (**Project** → **Settings** menu) and select the **Web** tab;
- Configure the **Data Server IP Address** field with the IP address of the Server station (station where Studio is running);
- Configure the path to the web files in the **URL** field. The web files are saved in the *Web* subfolder of the application. This path will vary according to the web root of your Server station (for example: http://<ServerIPAddress>/);
- Open the *Project Station* dialog window (**Project** → **Status** menu), select the **Execution Tasks** tab and set the *TCP/IP Server* module as **Startup=Automatic**;
- Open the application tags database and set to the **Server** option instead of **Local** in the *Web Data* column for all tags that must exchange value between the Server and the Web Thin Client station;
- Execute the **Tools** → **Verify Application** command to update the Project Settings in the HTML files;
- If there is no Web Server running in the computer, copy the program NTWebServer.exe from the *BIN* folder of Studio into the web root (for example: \<ApplicationFolder>\Web) and execute it. The path where this Web Server

program is executed will be the web root of the station. The Web Server is necessary to export data (web files) in HTTP protocol to the Web Thin Clients;

- Execute the application in the Server station;
- Using a browser (for example: Internet Explorer v4.1+SP1 or newer version) in the Web Thin Client station, type the URL address to download the screen which had been saved in HTML format (for example: `http://<ServerIPAddress>/<ScreenName>.html`).


 **Note:** The Web Thin Client requires an Active-X component (ISSymbol.ocx) to handle the screens on the browser. If the Web Thin Client is connected to the Internet, this component is downloaded and registered automatically. Otherwise, you must copy it to the \<OSPath>\System32 directory of the Web Thin Client and register it by executing the **regsvr32 ISSymbol.ocx** command. This file can be found in the \BIN folder from the Studio installation directory.

**How to send an email from Studio's application?**

 Follow the procedure below:

- Execute the **CNFEMail(strSMTP, strFrom, strPOP3, strUser, strPassword, numTimeOut)** function to configure the overall parameters used to send emails. After executing this function once, the parameters are kept in the system until the application is shutdown.
- Execute the **SendEMail(strSubject, strMessage, strTO)** function and/or the **SendEMailExt(strSubject, strMessage, strTO, strCC, strBCC, strFile1, ..., strFileN)** function each time that an email message must be sent. The difference between both functions is listed in the following table:

Characteristic	SendEmail() function	SendEmailExt() function
Execution	Synchronous	Asynchronous
Supports Subject text	✓	✓
Supports Message text	✓	✓
Supports TO addresses	✓	✓
Supports CC addresses	✗	✓
Supports BCC addresses	✗	✓
Supports attached files	✗	✓

 **Note:** Detailed information about the Studio's EMail functions can be found in the Studio Technical Reference Manual.

## Troubleshooting

- ❑ **The runtime task (TCP/IP, OPC, DDE, ODBC, etc) does not work.**
  - 📖 Make sure the runtime task is set to *Automatic* in the **Execution Tasks** tab from the *Project Status* dialog window (**Project** → **Status** menu). Select the runtime tasks which must be executed (for example: TCP/IP Server), press the **Startup** button and set to **Automatic**.
  
- ❑ **The Browser from the Web Thin Client does not display the screen and launches a warning message regarding ISSymbol.ocx.**
  - 📖 Make sure the runtime task is set to *Automatic* in the **Execution Tasks** tab from the *Project Status* dialog window (**Project** → **Status** menu). Select the runtime tasks which must be executed (for example: TCP/IP Server), press the **Startup** button and set to **Automatic**.
  
- ❑ **Browse of the Web Thin Client launches an error message missing the ISSymbol.ocx and does not display the screens from the Server.**
  - 📖 Issymbol.ocx is the Studio ActiveX object used by the browser from the Web Thin Client to view the web pages. If the Web Thin Client is connected to the Internet, the ISSymbol.ocx control is automatically downloaded and registered in the Web Thin Client station. Otherwise, you must copy it to the `\\WinNT\System32` folder of the Web Thin Client station and register it manually. Once it is registered, your browser will be able to access the pages.


🔗 **Note:** Use the `regsvr32 ISSymbol32.ocx` command to register the ActiveX component in the Web Thin Client.

- ❑ **The screens are shown on the Web Thin Client (Browser); however, the data (tags values) are not read from the Server.**
  - 📖 Make sure the parameter in the *Web Data* column from the application tags database is set as **Server** instead of **Local**. The tags set as **Server** keep the same value in the Server and in the Web Thin Client (Browser). The tags set as **Local** have independent values in the Server and in the Web Thin Client (Browser).


🚫 **Caution:** You must execute the **Tools** → **Verify Application** command after modifying the tags settings. Otherwise, the changes will not be updated in the web files.

- ❑ **The “On Up” expressions configured in the Command dynamic are not executed.**
  - 📖 The “On Up” expressions from the **Command** dynamic are not executed if the mouse pointer is dragged out the object area before releasing it. If the **Release** check-box from the **Command Object Properties** window is enabled, the **On Up** expression is executed even if the mouse pointer is dragged out the object area before it is released.
  
- ❑ **The Trend History does not work after adding or removing tags in the Trend worksheet.**
  - 📖 When a tag is inserted or removed FROM a *Trend* worksheet, the format of the history files (\*.HST) is modified. The same HST file cannot have two different formats, otherwise the data will not be retrieved from it properly by the Trend object. If you need to add or remove tags for history files, there are two valid procedures: Create a new *Trend Worksheet* or delete the old \*.HST files.
  
- ❑ **The value of indirect tags ( @<TagName> ) are not shown in the web thin client application.**
  - 📖 When a screen is saved as HTML, Studio saves a <ScreenName>.TAGL file in the `\\WEB` subfolder. This file has the list of all tags configured in the screen (objects and dynamics). When a screen is opened in the Web Thin Client browser, the tags listed in the <ScreenName>.TAGL are “enabled” for TCP/IP communication with the server station. It provides an optimized communication between the server station and the Web Thin Client stations.


Therefore, when using indirect tags (@<IndirectTag>), the tags pointed will not exchange data with the Server, unless they have been configured in the screen. In other words, the tags that will be pointed in the screen MUST be configured in any object of the screen to enable the TCP/IP communication for these tags with the server station.

 **Tip:** Add a transparent rectangle (no fill and no line) in the screen corner. Apply the **Command** dynamic to this rectangle and configure the tags (which can be pointed by indirect tags during the runtime in the Web Thin Client station) in the **Expression** field (keep the **Tag Name** field blank). So, these tags will be added to the <ScreenName>.TAGL file and they will be available for TCP/IP communication with the Server station.


**Newer versions of Studio are compatible with older versions of Studio?**

 When generating new versions of the product, Studio must keep compatibility with applications built in older versions of the product. However, in a few cases, it is not possible to keep full compatibility to some technological enhancements for the product. The known incompatibilities with older versions of the product are:

Older Version	Newer Version	Comment
V1.X	V2.x or newer	<b>Custom Object:</b> Studio v1.x allowed the user to create a special object called Custom Object. This object is no longer supported by v2.x or newer.
V2.x	V3.x or newer	<b>Transparent Color:</b> The transparent color applied to some bitmap objects may need to be reconfigured when running an application under Studio v3.x or newer, which had been developed originally with Studio v2.x or older. It does not crash the system, but some bitmaps can lose the transparent color setting. Also, only a few bitmap objects may require reworking. To configure the transparent color again, just select the bitmap object and move the internal track on the transparent color.
V1.x	V6.0 or newer	<b>Mnemonics:</b> The syntax #labelA:labelB was treated as a single label #labelA:labelB. In v6.0 or newer, #labelA is treated as a label (mnemonic) and labelB is treated as a tag name (value for the label).

 **Note:** Studio is able to open applications developed with older versions of the product. Nevertheless, Studio is not able to open and execute properly applications developed with newer versions of the product.

**Which functionalities are not supported by Pocket PC platforms (for example: IPaq, Cassiopeia, Jornada)?**

 WinCE devices powered Pocket PC does not support some functionality which are supported by WinCE devices powered by the “standard” Windows CE version:

**Functionality not supported by Pocket PC devices**

DCOM (Distributed Component Object Model): It means that all features based on DCOM (for example: remote OPC communication) are not supported by PocketPC devices.

CERasSvr.exe: This program was developed to allow the WinCE device receiving Dialup or Direct Connection incoming calls from remote stations. This feature is not supported by Pocket PC devices.

The **DialGetClientIP()** function does not work for Pocket PC devices.

□ **How to enable the “Hibernate” options from the operating system after installing Studio on a notebook?**

📖 Follow the procedure below:

- Run the "Registry Editor" (<Start button> \ Run \ regedit);
- Select the following path from the "Registry Editor":  
HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\Proteq\Parameters;
- The "IoPortAddress" parameter from the path mentioned above is set with the hexadecimal value: 0x00000111. Set this parameter with the hexadecimal address of the LPT1 parallel port of your notebook (for example: 0x00000378).
- Close the "Registry Editor" and reboot the computer.

📌 **Tips:** The hexadecimal address of the LPT1 parallel port of the notebook can be found in the Control Panel (System\Hardware\Device Manager\Ports (COM & LPT) \ Printer Port (LPT1)\Properties\Resources). Pick the initial address of the "I/O Range". Usually it is the hexadecimal address 0x00000378.