INFILINK-HMI

Industrial Automation Software

Infilink 4.06.01 New Feature Documentation

This document is to provide a description of features new to version 4.06.01; these are all the features added since version 4.02.02.

Document updated 16-September-2004

Table of Contents

Upgrade Cost 2
Edit Users in Run Mode 3
Importing Windows from Another Project 3
Retentive Tags 4
Active Unacknowledged Alarms Flashing 4
Telephony Scripting Commands 5
TextToSpeech: *SpeakOut* Scripting Command 7
Trend Printing 8
Animated GIF Object 9
Making bitmap objects fully / partially transparent 10
Window Printing 11
Gradient Fills and Transparency 12
SendKeys Scripting Command 14
Language Switching 17
Background image to HMI window 19
System tags: _FormattedDate, _FormattedTime 20
Script commands: _ThisWin.<Trend object name>.PenTag 20
Script commands: Log2, Log10 20
Script Command: DataToFile 21
Trend View Increased: 21
DDE Share Name Change 21
Round Scripting Command 21
OnScreen French Keyboard 22
Bug Fixes 22

Upgrade Cost

Lamonde is not increasing prices for these new features, and it is a free upgrade from Infilink version 4.00.20 and later to version 4.06.01.

Some of the new features require INFILRUN-0000 (unlimited tag version). Current users, and new purchasers, of INFILRUN-0000 have these features at no additional cost.

- Text-To-Speech
- Telephony
- Iviewer (5 license)

If you have one of the lower tag versions now, and want to upgrade, the upgrade price is reasonable. Basically, it is the price difference plus £31. Contact Lamonde for details.

The other new features are available in all tag counts of Infilink.

Edit Users in Run Mode

In the "user properties" dialog a check box was added, "Allow runmode editing of user table".

Editing of user table in runmode can be done in two ways:
1. Menu item "Users - Edit users".
2. Script command _Syscmd.EditUsers.

Menu item "Users - Edit users" will be enabled only when there is a logged in user has the permission to edit user table at runmode.

User table edited at runtime is saved as file “ProjectUsers.dat” in the project directory. This file is encrypted.

Runmode uses the table from “ProjectUsers.dat” and overrides the DesignMode table.

Runmode on exit creates the file “ProjectUsers.dat”. Runmode edited user table can be imported into design mode by importing the file “ProjectUsers.dat” by clicking the button “Import run mode user table” on “User list” tab of “Project properties”.

“ProjectUsers.dat” file is project specific. The file generated for one project will not be accessible to another project. This is done to prevent somebody from copying the file, creating another project and then getting user names and passwords.

In some cases it is possible that the Infilink design mode work is done in some office and run mode runs at different location. User table edited in runmode can be “Imported” by copying the file “ProjectUsers.dat” to computer running design mode. The file should be copied in the project directory. The file can be imported by clicking button “Import run mode user table” on “User list” tab of “Project properties”. As “ProjectUsers.dat” is tied to a specific project, importing it can be done only for that project.

Importing Windows from Another Project

Using design mode menu item “Window – Import” it is now possible to import HMI window made in another project. All the contents of source window (shapes along with their animations, trend / alarm objects, Window background picture if defined, animated GIF files if present, window scripts) are imported. Any tag names present in the imported window if not found in the project tag table will generate errors while compiling project.

Import window functionality is upward compatible for different Infilink versions.

Retentive Tags

While defining memory tags, check the check box “Retentive” to make it retentive.

Value of that memory tag at project close will be stored and will be assigned to that memory tag on next execution of the project, overriding its initial value setting.

Values of retentive tags are stored in file “ProjectRetentiveTags.dat” in project directory. This file is encrypted and is tied to a particular project.

While getting the tag values from the file “ProjectRetentiveTags.dat” Infilink uses “Group name”, “Tag name” and its data type to identify a particular tag.

Changing the data type of Tag1 (from Word to Long for example) causes the retentive value to be lost.

Error messages are posted to Eventlogger in the following cases:

- Retentive tags are defined in a project but “ProjectRetentiveTags.dat” is not found.
- “ProjectRetentiveTags.dat” is copied into project directory from different project.
- “ProjectRetentiveTags.dat” is corrupt.
- Runmode is unable to save file “ProjectRetentiveTags.dat” on project exit.

Active Unacknowledged Alarms Flashing

There is now a setting in the Alarm Viewer properties. Active Unacknowledged Alarms can be made to flash between two different colors, and the colors are selectable.

This provides more prominent notification of alarms.

Telephony Scripting Commands

Script command	Description
Ph_MakeCall	Make outgoing call.
Ph_DropCall	Drop current call
Ph_EnableIncoming	Enable incoming calls
Ph_DisableIncoming	Disable incoming calls
Ph_TTSString	Say a string over phone using text to speech
Ph_TTSFile	Say contents of file over phone using text to speech
Ph_PlayWaveFile	Play a wave file over phone
Ph_StopVoiceOut	Stop playing of wave file or text to speech
Ph_EnableDTMF	Enable sensing of digits pressed over remote phone
Ph_DisableDTMF	Disable sensing of digits pressed over remote phone

Syntax of all of above script commands is like: Ph_MakeCall("G1");
Each command expects a string expression / string constant in the parenthesis. This string is used to identify the memory tag group which is to be associated with that command.

Using a tag group name as a parameter to a command allows the user to work with multiple phone lines.

Tags used with above telephony commands.

Tag name	Data type	Usage
LineName	string	Specifying line / phone device name
PhoneNumber	string	Specifying phone number to call
CallStatus	long	Call status code
CallStatusString	string	Call status string
IncomingRings	long	Number rings after which incoming call should be answered
VoiceOut	string	String to use for Ph_TTSString / Ph_TTSFile or file name to use for Ph_PlayWaveFile
VoiceOutStatus	long	Code which gives the status of text to speech output or playing of wave file
LastDTMFReceived	long	Last digit received from remote touch tone phone
ErrorCode	long	Error code indicating last error.
ErrorString	string	String informing about ErrorCode
CallerID	string	Caller ID received for incoming call
Timer1, Timer2, Timer3	long	Generic timers, automatically incremented by Infilink at every second. Setting –1 to these tags stops incrementing operation.
IncomingEnabled	Discrete	Whether incoming calls are enabled or not.

Call status codes.

Code	Description
1	Call state Idle
2	Call state Offering (Incoming call)
4	Call state Accepted (Incoming call)
8	Call state Dialtone
16	Call state Dialing
32	Call state Ringback
64	Call state Busy
128	Call state Special Info
256	Call state Connected
512	Call state Proceeding
1024	Call state On Hold
2048	Call state Conferenced
4096	Call state On Hold Pending Conference
8192	Call state On Hold Pending Transfer
16384	Call state Disconnected
32768	Call state Unknown

Voice out status codes.

-1	Voice status undefined
100	Wave file playing in progress
101	Wave file playing completed.
102	Text to speech in progress
103	Text to speech completed

Defaults : For each telephone device Infilink assumes the following defaults: Incoming calls disabled, DTMF input enabled.

TextToSpeech: *SpeakOut* Scripting Command

SpeakOut (TextToSpeech, Repetition);

TextToSpeech: This is a string enclosed in quotation marks, or a string tag.
Repetition: This is the number of times that TextToSpeech is repeated. This is a number, or a numeric tag.

Example:
SpeakOut (MemGrp\SayIt,1);
This takes the text in the tag "SayIt", and sends it one time out the PC's speakers.

Script command syntax:
SpeakOut (string expression, nRepeatCount);
SpeakOut (string tag, nRepeatCount);
SpeakOut (string literal, nRepeatCount);

The string is spoken using the installed TTS engine. TTS engine is configured through control panel for settings like speed, voice type etc.

NRepeatCount has a valid range between 1 and 10 (inclusive). Attempt to program number outside this range results in clamping it to valid range.

Attempt to issue more SpeakOut commands before finishing the previous one results in queuing of the SpeakOut commands.

Script command syntax:
StopSpeaking;

Stop speaking / TTS. This command terminates any speaking going on and also empties all the queue if built.

SpeakOut and StopSpeaking commands operate independent of telephony commands. It is possible to have two different text to speech sessions, both over telephone line and over computer speakers.

Trend Printing

Brief notes about trend printing.

- Trend printing is done by using script command "_ThisWin.Trend.TrendPrint".
- Default printer is used for printing.
- This command prints real time and historical trends, whatever is seen on the screen as a "Trend Object" is printed on the printer.
- Any drawing objects appearing above or below the trend objects are not included in print.
- Works for both "Portrait" and "Landscape" modes.
- For historical trends, the inactive period black rectangles shown on the screen are printed with black cross-hatched brush. This improves appearance and reduces ink or toner usage.
- While printing the trend object, proper scaling is automatically done to fit the trend on the paper in both "Portrait" and "Landscape" modes.
- A printer typically offers more resolution (total # of horizontal and vertical dots) than a PC screen. It is likely that more detailed plot is obtained on the printed trend than is seen on the screen. Infilink plots the trend utilizing the higher resolution available on the printer. In the unlikely event that the printer resolution is less than the height and width of the trend, Infilink will not print the trend and it will post an error message in the event logger stating this.
- Using print spooling (instead of choosing "Print directly to printer") is best; otherwise Infilink program execution will slow while printing.

Animated GIF Object

- Select “Animated GIF” drawing tool, or select menu item “Draw-Animated GIF”.
- Draw a rectangle on the window.
- Animated GIF properties dialog appears.
- Select the Animated GIF file for this object.
- Check the check box “Animate in design mode” if you want to see the animation in design mode. Run mode will always animate the object, design mode gives you the choice.
- Click OK, you should see the animated picture.
- This object can be resized, moved, duplicated (etc.) just like other Infilink objects.
- If the object is being animated and if a zoom level other than 100 % is chosen, animation stops. Coming back to 100 % zoom should restore animation if it was on.

If the selected file does not contain valid Animated GIF (or mis-typing of file name) then you should see following bitmap indicating the failure to load the file.

Following animations are allowed for this object.

1. Show / Hide
2. Take Action
3. Enter Data

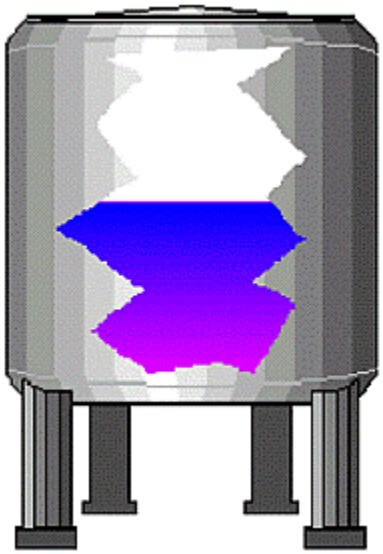
Animated GIF objects when being animated is always displayed on top of all other objects (irrespective of their Z level).

Making bitmap objects fully / partially transparent

- Create the bitmap object as usual.
- Go to “Bitmap properties” dialog, “Bitmap mask” tab.
- Select the bitmap file to be used as mask. This file has to be of type bmp (bitmap), it can not be in other image formats like JPEG, TIFF etc. (As opposed to basic bitmap object which allows most of the image formats to be used)
- Clicking OK shows the bitmap with transparent portions as defined in Mask bitmap.
- Mask bitmap is automatically stretched / shrunk in memory to the size of base bitmap before applying the mask.
- For all the “White” pixels (RGB values 255, 255, 255) found in mask bitmap, corresponding pixels in base bitmap are made transparent.

Used “Animated GIF” and “Mask bitmap” files are copied into project sub-directories “AnimatedGIFs” and “BitmapMasks”.

“Ragged Tank Cut-Out” has been a typical request which could not be done until this feature was added to Infilink:



Window Printing

In design mode user can print the HMI window by using menu item “Window – Print…” or by right clicking on empty portion of HMI window and selecting “Print…”.

The window is printed to the default printer.

User is able to print three lines of footer information. Infilink before printing presents default print lines and allows the user to edit them.

In run mode user prints the HMI window by using script command
“Print Window <Window Name>”

The <Window Name> must be a string literal. Only the visible window is printed to the default printer.

Gradient Fills and Transparency

This applies to following shapes:
Polygon, Rectangle, Rounded Rectangle, Ellipse, Pie, Chord, Button, Live and History Alarm Viewers, Trend.

Gradient filling of these objects is done by selecting the Shape Properties dialog and choosing the Interior tab. Previous method of filling objects is retained as “Normal fills”. New “Gradient fills” style is introduced. Checking this radio control results in gradient fill. Colors / gradient pattern / transparency / gamma correction is configured by clicking “Configure Gradient fills…” button.

Focus X and Focus Y edit boxes accept numbers between 0 – 99, both inclusive. These numbers specify percentage of center color portion in X and Y directions.

Using Gamma Correction option has also some specific effect on gradient.

Alpha level (transparency) of 0 – 255 (both inclusive) can be assigned to each color. 255 being least transparent, 0 means most transparent.

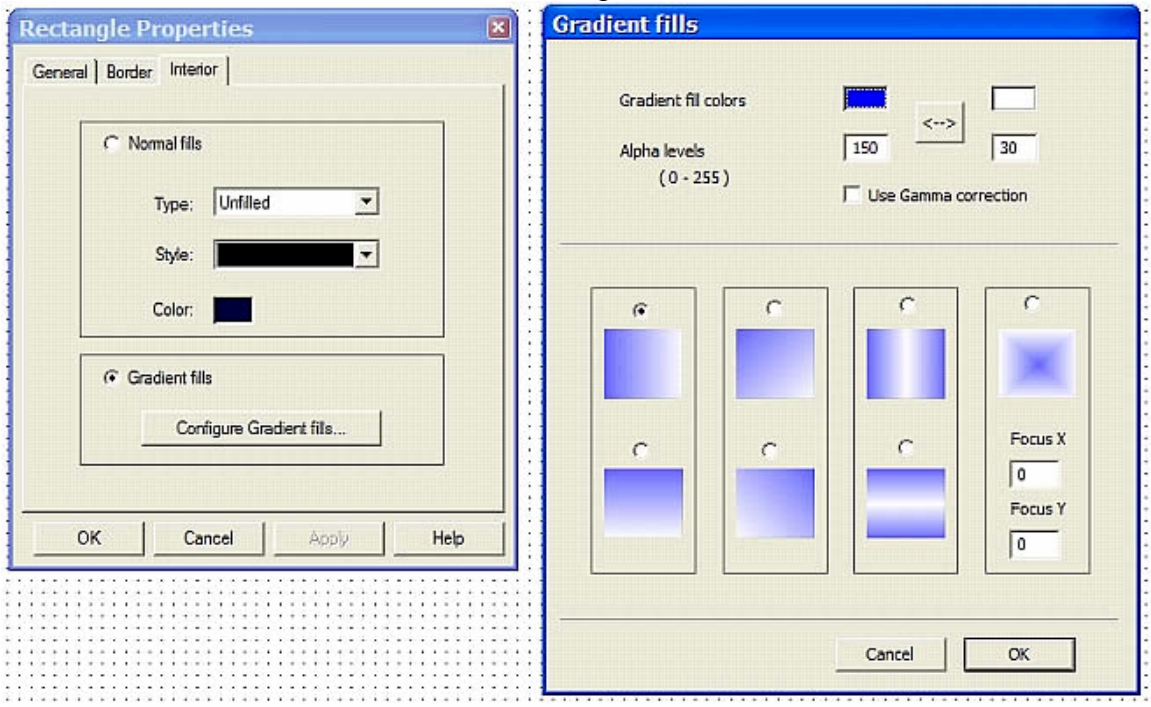
All the objects except Trends show proper transparency. Trends ignore alpha values.

Line styles like dashed, dash dot for objects filled with “Normal fills” and having line thickness greater than single pixel were resulting in a solid line. This is unchanged. But now with fill style “Gradient fills” selected, proper line style is available for all line thickness values.

Blink and Color fill animations do not work if object is Gradient filled.

Here is an example of a bitmap with a rectangle on top of it. The rectangle has gradient fills and alpha (transparency) properties:

This is the setup:



SendKeys Scripting Command

SendKeys(strWinCaption, strKeyStrokes, nFlags);
Send keystrokes to other top level windows.

Parameter	Data Format	Meaning
strWinCaption	Tag with datatype string, String expression, String Literal	SendKeys command sends the key strokes to the window(s) having this caption on the same computer as Infilink
strKeyStrokes	Tag with datatype string, String expression, String literal	String holding characters / keystrokes to be sent to target window(s).
nFlags	Tag with datatype other than string or discrete. Relational expression resulting in number, Number literal.	Indicates how strWinCaption should be used to find target window.

Examples:
SendKeys (“Notepad”, “helloworld<enter>”, 0);
SendKeys (strTag1, strTag2, longTag3);
SendKeys (strTag1 + strTag2, strTag3 + strTag4, lTag5 – lTag6);

Which keystrokes can be sent:
Generally all characters found on US English keyboard are sent as keystrokes to target window. To send the keystrokes like: “Enter” use syntax <enter>. Following table lists the keystrokes which can be sent in this way.

Keystroke	Required string to be entered, non case sensitive
Numpad 0	<Numpad0>
Numpad 1	<Numpad1>
Numpad 2	<Numpad2>
…..	…..
Numpad 9	<Numpad9>
“Shift” press	<Shift_Down>
“Shift” release	<Shift_Up>
“Control” press	<Ctrl_Down>
“Control” release	<Ctrl_Up>
“Alt” press	<Alt_Down>
“Alt” release	<Alt_Up>
Back space	<Back>
Tab	<Tab>
Escape	<Escape>
Home	<Home>
End	<End>
Left Arrow	<Left>
Right Arrow	<Right>
Up Arrow	<Up>
Down Arrow	<Down>
Enter	<Enter>
Insert	<Insert>
Delete	<Delete>
F1	<F1>
F2	<F2>
…….	
F24	<F24>

Examples:
1. String “John <home>F <home>Kennedy ” results in the following string when sent to Notepad application, “Kennedy F John”
2. String “ABC <left><left> DEFGHI ” results in the following string when sent to Notepad application, “A DEFGHI BC “

Explanation of nFlags:
Two attributes of window caption matching can be controlled:
1 Attempt matching given caption string **exactly or non exactly** with the target window caption.
2 Attempt matching given caption string to target window caption with **case sensitivity of not.**

Value of nFlag	Attempt Exact match	Case sensitive matching
0	No	No
1	Yes	No
2	No	Yes
3	Yes	Yes
4 or any other value	No	No

Keystrokes are sent to all the top level windows with matching caption. It is possible to send the keystrokes to single or multiple windows simultaneously.
Examples of nFlags usage.
Sending keystrokes to notepad window with caption "Untitled – Notepad"

Value of strCaption	Value of nFlags	Will keystrokes be sent to notepad ?
"notepad"	0	Yes
"Notepad"	2	No
"untitled – notepad"	1	Yes
"Untitled – Notepad"	1	Yes
"ntitled – Notepad"	1	No
"Untitled – Notepad"	3	Yes
"ntitled – Notepad"	3	No

Language Switching

General Concept:

Assume that the user is looking for creating runtime displays in "English" and "Spanish". (more than two languages are supported).

Infilink design mode allows the user to create a string table. It is basically a table with "n" rows and two columns since we are assuming two languages in this example. Row1 contains two strings, 1. "High Temp in English", 2. "High Temp in Spanish". Similarly other rows contains two strings each with same meaning.

Currently "Text object" contains a string. With this feature, user is able to assign a row number from string table to the text object. Script command "SelectLanguage (<Language Name>)" is available to the user.

Let us assume that user creates 10 text objects on a HMI window. At start of run mode these 10 text objects displays text from string table column 1. User executes script command "SelectLanguage (Spanish)" on a button press. After executing this script command Infilink replaces text of all 10 text objects by picking up strings from the Spanish column.

The display which was in English is now in Spanish.

Following Infilink objects will assume similar behaviour as of "Text object" because of language switching feature.

1. Button (text displayed on button), 2. Alarm comments. 3. HMI Window caption..

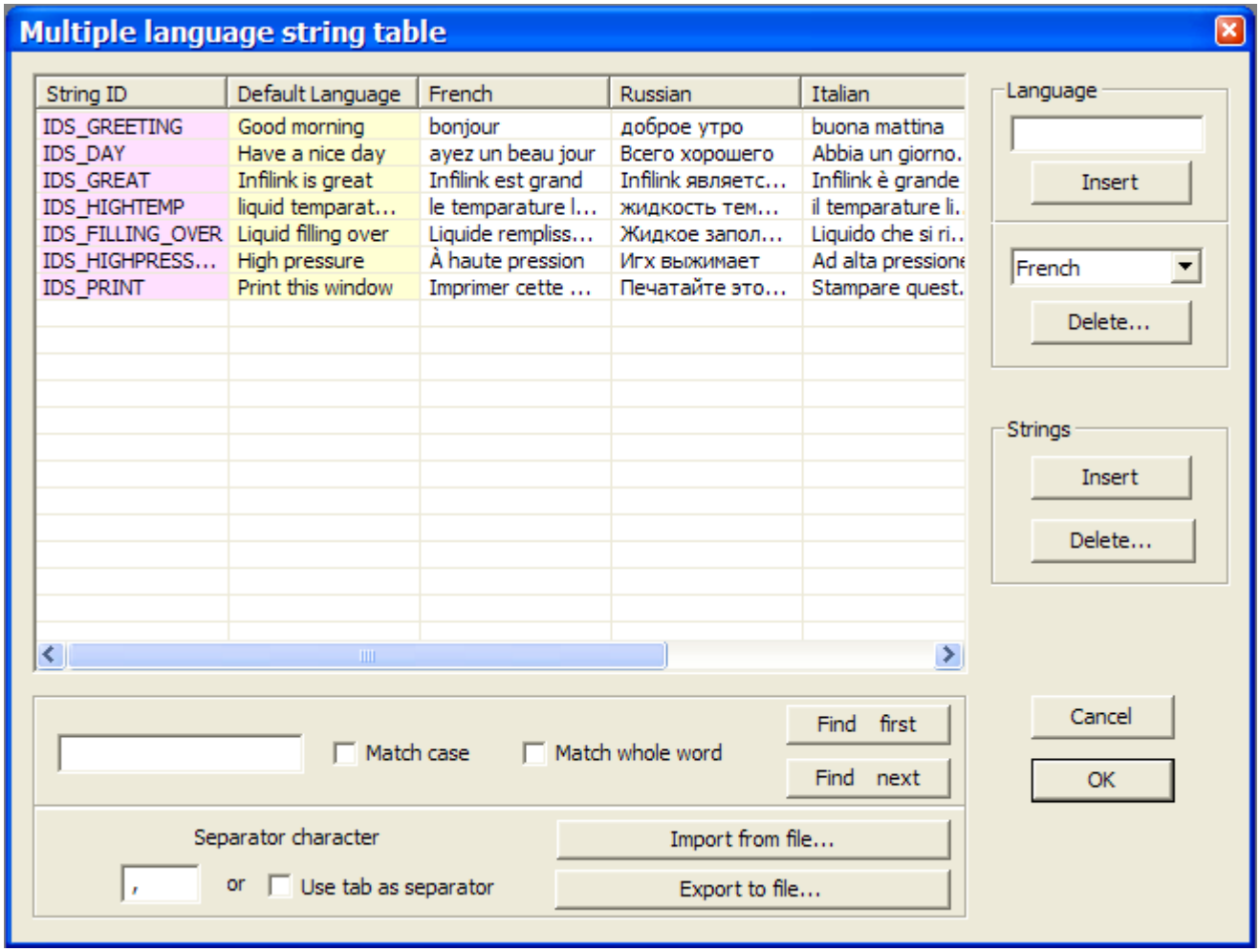
Design mode offers a string table editor. User is able to set required number of columns (Languages) and go on adding / editing rows.

Details:

User can create user interfaces (HMI windows) in such a way that in run mode the user interface is shown in any one of the configured languages.

User creates a string table in design mode using menu item “Tools – String table…”, a dialog box similar to following will appear.

Multilanguage strings can be import/exported to/from a CSV file. This allows most of the editing to be done within a spreadsheet.



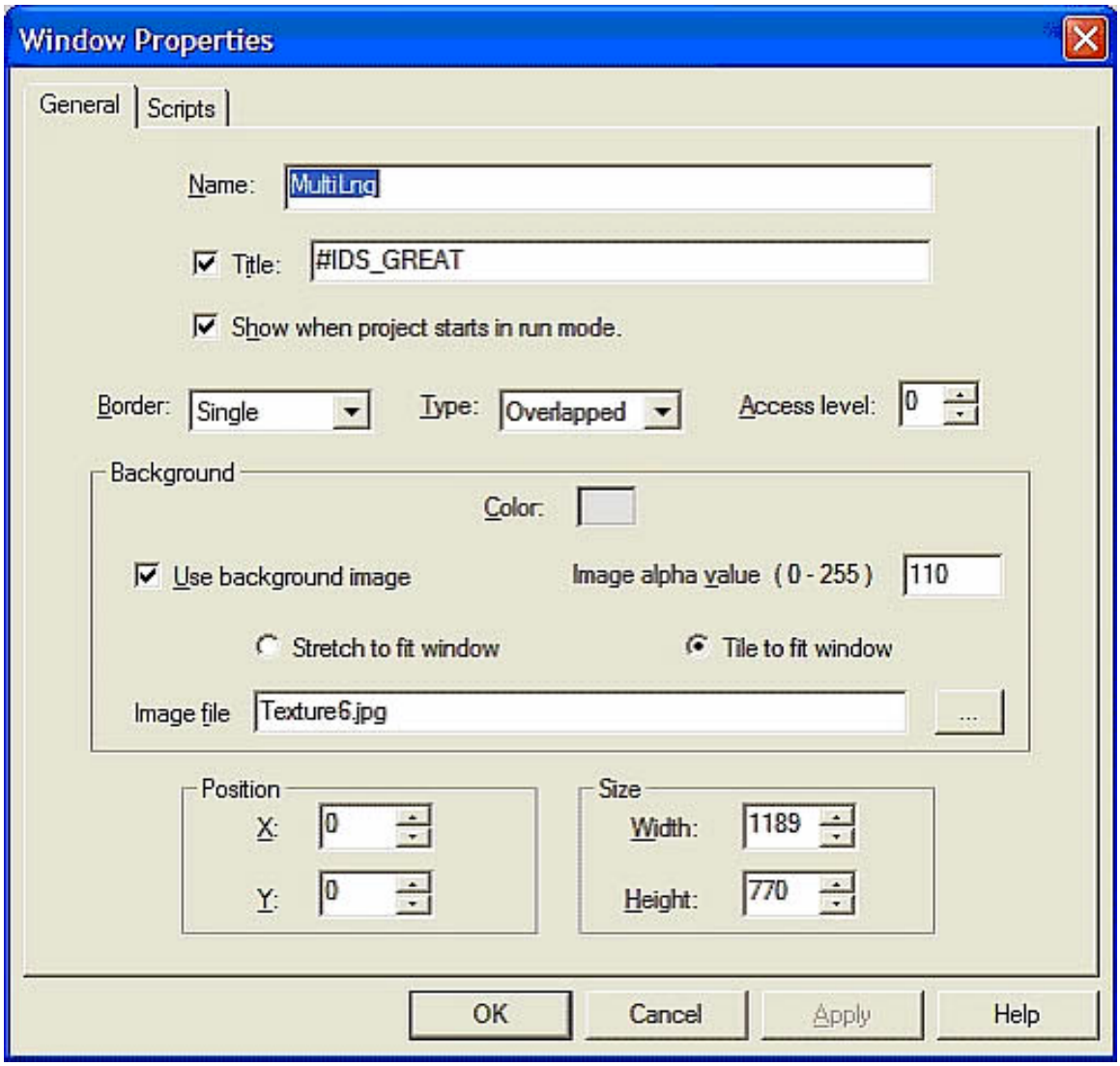
User can create many strings in multiple languages. Each string has a string identifier (String ID). Duplicate String IDs are trapped by Infilink.

Text object, button object, Window Caption and Alarm comment can be assigned a string ID instead of a string. While assigning string ID, character '#' should be put as initial character before actual string ID.
To assign a string ID of IDS_GREETING to a text object, #IDS_GREETING should be entered as text.
Infilink upon seeing first character as "#" will treat that string as string ID. At run time string associated with that string ID for currently selected language are assigned to the text object.

The script command "SelectLanguage <Language Name>" should be used to select current language. <Language Name> is expected to be a string expression. Passing an empty string to SelectLanguage command causes "Default Language" to be selected. The other languages are selected by passing the actual language name, which is the same as the column heading for that language. The string IDs and language names are not case-sensitive.

Background image to HMI window

An image in standard image format (BMP, JPEG, GIF, PNG) can be assigned as a background to HMI window.
While setting window properties in following dialog box, background image is selected.



The image can be used in semi transparent form by assigning “Image alpha value” less than 255. For semi transparent images the background color of the window is seen through the image. Assigning value of 255 to “Image alpha value” will make the image completely opaque and at that setting background color will not be seen.

System tags: _FormattedDate, _FormattedTime

These string tags will contain formatted “System Date” and “System Time”. The format is set in the windows control panel. These tags are typically used with show value animation for text object, to show nicely formatted time and date.

Script commands: _ThisWin.<Trend object name>.PenTag

_ThisWin.<Trend object name>.PenTag = <string expression> is used to assign a tag to a trend pen. <String expression> is used to get the tag name to be used with pen.

The pen with index set by _ThisWin.Trend.CurrentPen is affected by this command.

Example:
_ThisWin.Trend.CurrentPen = 3 ;
_ThisWin.Trend.PenTag = “LiquidTemp” ;

PenTag command does nothing in the following cases.

1. CurrentPen index is greater than the number of pens used for the trend.
2. Attempt to assign a tag name which is already used for any pen for that trend.
3. Invalid or non existing tag name.
4. Tag name expressed without group name and tag name not unique across tag groups.

Event logger will show error messages to inform about these error conditions.

Assigning a tag to real time trend will cause that pen to start plotting from beginning. Data logging is enabled for the tag if required when executing PenTag command.

History data for any tag which was not being logged before executing PenTag command will not be available for the time before executing PenTag command. Hence that data can not be seen in history plot. However data for time after logging enabled is available and is seen on history plot.

Assignment of a tag name to a trend pen is retained only while the parent window is open.

Script commands: Log2, Log10

Log2(<Numerical expression>) will cause log to the base of 2 to be calculated.
Log10(<Numerical expression>) will cause log to the base of 10 to be calculated.

Result 0 is assumed for input numbers of 0 and less than 0.

Script Command: DataToFile

_ThisWin.<Trend object name>.DataFileName
_ThisWin.<Trend object name>.DataToFile

User can save the data plotted on History or Real time trend by using script command _ThisWin.<Trend object name>.DataToFile.

Data for time period trend start to trend end (time corresponding to trend left edge to right edge) for all the trend pen tags is stored onto disk file.

The file name set by _ThisWin.<Trend object name>.DataFileName is used to store the data. If the file name is not set (i.e. DataFileName property of trend is an empty string) before executing DataToFile command then Infilink will generate a file name using system time and store the file in "Project directory\DataLog".

Example:
_ThisWin.Trend.DataFileName = "c:\InfilinkTrendData\May13.dat" ;
_ThisWin.Trend.DataToFile ;

The name set by DataFileName command is retained while the window is open. The same file name is used in subsequent executions of the DataToFile command. User is expected to change the file name before executing DataToFile command if overwriting of previous file is not expected.

Trend View Increased:

Now max time span 1440 hrs, Max pens 12.

DDE Share Name Change

Infilink while acting as NetDDE server was registering "INFILINK HMI" as DDE share name, it is now registering "INFILINK" as DDE share name.

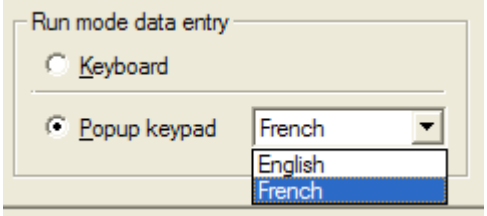
Round Scripting Command

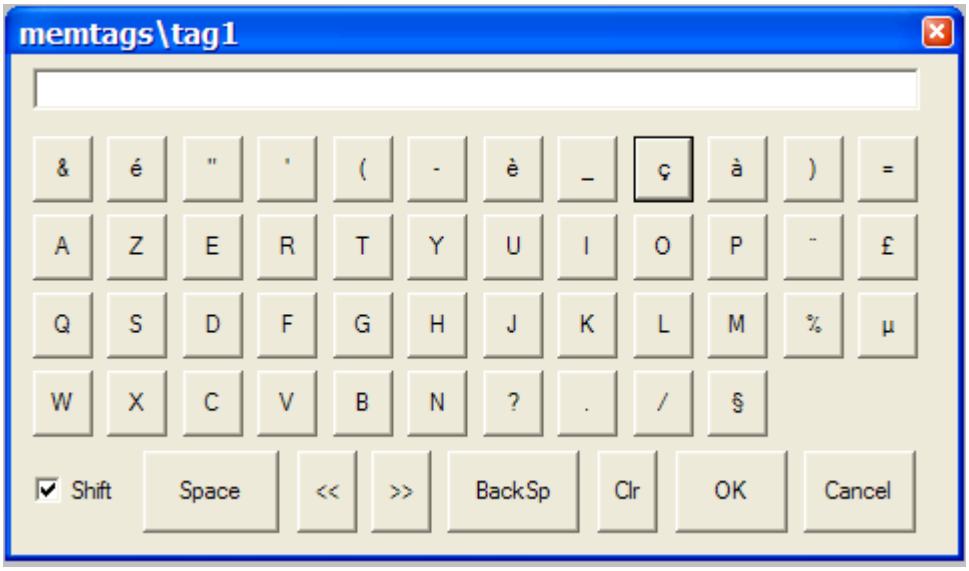
Tag1 = Round (Expression resulting in number);
Tag1 = Round (Tag1);
Tag1 = Round (Number literal);

Tag5 = Round (5.45) + Round (Tag1);

OnScreen French Keyboard

Onscreen French keyboard added. English or French keyboard can be invoked for data entry animation. Setting is made through project properties dialog in design mode.





Bug Fixes

Flat lines were being displayed on trends if System Time changed; this is now fixed.

If HMI windows were created with width and height equal to screen resolution then scrolling to rightmost and bottommost portions of those windows was not possible; fixed.

Infilink is now able to respond to iViewer information requests from computers running WindowsXP