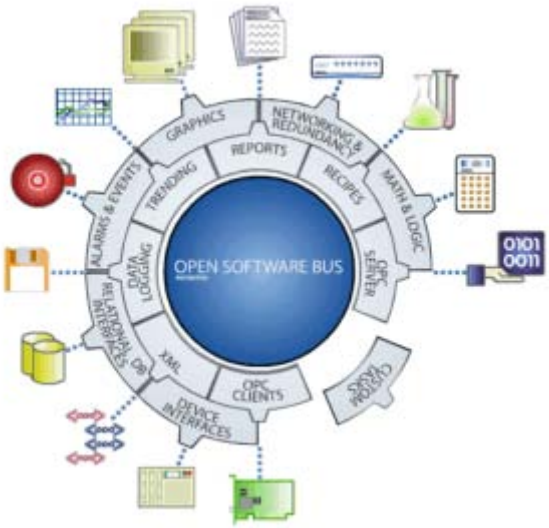
SIEMENS



FactoryLink System Utilities

for

FactoryLink 7.5.x

FINTsys

Version 2.0
Printed: Saturday, June 19, 2010

Introduction

The FLNTsys utility is a FactoryLink task, which enables NT or W2K system functionality's inside a FactoryLink application. These functions include enabling, disabling keys, request system information regarding disk space, screensaver settings, shutting down the system or logging off a user, logging event-message to the event viewer.

This manual is intended for use by a technician who is familiar with the FactoryLink ECS software. Covered in this manual are the principles of operation, installation and configuration of the software. Included in the shipment is a demo-application, which can be used as an example of how to implement the futures in your FactoryLink application.

This software is protected with a software key, a serial number and configuration sequence. Both are included in your shipment, registration is already completed on receiving the complete package.

FactoryLink is a registrated trademark of United States Data Corporation, Richardson Texas USA.

RLD Automation
Van Sonsbeeckstraat 11
5344 JB Oss
The Netherlands

tel. +31(0) 412 655 990
fax +31(0) 412 655 991
e-mail : support@rldautomation.eu

Contents

1 Operation principle

The FLNTsys utility is a FactoryLink task which enables NT4 or W2K system functionalities inside a FactoryLink application. These functions include enabling, disabling keys, request system information regarding diskspace, screensaver settings, shutting down the system or logging off a user, showing a simple message box to notify users.
The operation principle differs for the three different main futures of the application, keystroke filtering, system settings and (simple) message boxes.

After installation a demo application and client builder project are present in the {FLINK}\mps\en directory, the application is stored as a multiplatform save file with the name: 'flntsys.mps'. The demo client builder project is in the file 'flntsys.cba'. After restoring and starting the application and the client builder project a screen with an overview of the capabilities of FLNTsys is shown to the user, see the figure below.

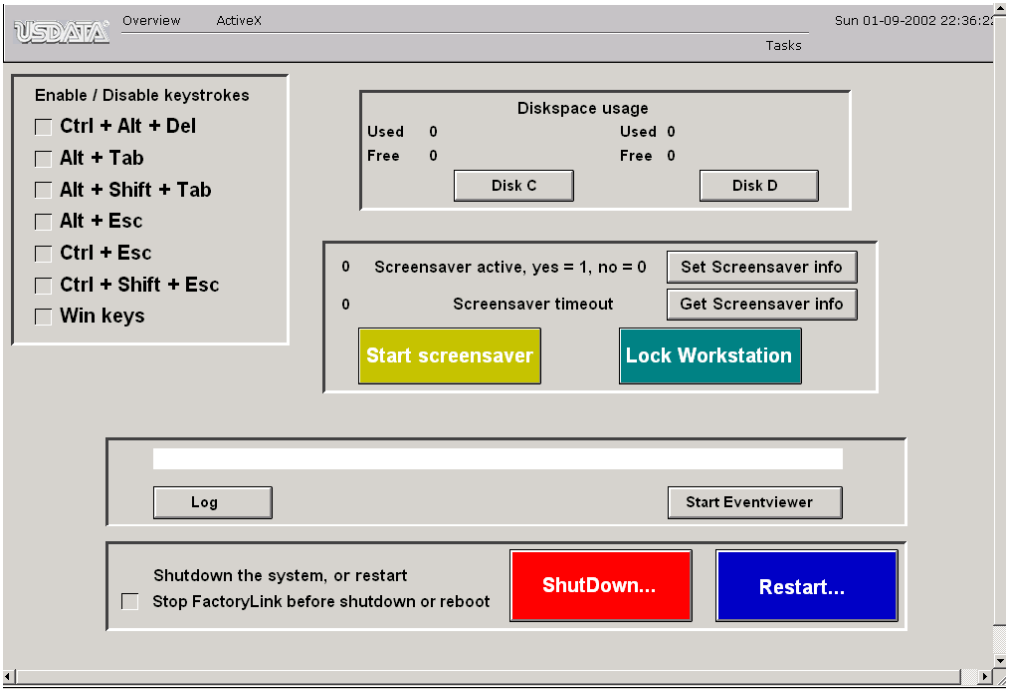


Figure 1.1. FactoryLink demo application.

1.1 Keystroke filtering

The FLNTsys utility is capable of filtering keystrokes from the keyboard buffer. The utility lets you specify which keys are to be filtered, these keystrokes can be altered from within a FactoryLink application. If a keystroke is filtered none of the (running) applications will receive this keystroke, it is removed from the keyboard buffer, so even before any program 'sees' the keystroke. In order to do this the utility depends on a device driver: KbFilter, which stands for Keyboard Filter. This device driver is just a normal device driver, its start-up options can be modified from the "control panel → Devices → Keyboard filtering" for NT4, or "Computer management → Device manager → Non-plug and play drivers → Keyboard filter for W2K".

As this driver attaches itself to the keyboard, it has the same class as a keyboard driver. This means you can start the driver manually (see the next figure), but you are not allowed to stop this device driver! NT will not allow you to stop a device driver of the keyboard class.

1.1.1 ActiveX keyboard filter component

The FlNTsys utility is designed as a FactoryLink task, and therefore allows you to filter keyboard events on the server which is running the FactoryLink application. The FactoryLink task can not filter keyboard events on a client system, only running a client builder application. In order to enable keyboard filtering on a client

system an ActiveX component, included in your shipment, should be installed on the client system. Together with the ActiveX control the keyboard filter device driver is installed.
With the ActiveX control “KeyboardFilter Control”, keyboard filtering is possible from within a client builder application. Enabling and disabling of keystrokes can now be done from within the client builder application, using the scripting options of client builder.

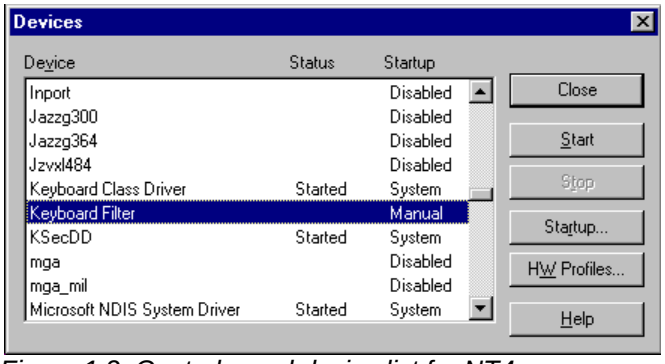


Figure 1.2. Control panel device list for NT4.

1.2 System settings

Several system settings can be queried or modified, they are divided in three main groups.

- Shutdown or logoff the system.
- Query information about free and used disk space
- Screen saver options

The shutdown and logoff commands allow you to logoff or shutdown, even reboot the system from within a FactoryLink application. Prior to shutting down or logging off, the currently active FactoryLink application is gracefully stopped.
Information about used and free disk space can be queried for the disks in your workstation. The results of a query are integer values representing the amount of free or used disk space in kBytes. The information is available in a standard FactoryLink tag of type analog or long analog.
The screen saver options allow you to activate the screensaver setting for the display. The option present the setting and querying of the inactivity time for a screen saver, the value is in seconds. Also the automatic activation of the screen saver can be enabled or disabled.

1.3 Event viewer messages

Messages can be written to the NT event log, these messages can be informal, warning or error messages. The messages are viewed with the standard event viewer of Windows NT.

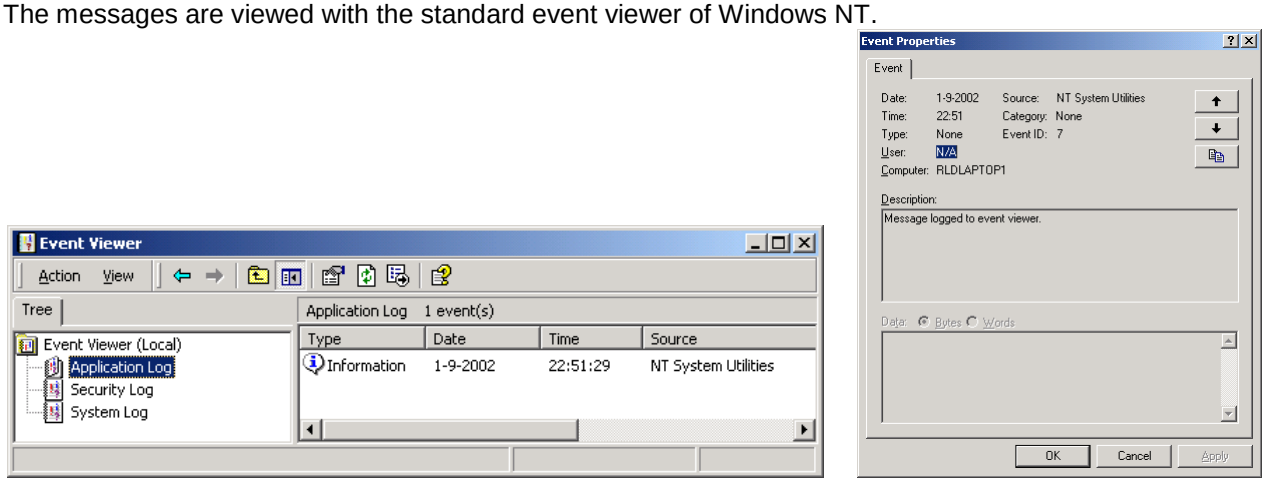


Figure 1.3. Logging to the event viewer.

2 Installation

2.1 System requirements

The FLNTsys utility will install and run on any IBM PC or PC-compatible computer equipped with:

- A processor equivalent to an Intel 80486 or later. RLD Automation recommends at least an Intel Pentium-class or compatible processor.
- A diskette station.
- At least 1.5 Mb of free hard disk space.
- Microsoft Windows NT 4.0, with at least service pack 4 installed; or Microsoft Windows 2000, with at least service pack 1 installed.
- USDATA FactoryLink ECS 7.0

2.2 FLNTsys installation

The main objective of the FLNTsys installation is copying two groups of files from the installation media to your workstation. In the first group all the files related to the FactoryLink environment are copied to your FactoryLink system directory (normally set by the 'FLINK' environment variable), including both the runtime and configuration options. The second group consists of the files for the keyboard filter device driver; those files are placed in the Windows NT system directory.
To start the installation locate the 'setup.exe' file on the installation media. Make sure that you are logged in as a user with administrator rights, otherwise the installation will be aborted. Installing a device driver requires administrator rights!

Figure 2.1. Setup the application (large icon view).

After starting the setup, some questions need some answers, most important one is where your FactoryLink system directory is located. The setup assumes that the environment variable %FLINK% is correct, but gives you the chance to change the directory. All the steps in the installation process are presented in the figures to come.

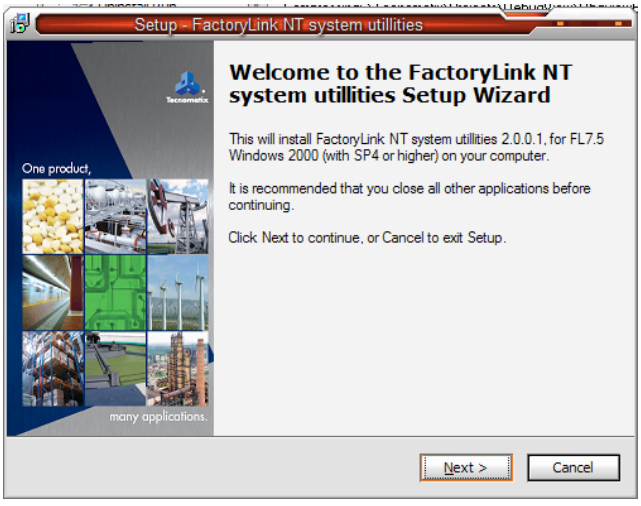


Figure 2.2. The welcome wizard panel.

The first screen displayed is the 'Welcome' screen, click 'Next >' to continue. The next wizard panel displays the FLNTsys end-user license agreement. read this agreement carefully. If you install FLNTsys you agree to abide by the terms of the license.

If you do not agree to the license terms, click 'I do not accept the agreement'. Setup will quit immediately. Otherwise, click 'I accept the agreement' followed by 'Next' to continue.

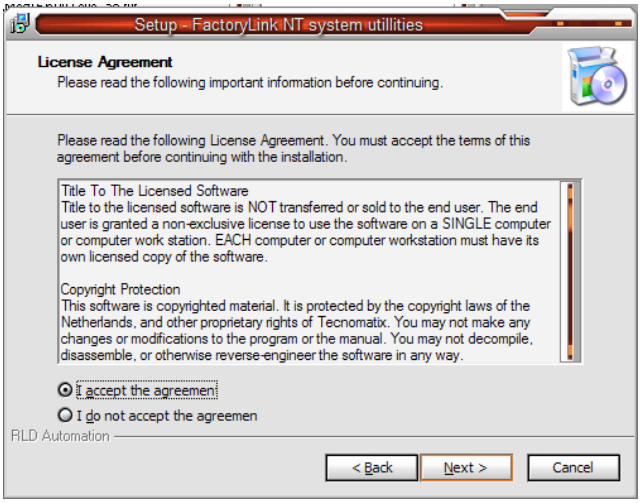


Figure 2.3. The software license agreement.

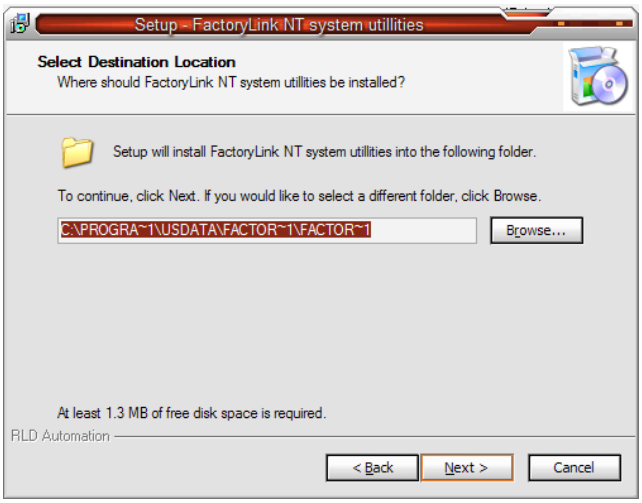
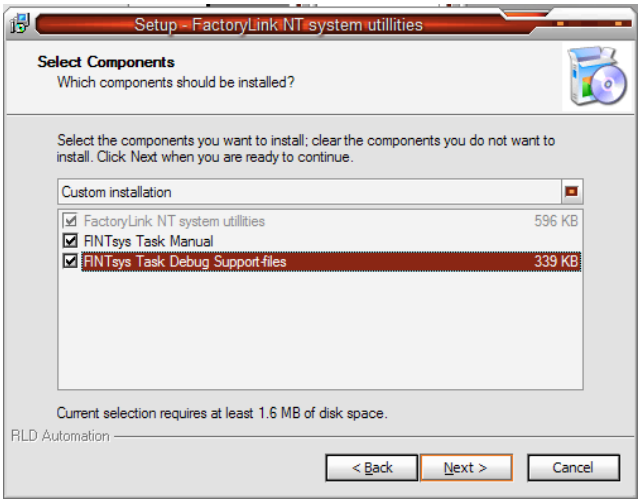


Figure 2.4. Select destination directory



Use the browse window to specify a directory for the FactoryLink system files, in case the one selected by the set-up wizard is not the one where you want to install this FactoryLink task.
Next you have to choose the setup type: a full installation or a custom installation. for a custom installation you can select the server and/or client part of the FINTsys utility. the server part of the application is the FactoryLink task and configuration environment for the Configuration Explorer. The client part is the ActiveX control to enable or disable keystrokes from within a Client Builder application. The keyboard filter driver which hooks to the system keyboard driver is always installed, as are the documentation files.

Figure 2.5. Select the type of installation

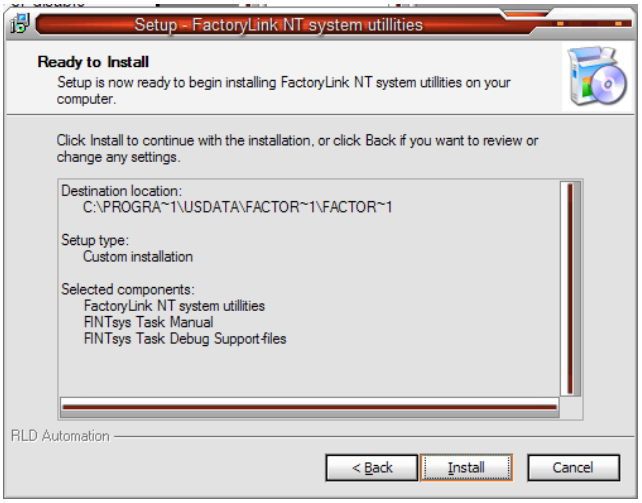


Figure 2.6. Ready to install the application.

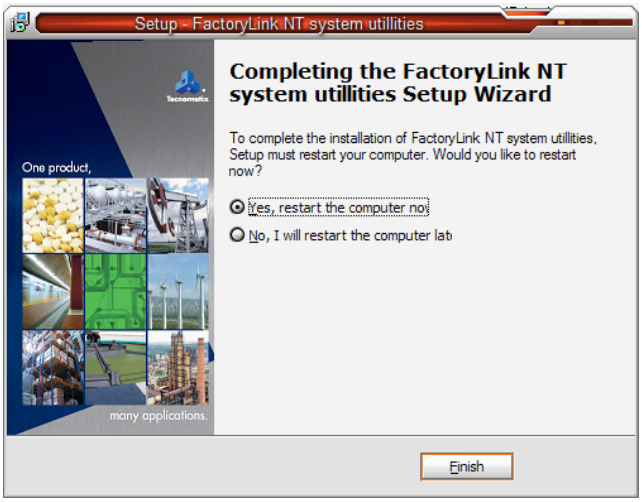


Figure 2.7. Installation is completed.

You need to confirm your settings and requirements, before the actual installation process is started. Setup requires you to restart your computer in order to complete your FLNTsys installation and to ensure that the Keyboard filter driver is recognized by your system. If you have other work you must do, select 'No, I will restart the computer later', then click 'Finish". Otherwise select 'Yes, restart the computer now', then click 'Finish' to reboot your system.

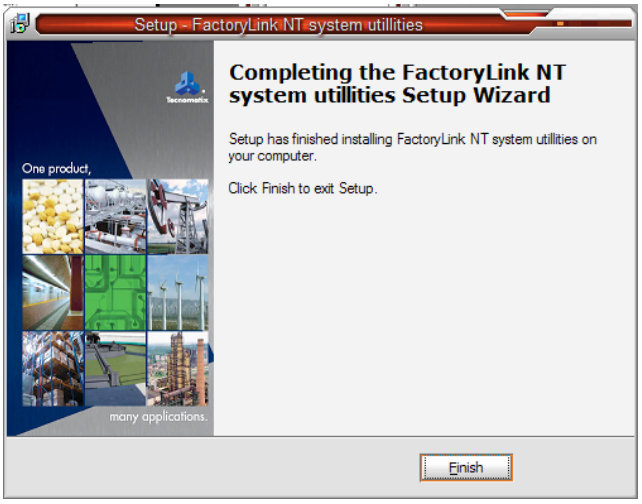


Figure 2.8. Finish the installation (no restart required).

2.3 Keyboard filter start-up

The keyboard filter device driver is responsible for filtering the keystrokes, before any application gets the keys. If this device driver is not installed or running, the FLNTsys application will not be able to filter any keystroke. After installation the keyboard filter device driver has a start-up priority of manual. This means you need to start the device driver manually, see the figure below. Normally this priority can be set to system or automatic. Remember that you need administrator rights to change the start priority, and to manually start the driver. Set startup priority to 'Automatic' if you want the driver to start everytime the system starts-up.

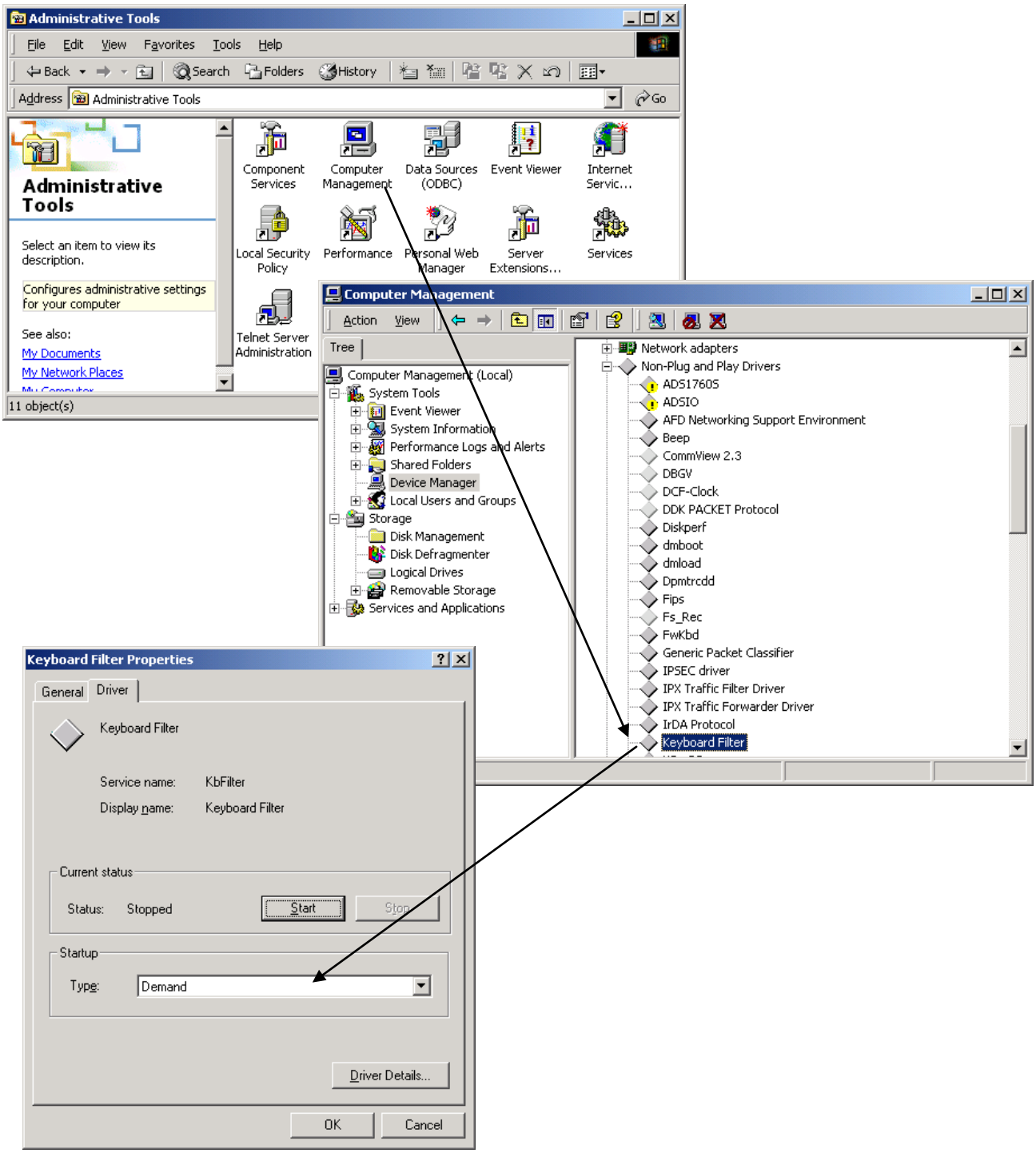


Figure 2.9. Start-up priority of the keyboard filter device driver for Windows2000.

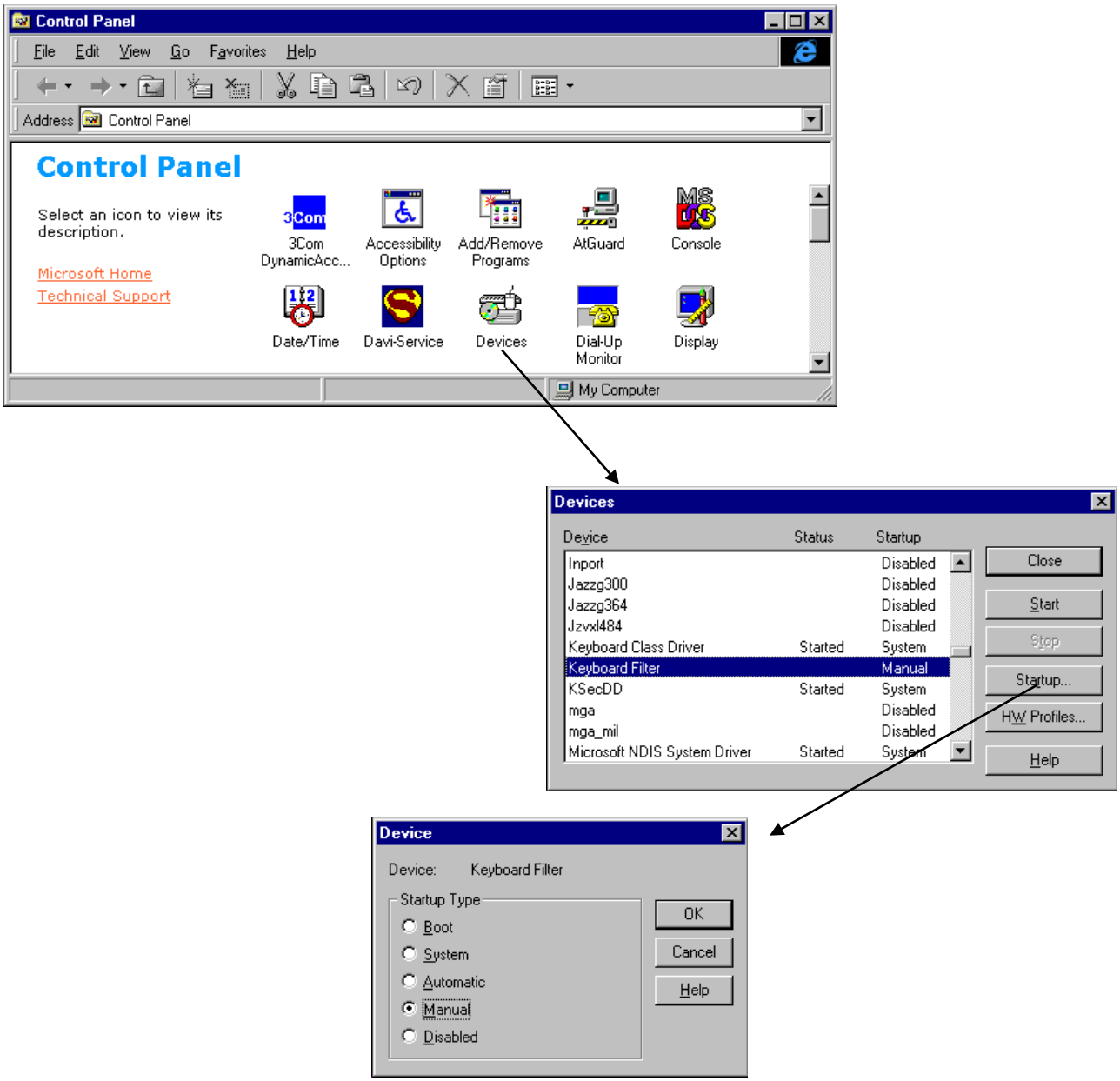


Figure 2.10. Start-up priority of the keyboard filter device driver for NT4.

If after installation the registry settings are not set, you can install the driver manually by running 'regedit kbfilter.reg', remember that you need administrator rights. The file with the registry settings can be found in your windows directory: %Systemroot%\system32\drivers.

2.4 FactoryLink task configuration

In order to activate the utility as a FactoryLink task it needs some configuration in the 'system configuration' table of the FactoryLink Configuration Explorer.

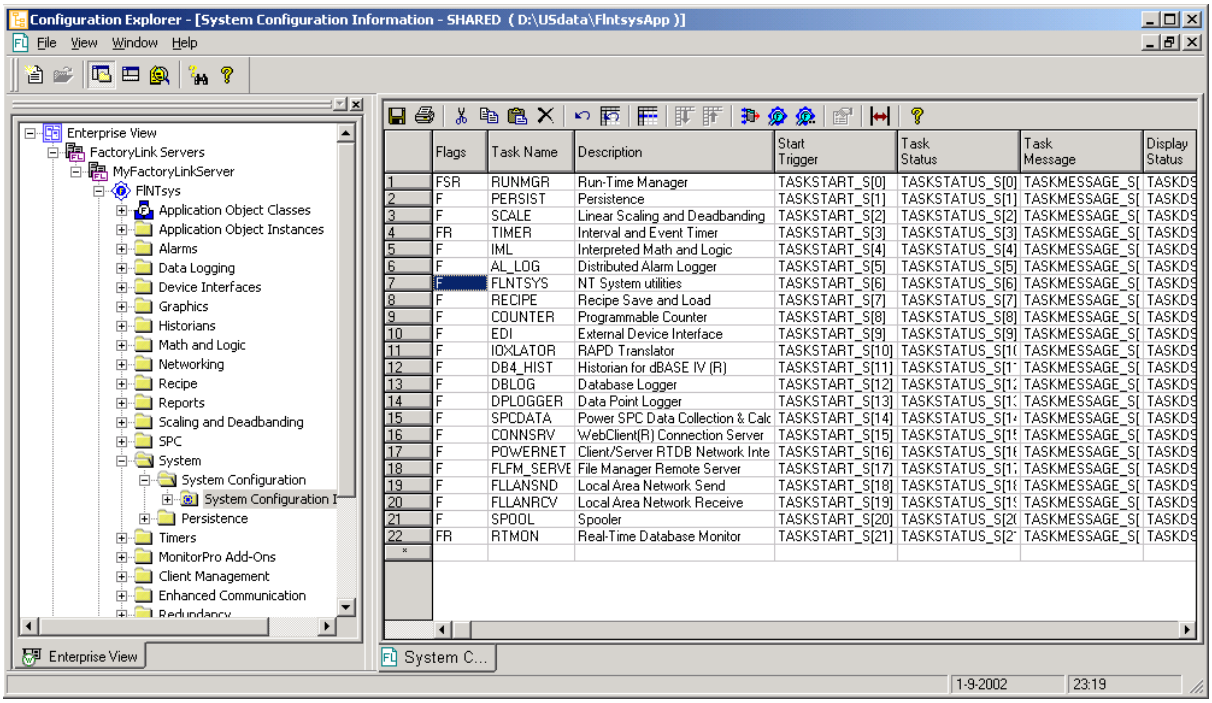


Figure 2.11. FactoryLink task definition.

The required settings for the FLNTsys task are its taskname and executable name, the last one is not visible in the figure above, but can be reached by panning through the configuration row.
the following settings apply:

Task Name	=	FLNTSYS
Executable File	=	bin/flntsys

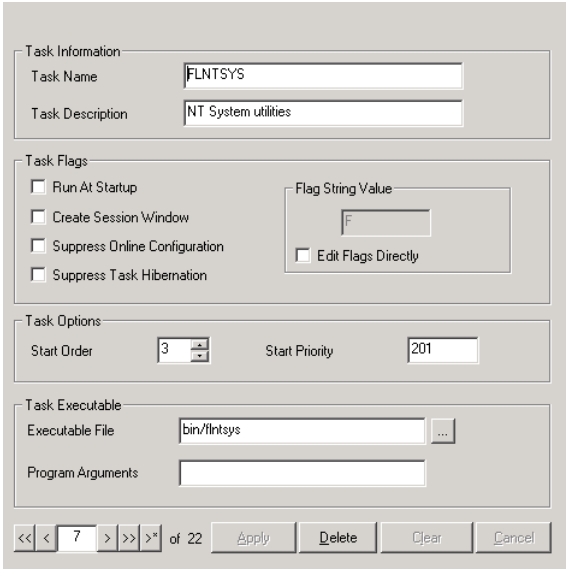
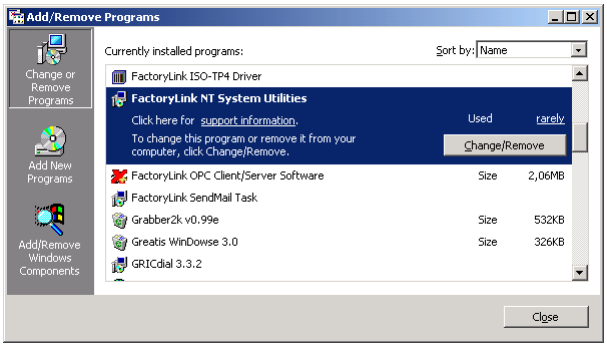


Figure 2.12. FlNTsys task definition.

2.5 FINTsys task de-installation

De-installation of the FINTsys task is done through the Control Panel applet 'Add/Remove Programs'. The FINTsys task has an entry in the list of applications, which can be de-installed: 'FactoryLink NT System Utilities'.



Information window for the FINTsys task

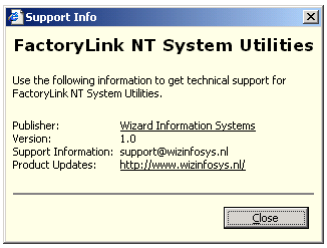


Figure 2.13. Control Panel applet for removing the FINTsys task.

Clicking on the button Change/Remove allows you to remove the FINTsys task from the system. Before the deleting of files starts, you are asked to confirm the removal of the application.

2.6 FINTsys INI file

The FINTsys module uses an INI-file, located in the FactoryLink 'bin' directory (%FLBIN%), and is named im_bas.ini. The location makes the settings in this file system wide, and not 'application settings'. Note that this file is not saved in an application save file! It should be reviewed if the target system differs from the development system.
This behavior can be considered as a limitation, and to overcome this limitation the INI-file can be part of the FactoryLink application. Placing the INI-file in the %FLAPP%/net directory makes the INI-file part of the application, and will be restored together with the application. The loading order of the settings in INI-file is the following order:

1. INI-file located in the net subdirectory of the FactoryLink application
2. INI-file located in the FactoryLink system (%FLBIN%) directory
3. A default value, hard coded in the FINTsys module

INI-file: im_bas.ini

```
# Debug section
# The level 1..5 is defined with DebugLevel.
# Seperate logging to file is requested with Log2File
# Seperate logging to the event viewer is requested with
Log2EventViewer

[Debug]
DebugLevel=4
Log2Screen=0
Log2DebugViewer=1
Log2File=0
Log2EventViewer=0
UseBlackBox = 1
UseDeadLockDetection = 0
# Task section
# A task can be forced to use polling mode with Polling,
# this is the opposite of event based.
# The polltime is a value in millseconds the task will
# sleep before a new search for command is done
# A protocol driver normally uses two task id's to exchange data
# with the FL kernel, if desired this setting can be overruled
# with the key Driver2Ids set to zero, the task switchs
(automatically)
# to forced polling mode.
```

```
[Task]
Polling=0
PollTime=5
Driver2Ids=0

[DiskSpace]
UseMB=1
UseGB=1

[TimeZone]
TimeZoneName="W. Europe Standard Time"
```

An INI-files has sections, and in every section one or more key's, every key is associated with a setting. Sec¬tion names are enclosed in brackets []. A key and it's value are separated by the '=' character.

2.6.1 Section: [Debug]

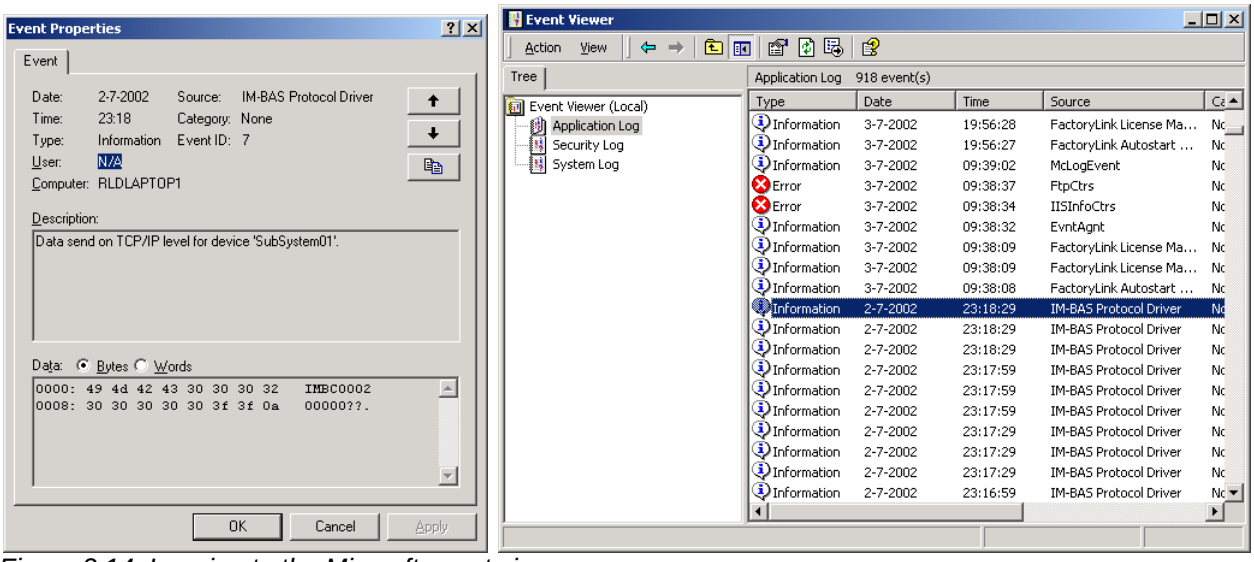


Figure 2.14. Logging to the Micrsoft event viewer

This section defines if and how many debug information is generated. The key DebugLevel defines if and how detailed the debug information will be, a level of zero disables debugging info. The maximum level is 5, and the highest level includes the data send and received from devices. The lowest level is level 1, for this level only the run-time manager messages are used as debug information.
Normally the debug information is displayed in a debug window of the Configuration Explorer, or to a command window if FactoryLink is started from the command prompt; for debugging information in the Configuration Explorer or the command window the key 'Log2Screen' must have any other value then zero.

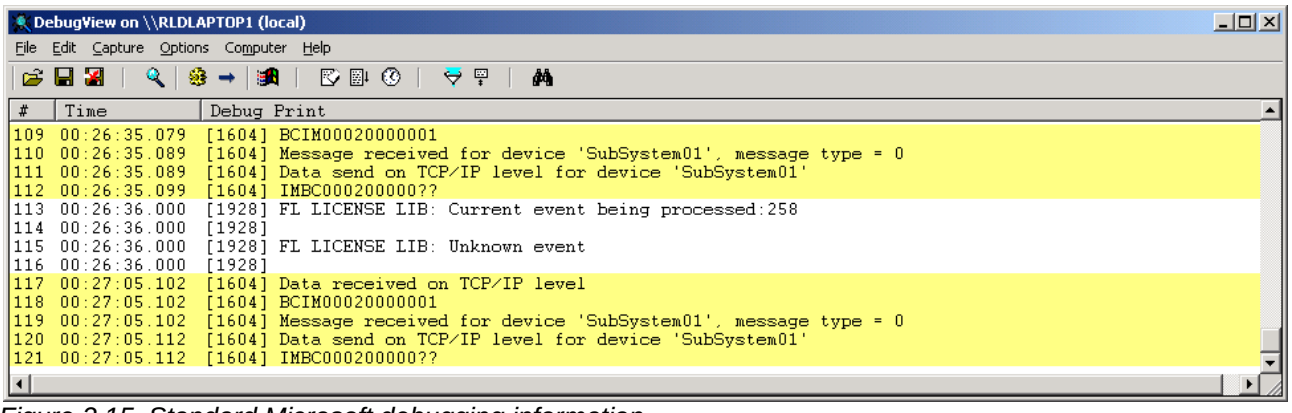


Figure 2.15. Standard Microsoft debugging information.

By setting the key Log2File to a value not equal to zero the debug information is logged to a file in the application directory: {FLAPP}\{FLNAME}\{FLDOMAIN}\log\im_bas.log. For default environment settings this

name will be: {FLAPP}]flapp1\shared\log\im_bas.log, where {FLAPP1} is your application directory. Every time the driver is started the log file is cleared, during run-time you have no control over the size of the log-file! Additional to logging to a file it is possible to log to the Microsoft event viewer, all you have to do is set the key Log2EventViewer to a value not equal to zero. Logging to the log-file and the event viewer can be active together, or you can choose only one (or none) of the logging methods.

Microsoft supports debugging of applications with a 'DebugViewer', an application that is capable of showing, filtering and saving debug information. In your shipment is included a debug viewer application. In order for this application to show the debug information, the driver should generate the Microsoft compatible debug information, to do so turn the option 'Log2DebugViewer'. The debug viewer included supports remote debugging, use any system in your network to view, filter or save the debug information form the driver, the only thing you have to do is connect to the server where the driver is running.

2.6.2 Section: [Task]

In this section is defined how the driver task interacts with the FactoryLink real-time database, and how multi-threading is configured for optimal speed, or real-time database access. A normal task uses one name, the task name (see previous paragraph) to get access to the real-time database.

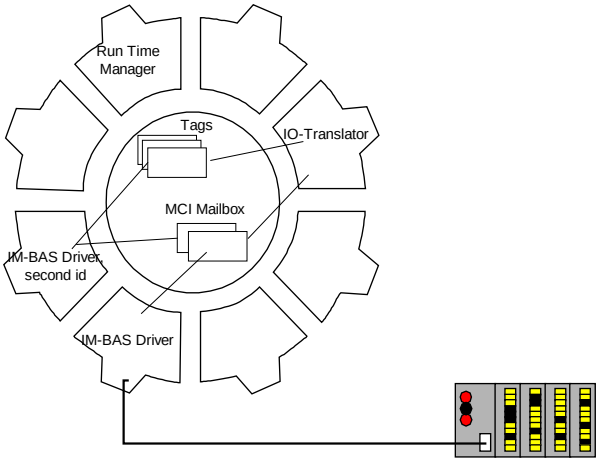


Figure 2.16. Second task-id for driver.

For optimal speed a task can use two task-id's instead of one, one task id will be used to receive commands from the I/O translator mailbox('s) and the second id will be used to reply to the I/O translator. Using two id's allows a task to simultaneously read and write from/to the real-database, with one id these actions are carried one after another. Using two id's is set with the key 'Driver2Ids' to a value not equal to zero.
For the second task-id there must be an empty entry in the system configuration table, not all 30 task entries should be filled in. During start-up the driver will automatically acquire two task-id's.
A normal task, with one task-id, can operate in a polled or event mode, default the event mode is used. However if desired, the polled mode is activated with the key 'Polling' set to a non-zero value. The poll-time is set with the key 'PollTime' to a time in milliseconds; this is the time between successive polls of the real-time database for commands or instructions.
Note: these settings are not used for the FlNTsys module.

2.6.3 Section: [DiskSpace]

UseMB: For diskspace calculations the result will in Mb. This specification can be superseded by UseGB. The value can zero or non-zero. A value of zero indicates that diskspace results will be returned in bytes. A non-zero value will give diskspace result in Mb. In case the setting UseGB is defined with a non-zero value this setting is actually ignored by the module.
UseGB: For diskspace calculations the result will in Gb. This specification can be superseded by UseGB. The value can zero or non-zero. A value of zero indicates that diskspace results will be returned in bytes. A non-zero value will give diskspace result in Gb.

2.6.4 Section: [TimeZone]

TimeZoneName: Setting defines the logical name of the local timezone. This the timezone used for performing timezone dependant calculations or conversions. The value is always a string in the English language. The default timezone is: "W. Europe Standard Time".

3 Configuration tables

The configuration tables for the NT system utilities can be accessed with the FactoryLink Configuration Explorer. For general information about entering data in FactoryLink configuration tables refer to the FactoryLink Manual. In the Configuration Explorer select 'Other tasks', one table entry 'NT System utilities' appears. Opening this option gives you the three main configuration parts of the task: 'Enable/Disable Keystrokes', 'Log to the Event Viewer' and 'System Settings'. These three configuration panels allow entering configuration information for the three different functionality's of the FactoryLink application: enabling or disabling keystrokes, logging messages to the event viewer and retrieving or setting system parameters.

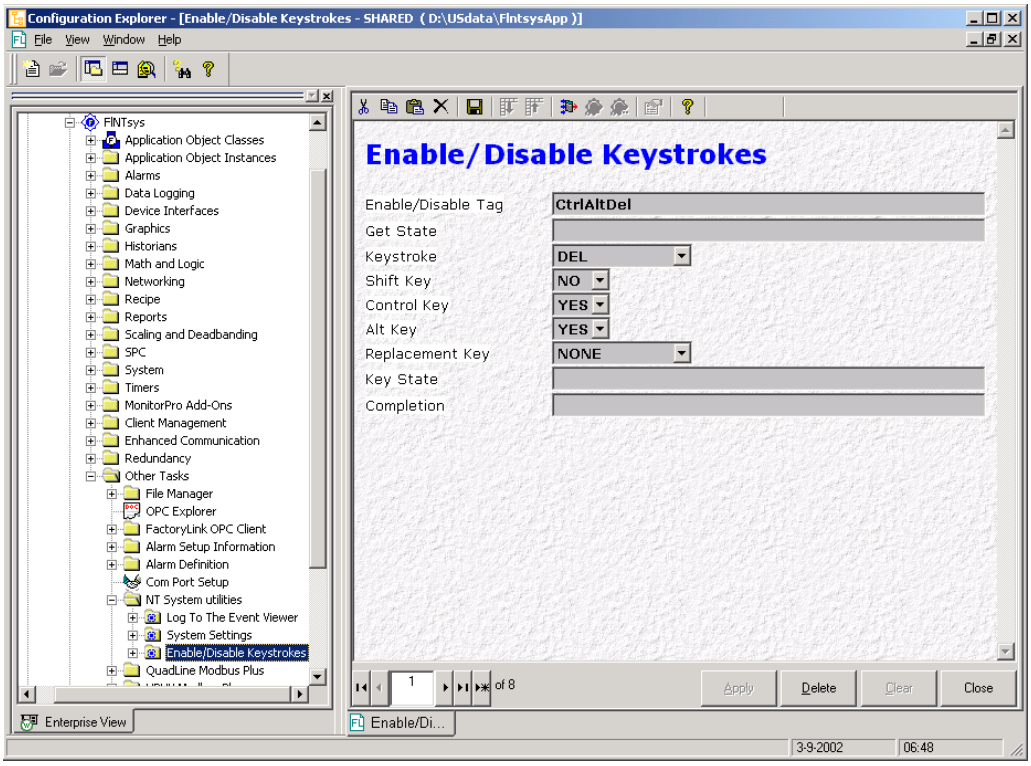


Figure 3.1. FactoryLink configuration Explorer.

3.1 Enable/Disable keystrokes

The 'Enable/Disable Keystrokes' panel is used to filter keys form the keyboard. Filtering is made active with a digital tag, filtering is set active with a digital value of ON (value is '1'). In case of active filtering the defined keystroke is removed from the keyboard buffer, this means no application will receive this keystroke. A keystroke can be combined with any combination of the alt, shift or control keys. For windows NT care should be taken with the combination 'Ctlr + Alt + Del', if this key combination is disabled, the workstation should not be locked, because the same keystroke is used to unlock a workstation and allow the user to enter a username + password to return to his desktop.
In fact the same applies to a screensaver, a screensaver can be used with and without a password, when using a password the screensaver ends only if the keys 'Ctrl + Alt + Del' can be pressed!

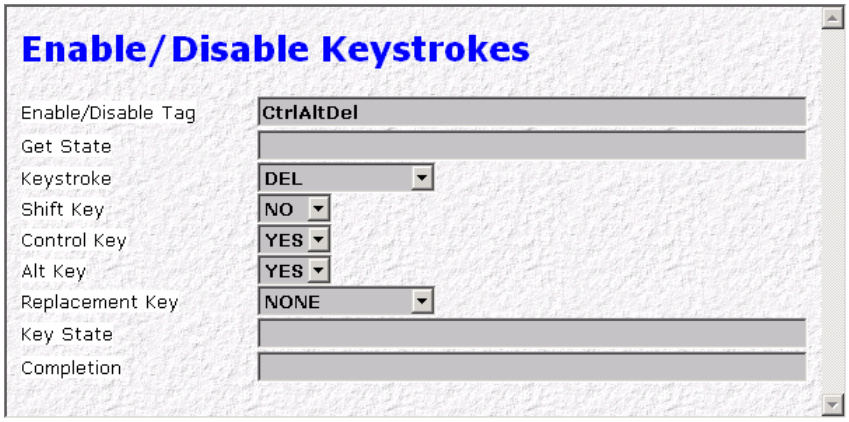


Figure 3.2. Configuration panel for keystroke filtering.

Every definition in the configuration panel completely defines the filtering behaviour for a particular keystroke. All the fields are discussed in the same order as they are presented in the configuration panel.

Enable/Disable Tag

A Digital tag which enables or disables the filtering (or removal of the keyboard buffer) of the defined keystroke (the keystroke is defined in the next field). The value of the tag defines the filtering type, value OFF (tag value is '0') enables the keystroke, the keystroke is not removed form the keyboard buffer. A value of ON (value is '1') disables the keystroke, the keystroke is upon detection (that is entrance in the keyboard buffer) removed from the keyboard buffer.

entry ✓ Required
entry type Tagname, tag type: DIGITAL

Get State

A Digital tag used as a trigger to retrieve the current state of the keystroke: enabled or disabled. The current state is returned as a tag value, this tag is defined in the field 'Key State'. The current state is only retrieved when the trigger is activated: the value of the digital tag is 'ON', and the change bits of the tag is set (indicating a value change from 'OFF' to 'ON' or a 'forced' write).

entry Optional
entry type Tagname, tag type: DIGITAL

Keystroke

The keystroke removed from the keyboard buffer if the value of the digital tag in the 'Enable/Disable Tag' field has the value 'ON'. A selection list is provided to select the keystroke, the keystroke to filter must be in

this selection list. If you may need more keystrokes then the ones defined, please feel free to contact us, the list can easily be extended! Standard the following keystrokes are provided:

Selection	Keystroke name
DEL	Delete key
ESC	Escape key
TAB	Tab key
FLYWINL	Left windows key
FLYWINR	Right windows key
POPUPMENU	
F1	Function key 1
F2	Function key 2
F3	Function key 3
F4	Function key 4
F5	Function key 5
F6	Function key 6
F7	Function key 7
F8	Function key 8
F9	Function key 9
F10	Function key 10
F11	Function key 11
F12	Function key 12
CAPS_LOCK	Caps Lock key
PRINTSCREEN	Print screen button
ENTER	Enter key
BACKSPACE	Back space key
SCROLL_LOCK	Scroll lock key
NUM_LOCK	Numeric Lock key
INSERT	Insert key
HOME	Home key
PAGE_UP	Page up key
DELETE	Delete key
END	End key
PAGE_DOWN	Page down key
UP	Up arrow key
LEFT	Left arow key
RIGHT	Right arrow key
DOWN	Down arrow key

entry ✓ Required
entry type Keyword from a selection list.
default DEL

Shift key

The keystroke to filter can be combined with the shift, ctrl or alt key and any combination of these keys. This field defines if the shift key needs to be pressed together with the keystroke for removing the keystroke from the keyboard buffer. A value of 'YES' sets the combination to: Shift + Keystroke.

entry ✓ Required
entry type Keyword form a selection list: YES, NO, Y, N
default NO

Control key

The keystroke to filter can be combined with the shift, ctrl or alt key and any combination of these keys. This field defines if the control key needs to be pressed together with the keystroke for removing the keystroke from the keyboard buffer. A value of 'YES' sets the combination to: Ctrl + Keystroke.

entry ✓ Required

entry type Keyword form a selection list: YES, NO, Y, N
default NO

Alt key

The keystroke to filter can be combined with the shift, ctrl or alt key and any combination of these keys. This field defines if the alt key needs to be pressed together with the keystroke for removing the keystroke from the keyboard buffer. A value of 'YES' sets the combination to: Alt + Keystroke.

entry ✓ Required
entry type Keyword form a selection list: YES, NO, Y, N
default NO

Replacement key

Instead of removing the keystroke from the keyboard buffer the key can be replaced with another key. For example the printsrceen key: this keystroke is normally handled by NT, and is not received by the FactoryLink graphics task. Replace this keystroke with 'F12' and the graphics task gets the 'F12' key any time the printscreen button is pressed. If you define a graphics 'print screen' action for 'F12' key, a print of the current screen will be generated every time the printscreen button is pressed.

entry ✓ Required
entry type Keyword from a selection list.
default NONE

Selection	Keystroke name
NONE	No replacement key active
DEL	Delete key
ESC	Escape key
TAB	Tab key
FLYWINL	Left windows key
FLYWINR	Right windows key
POPUPMENU	
F1	Function key 1
F2	Function key 2
F3	Function key 3
F4	Function key 4
F5	Function key 5
F6	Function key 6
F7	Function key 7
F8	Function key 8
F9	Function key 9
F10	Function key 10
F11	Function key 11
F12	Function key 12
CAPS_LOCK	Caps Lock key
PRINTSCREEN	Print screen button
ENTER	Enter key
BACKSPACE	Back space key
SCROLL_LOCK	Scroll lock key
NUM_LOCK	Numeric Lock key
INSERT	Insert key
HOME	Home key
PAGE_UP	Page up key
DELETE	Delete key
END	End key
PAGE_DOWN	Page down key
UP	Up arrow key
LEFT	Left arow key
RIGHT	Right arrow key
DOWN	Down arrow key

Key State

If defined this tag has the current filtering state of a keystroke, that is after setting the filtering state with the 'Enable/Disable Tag' or after activating the trigger 'Get State'. The tag value can be 'ON', indicating a disabled keystroke, or 'OFF', indicating an enabled keystroke.

entry	Optional
entry type	Tagname, tag type: DIGITAL, ANALOG

Completion

After setting or retrieving the filtering state of a keystroke, and all output tag's are written, the tag defined here is forced to 'ON'. This tag can be used to signal the completion of setting/retrieving the filtering state of a keystroke.

entry	Optional
entry type	Tagname, tag type: DIGITAL, ANALOG

3.2 System settings

The 'System Settings' panel is used to change system settings or perform system actions. There are different groups of actions or settings. The first group is for shutting down the system, depending on the configuration of your system the system can be shutdown, powered off or logged off. The success of every one of these actions depends on the settings for security, if a logged on user does not have the privilege to shutdown the system, executing the action will not result in shutting down the workstation. Prior to shutting down (or logging off) the complete FactoryLink application is stopped.
The second group deals with disk space, both the numbers of kBytes free disk space and used disk space can be requested. The result will be available as a numeric tag value. In the third group settings for the screen saver can be requested and set. The inactivity time before a screensaver gets active can be queried and changed. The setting if a screensaver should appear after some time of inactivity can be read and changed.

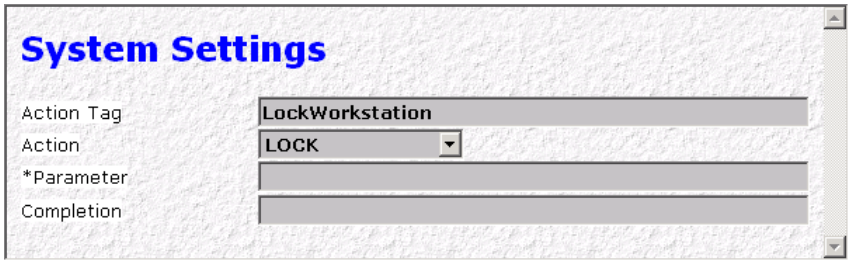


Figure 3.3. Configuration panel for system settings and actions.

Every definition in the configuration panel completely defines a setting or action. All the fields are discussed in the same order as they are presented in the configuration panel.

Action Tag

Digital tag associated with the action defined in the next field, the tag is used as a trigger tag to start the action. For a digital trigger tag this means the value and the change flag for the tag must be both 'ON' for executing the action. The actions are described in the next field.

entry ✓ Required
entry type Tagname, tag type: DIGITAL

Action

The action to perform if the trigger defined in the previous field is active. A selection list is provided to select the action, the action must be in this selection list. If you may need more or different actions then the ones defined, please feel free to contact us, on request the list can be extended. In the table below the possible actions are listed, together with a description. For some action the next field '*Parameter' is essential, for the action this field is interpreted as a parameter. It can be input or an output parameter for an action.

Action	Description	Parameter
LOGOFF	Log the current user off, FactoryLink application is stopped prior to logging off.	*not used*
POWEROFF	Log the current user off, shutdown and power off the workstation. The FactoryLink application is stopped prior to logging off. Note that your workstation should have powermanagement installed properly for correct functioning of this action.	*not used*
REBOOT	Log the current user off, shutdown the workstation and reboot. The FactoryLink application is stopped prior to logging off.	*not used*
SHUTDOWN	Log the current user off, shutdown the	*not used*

Action	Description	Parameter
	workstation. The FactoryLink application is stopped prior to logging off.	
LOCK	Lock the workstation.	*not used*
SCREENSAVER	Activate the current installed screen saver. Don't move mouse of hit the keyboard after activation, otherwise the screensaver will end… Tag triggers the activation of the screensaver, triggering is on the change bit, any value allowed, except for a digital tag which also needs a value of 1 (ON).	*not used*
SET_SCRNSAVER	Boolean setting for the screensaver: the parameter value determines if a screensaver after a certain time of inactivity is displayed. The value is set for the workstation.	Digital value 0, 1 to prevent or allow a screensaver after some time of inactivity.
GET_SCRNSAVER	Boolean setting for the screensaver: the parameter value determines if a screensaver after a certain time of inactivity is displayed. The value is queried from the workstation.	Tag tag has the actual setting for the screensaver activation: 0 means screensaver will never be activated, 1 means that the screensaver will get active after a certain time of inactivity.
SET_SCRNTIME	The inactivity time in seconds before a screensaver gets active. Note: the activation of a screensaver should be allowed! The value is set for the workstation.	Decimal value representing the inactivity time in seconds, ranging from 1 and up.
GET_SCRNTIME	The inactivity time in seconds before a screensaver gets active. Note: the activation of a screensaver should be allowed! The value is queried from the workstation.	Tag which has the actual value of the inactivity time in seconds after execution of the action.
SET_PSWSAVER	A screensaver can be password protected, with this action the current password protection is set.	Digital value 0, 1 to set or remove password enabling of the screensaver.
GET_PSWSAVER	A screensaver can be password protected, with this action the current password protection is retrieved.	Tag that has the actual value of the password enabling of the screensaver, 1 means that the screensaver is password protected.
SET_LOCK	After pressing Control+Alt+Delete, a security screen will pop-up, on this screen there is a button to lock the workstation, this button can be enabled or disabled. With this action the button state is set to enabled or disabled.	Digital value 0, 1 to enable or disable the 'lock' button on the security screen.
GET_LOCK	After pressing Control+Alt+Delete, a security screen will pop-up, on this screen there is a button to lock the workstation, this button can be enabled or disabled. With this action the actual button state is retrieved.	Tag that has the actual value of the state of the 'lock' button on the security screen, 1 means the button is disabled.
SET_TASKM	After pressing Control+Alt+Delete, a security screen will pop-up, on this screen there is a button to start the task manager, this button can be enabled or disabled. With this action the button state is set to enabled or disabled.	Digital value 0, 1 to enable or disable the 'task manager' button on the security screen.
GET_TASKM	After pressing Control+Alt+Delete, a security screen will pop-up, on this screen there is a button to start the task manager, this button can be enabled or disabled. With this action the actual button state is retrieved.	Tag that has the actual value of the state of the 'task manager' button on the security screen, 1 means the button is disabled.
GET_TIMEBIAS	Retrieve the actual time bias, time difference between UTC and local time, in minutes. The value can be positive or negative, depending on the actual time zone and daylight saving time, it is the difference between the local time (as set on the workstation) and the UTC time.	Tag that has the actual value of the time bias in minutes.
GET_DAYLIGHT	Retrieve the actual value for daylight saving time in seconds. This time depends on the settings on	Tag that has the actual value of the offset caused by daylight

Action	Description	Parameter
	the workstation , if daylight saving time is disabled the value will always be zero! There is no difference between local time and UTC time. This value will give the offset between local time and UTC time, that is the offset caused by daylight saving time.	saving time.
SET_UTCTIME	Set the actual UTC time on the parameter tag (user is responsible for correct format of tag), The UTC time is the local time corrected with the actual daylight saving offset. The local time is assumed to be the value of the trigger tag. Using the tag 'SECTIME' as the trigger and eg 'UTCTIME' as the parameter tag will correct the value of 'SECTIME' with daylight saving time offset and place the result in 'UTCTIME'.	Tag that has the actual UTC time in seconds corrected with the offset caused by daylight saving time.
DAYLIGHT_SAVING	Set the actual of the daylight saving time offset on the parameter tag (user is responsible for correct format of tag). Timezone used is specified in the ini-file flntsys.ini.	Tag that has the actual daylight saving time offset in seconds.
SET_ DAYLIGHT	Set the actual UTC time, corrected with daylight saving offset on the parameter tag (user is responsible for correct format of tag), The UTC time is the local time corrected with the actual daylight saving offset. The local time is assumed to be the value of the trigger tag. Using the tag 'SECTIME' as the trigger and eg 'UTCTIME' as the parameter tag will correct the value of 'SECTIME' with daylight saving time offset and place the result in 'UTCTIME'. Timezone used is specified in the ini-file flntsys.ini.	Tag that has the actual UTC time in seconds corrected with the offset caused by daylight saving time.
MSG_ISO8601	Conversion of time value used in FactoryLink ("SECTIME") to a string. If a value of less then 190000 is given on the trigger tag, the output will be a time string only, the date is left off. Conversion format: `year-mo-dyThr:mi:sc`	Tag that has the actual datetime string.
FREE_?	Requests the amount of free disk space in kBytes of drive ?.	Tag which has the actual amount of free disk space.
USED_?	Requests the amount of used disk space in kBytes of drive ?.	Tag which has the actual amount of used disk space.

entry ✓ Required
entry type Keyword from a selection list.
default SHUTDOWN

*Parameter

Parameter value associated with the action defined in the previous field. The entry can be a fixed parameter (or numerical value) or a tag name. In case of a tag the application uses the tag value as the parameter value. This applies only to parameters, which are 'inputs', for the action, see the description of the actions in the table above. For output parameters only a tag name is useful, the application will set the tag value to the result of the action.

entry Optional
entry type Tagname, tag type: DIGITAL, ANALOG, LONG ANALOG, FLOAT

Completion

After executing an action, and if needed writing a value to the parameter tag, the tag defined here is forced to ‘ON’. This tag can be used to signal the completion of an action.

entry	Optional
entry type	Tagname, tag type: DIGITAL, ANALOG

3.3 Event Logging

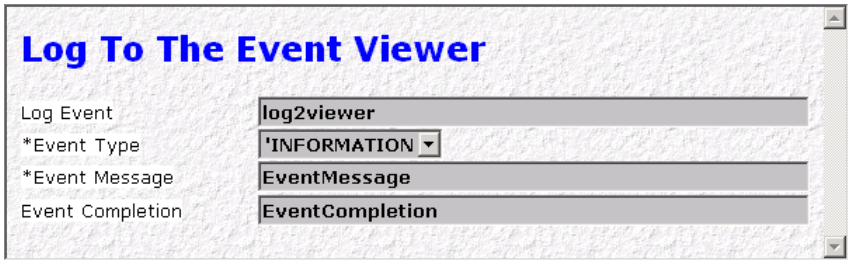


Figure 3.4. Configuration panel for event logging.

Every row in the configuration panel completely defines an event logging mechanism. All the fields are discussed in the same order as they are presented in the configuration panel.

Log Event

Digital tag used as a trigger to start the logging action, the type of event and the message logged are defined with the next two fields. For a digital trigger tag this means the value and the change flag for the tag must be both 'ON' to start the logging action.

entry ✓ Required
entry type Tagname, tag type: DIGITAL

*Event Type

Type of the event logged is defined with this field, the entry types allowed are an alphanumeric string or an analog tag. Three different types are possible, every one of them has its own icon in the event viewer: Information, error or warning message. See the figure and table below for the appearance in the event viewer and the associated icons for every type. The alphanumeric allowed entries in the field correspond with a decimal value of 0 ..2, this value is the equivalent of the selections 'INFORMATION', 'WARNING' and 'ERROR'. And these are the only values accepted if an analog tag name is entered in the field, values less then zero are (internally in the task) set to zero, and values greater then two are (internally in the task) set to 2. Don't forget to start the selection text with an apostrophe ', only with this sign the Configuration Explorer will interpret the entry as a (static) selection, otherwise the entry is assumed to be a tag name.

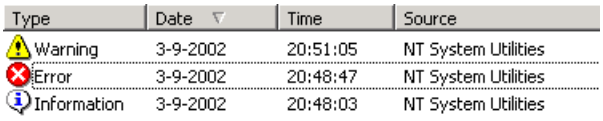


Figure 3.5. Event types in the event viewer.

Selection	Decimal	Icon
WARNING	0	
INFORMATION	1	
ERROR	2	

entry ✓ Required
entry type Tagname, tag type: ANALOG
Keyword from a selection list: INFORMATION, ERROR, WARNING
default INFORMATION

*Event Message

The message logged to the event viewer can be static or dynamic, dynamic means the entry in this field is a message tag, the text in the tag is used as the log-message. For a static configuration the message text is

configured in the panel, don't forget to start the text with an apostrophe '. Only with this sign the Configuration Explorer will interpret the entry as static text, otherwise the entry is assumed to be a tag name.

entry	Optional
entry type	Tagname, tag type: MESSAGE Alphanumeric string of up to 250 characters

Completion

After the logging is completed the tag defined here is forced to 'ON'. This tag can be used to signal the completion of a logging action.

entry	Optional
entry type	Tagname, tag type: DIGITAL, ANALOG

4 ActiveX Control 'Keyboard Filter'

If you have installed the 'client' part of the FlNTsys application, a new ActiveX control is now registered on your system: Keyboard Filter. Together with this control the keyboard filter driver is installed on the system, together they can filter keystrokes from the keyboard on a 'client' system. That is a system without an active FactoryLink server! The control is easily integrated in Client Builder, and can be used in combination with the FlNTsys FactoryLink server task.
To enable keystroke filtering from within a Client Builder application one must install the ActiveX control from the installation media, and next select the control as a favourite in the Client Builder environment, see the figure below.

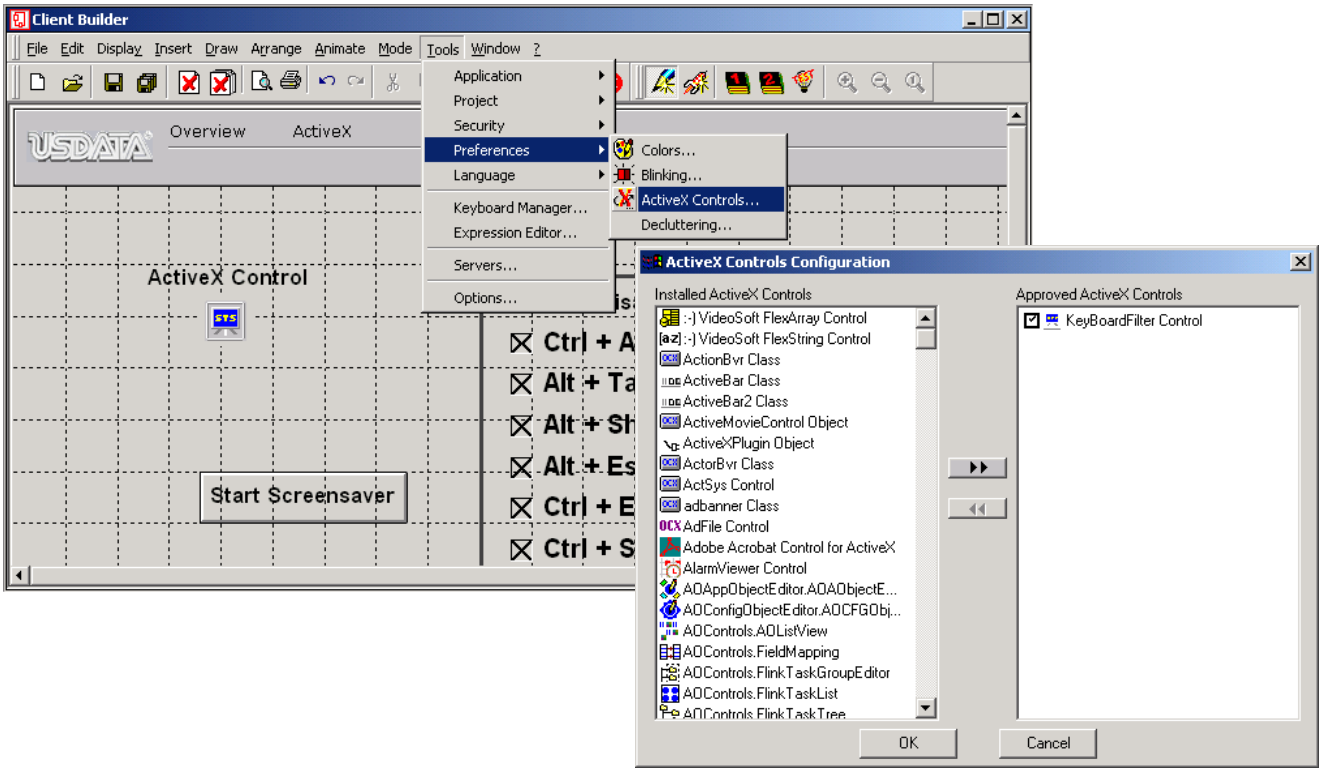


Figure 4.1. Setting the 'KeyboardFilter Control' as a favourite control in Client Builder.

If the control is added as an approved control, you can insert the control on a page, see the figure below.

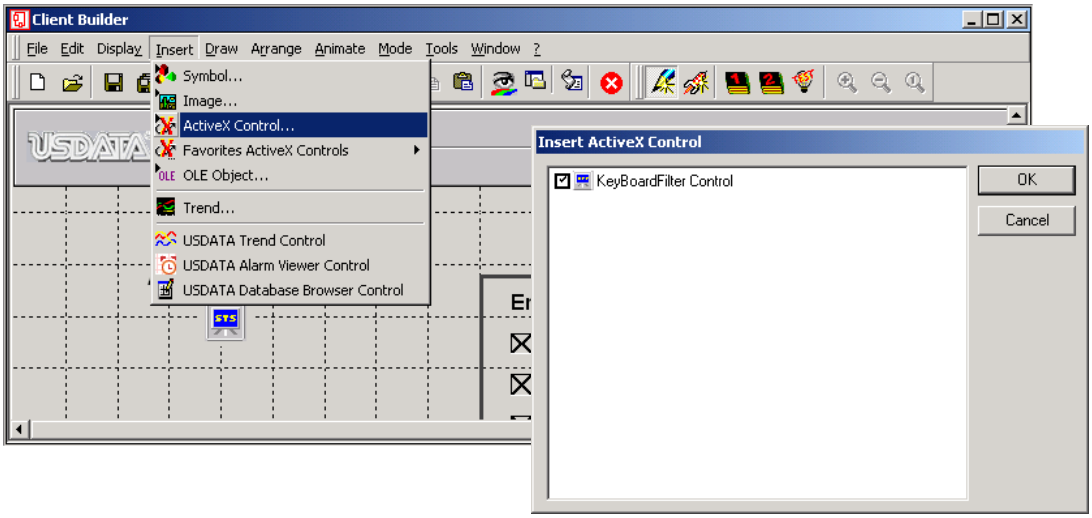


Figure 4.2. Inserting th e'KeyboardFilter Control' on a page.

After the insertion of the control on a page, you can set one property: the control can be visible or not. A visible control looks just like the icon on the page in the figure above. An invisible control is present on a page, but invisible for the user (and developer).

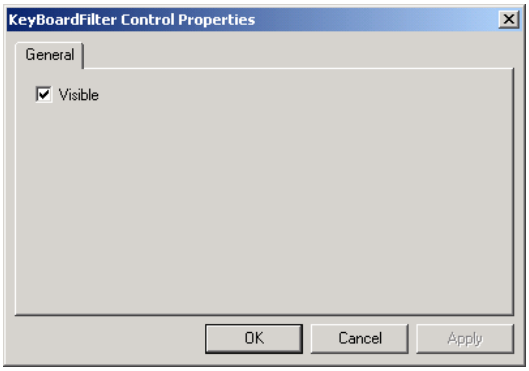


Figure 4.3. Properties of the ActiveX control.

To disable or enable keystrokes the appropriate methods of the control should be invoked. The control has seceral methods and properties which you can invoke, retrieve or set. An example of invoking a method is given in the sample script below.

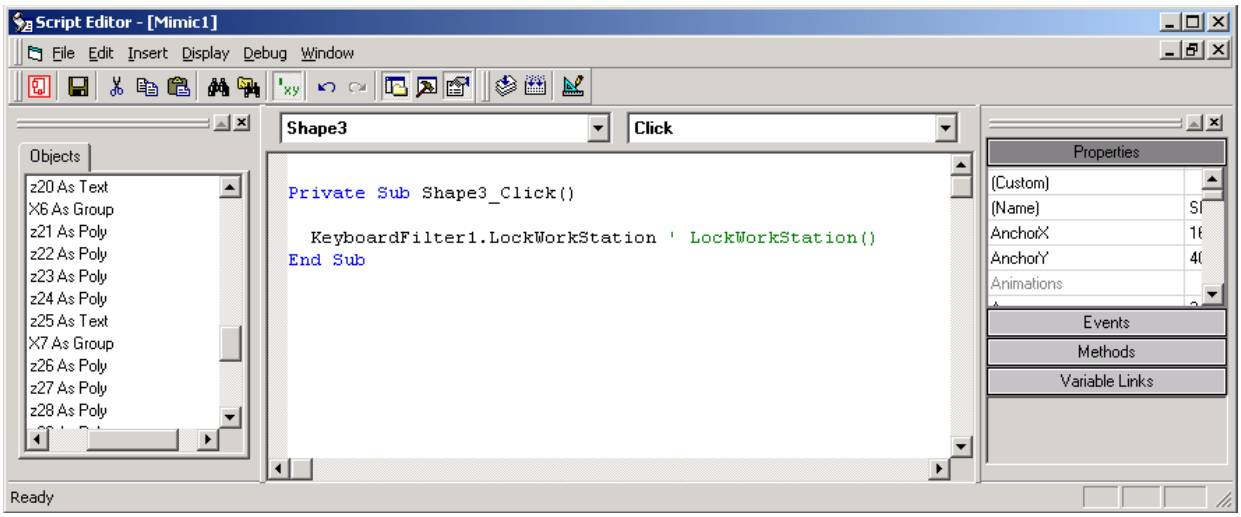


Figure 4.4. A script, invoking a method of the 'KeyboardFiltering Control'.

4.1 Properties

There is only one property of the control that can be read and written to, which is 'Visible'. This is the same property which is displayed as an option on the configuration dialog of the control. The other properties are read-only, and are standard properties of an ActiveX object.

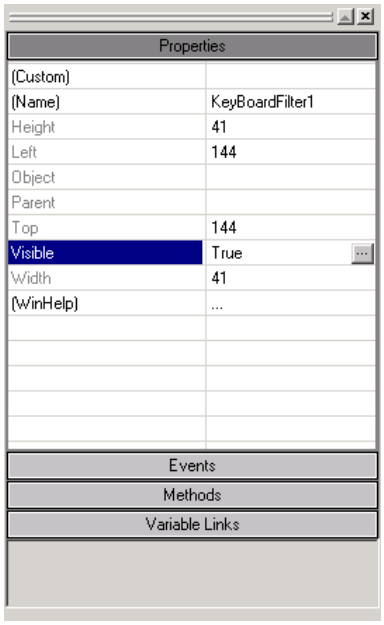


Figure 4.5. 'KeyboardFiltering Control' properties.

4.2 Methods

The methods of the 'KeyboardFilter Control' are functions to disable, enable keystrokes, lock the workstation or invoke the screensaver.

Method	Function prototype
AboutBox	Sub AboutBox ( ) Show the about box of the control, the dialog is dismissed with the 'Ok' button.
DisableKey	Sub DisableKey (ByVal ScanCode As Integer, ByVal Shift As Boolean, ByVal Alt As Boolean, ByVal Control As Boolean) Disable the specified keystroke (Scancode), and define the exact key combination with Control, Alt and/or Shift.
DisableLockWorkstation	Sub DisableLockWorkstation (ByVal Disable As Boolean) Disable the 'Lock Computer' button on the security page, that is invoked with the key combination: Control+Alt+Delete.
DisableTaskManager	Sub DisableTaskManager (ByVal Disable As Boolean) Disable the 'Task Manager' button on the security page, that is invoked with the key combination: Control+Alt+Delete.
EnableAllKeys	Sub EnableAllKeys ( ) Just enable all keystrokes.
EnableKey	Sub EnableKey (ByVal ScanCode As Integer, ByVal Shift As Boolean, ByVal Alt As Boolean, ByVal Control As Boolean) Enable the specified keystroke (Scancode), and define the exact key combination with Control, Alt and/or Shift.

IsKeyDisabled	Sub IsKeyDisabled (ByVal ScanCode As Integer, ByVal Shift As Boolean, ByVal Alt As Boolean, ByVal Control As Boolean) Retrieve the actual state (enabled = 0, disabled <> 0) of the specified keystroke (ScanCode), the combination of Control, Alt and Shift is also defined.
LockWorkStation	Sub LockWorkStation ( ) Lock the workstation. If the key combination 'Control+Alt+Delete' is disabled, the user can never unlock the system!
ReplaceKey	Sub ReplaceKey (ByVal ScanCode As Integer, ByVal Shift As Boolean, ByVal Alt As Boolean, ByVal Control As Boolean, ByVal ReplaceCode As Integer) Replace the specified keystroke (Scancode)with the one given in 'ReplaceCode', and define the exact key combination with Control, Alt and/or Shift. The Control, Alt and Shift setting also apply for the replacement keystroke.
SetScreenSaverSecure	Sub SetScreenSaverSecure (ByVal Secure As Boolean) Set or remove the password protection from the screensaver, specify a boolean value true to secure the screensaver.
StartScreenSaver	Sub StartScreenSaver ( ) Start the (default) screensaver on the system, if there is no screen saver activated the function will fail.

You can find the most used scancodes in the list below.

```
'Key          Hex code   Decimal
'DEL          53         83
'ESC           1         1
'TAB           F         15
'FLYWINL      5B         91
'FLYWINR      5C         92
'POPUPMENU    5D         93
'F1           3B         59
'F2           3C         60
'F3           3D         61
'F4           3E         62
'F5           3F         63
'F6           40         64
'F7           41         65
'F8           42         66
'F9           43         67
'F10          44         68
'F11          57         87
'F12          58         88
'CAPS_LOCK    3A         58
'PRINTSCREEN  37         55
'ENTER        1C         28
'BACKSPACE     E         14
'SCROLL_LOCK  46         70
'NUM_LOCK     45         69
'INSERT       52         82
'HOME         47         71
'PAGE_UP      49         73
'DELETE       53         83
'End          4F         79
'PAGE_DOWN    51         81
'UP           48         72
'Left         50         80
'Right        4B         75
'DOWN         4D         77
```

5 Error messages

If an error condition is present in the application, an appropriate message is send to the FactoryLink kernel. The message can be seen in the 'run-time manger' screens which are part of the starter application. In the standard screens the error message will appear to the right of the 'FLNTSYS' task. Errors are marked by a red button, if an error occurred but the task can continue. Severe errors cause a FactoryLink task to stop, these errors are marked with an error description and a grey button (same as an inactive task).

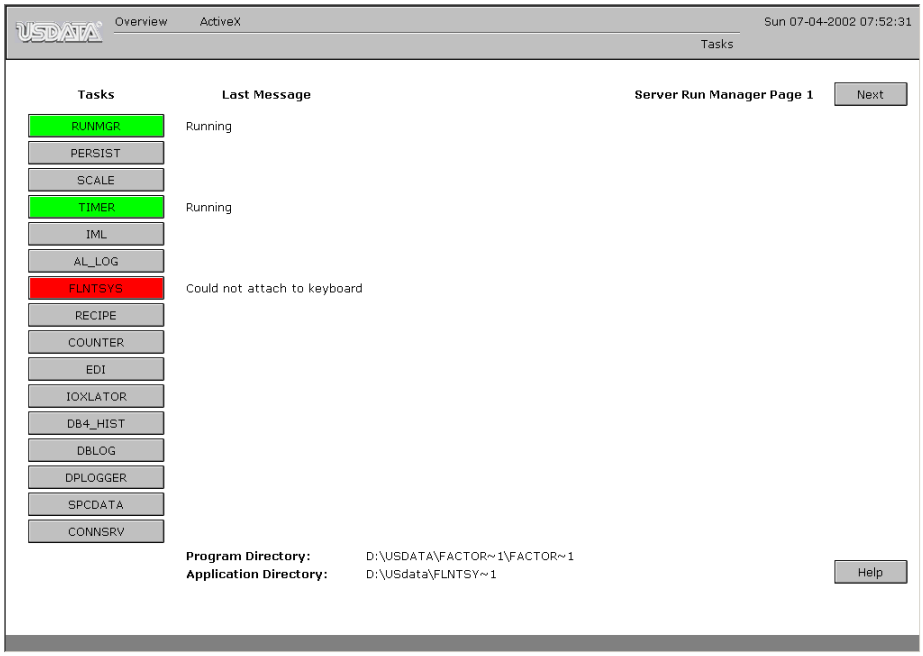


Figure 4.1. Run-time manager screen.

FLNTSYS **Could not attach to keyboard**

The application was unable to find the keyboard filter device, keyfiltering is not possible.

Possible solutions: Make sure the keyboard filter driver is installed and running, to start a device in NT4 go to the control panel → devices → Keyboard filter, and try to start the device if it is not running. For Windows2000 open the control panel → Administrative tools → Computer management. In the application computer management select in the tree on the left side: Device manager → Non-plug and play drivers → Keyboard Filter. If the entry 'non-plug and play drivers' is not visible, activate it with the menu command 'View → Show hidden devices'. For starting a device one needs administrator rights. If the driver is not in the list (re)install the application.

FLNTSYS **System tag for exiting FactoryLink is not found**

The application was unable to find the system tag to shutdown the FactoryLink application, shutting down FactoryLink is now impossible. Possible causes are a corrupted database (see 'Could not access '%s") or missing of the system tag, the system tags can be removed!

Possible solutions: Make sure the system tags are all present.

FLNTSYS **Changing system information failed**

Changing the system settings for screensaver, retrieving disk space, shutting down the system failed. Most likely cause is not enough rights to change or even request a system setting.

Possible solutions: increase the rights of the user, or log on as a different user with more rights.

FLNTSYS **Reading system information failed**

Reading the system settings for screensaver, retrieving disk space, shutting down the system failed. Most likely cause is not enough rights to change or even request a system setting.

Possible solutions: increase the rights of the user, or log on as a different user with more rights.

FLNTSYS **Could not access '%s'**

The application was unable to access the specified tag (on position %s), most likely cause is a corrupted application database and/or CT-file.

Possible solutions: The normal advised action to repair a damaged application database, try generating the CT-file by hand. One point to check is if the application is correctly installed in your FactoryLink system directory. There must be entry with FLNTsys in both the {FLINK}/ac/titles file and the {FLINK}/ctgen/ctlist file. Is either one of them is missing try to reinstall the application.

FLNTSYS **CT-file not found**

The application was unable to access the task specific CT-file, the file was not found in the application directory.

Possible solutions: Make sure the FlNTsys task has a configuration in the Configuration Explorer, if so please check if this is the same application as you started.
Check if there is a configuration file 'flntsys.ct present in the application, directory: {FLAPP}\shared\ct. If there is no such file present, see if FactoryLink is instructed to generate this file. In the ASCII-file '{FLINK}\ctgen\ctlist' there must an entry: **flntsys: ntsys_k ntsys_s ntsys_m**.

FLNTSYS **No triggers found in CT-file**

The FlNTsys task found a configuration file 'flntsys.ct' in the application directory, however there are no triggers defined for the task.

Possible solutions: Make sure the FlNTsys task has a valid configuration in the Configuration Explorer, if so please check if this is the same application as you started.
Check if there is a configuration file 'flntsys.ct present in the application, directory: {FLAPP}\shared\ct. If there is no such file present, see if FactoryLink is instructed to generate this file. In the ASCII-file '{FLINK}\ctgen\ctlist' there must an entry: **flntsys: ntsys_k ntsys_s ntsys_m.**
If a configuration file (CT-file) present, please generate a list file of the configuration, before contacting the support desk. The listing file is generated with a utility called "ctlist.exe", check first if the program is present in the {FLINK}\bin directory, if so open a command prompt and go to the application directory. In the application directory is heck if there is a configuration file 'flntsys.ct present in the application, directory: {FLAPP}\shared\ct. In the application directory go to the subdirectory: shared\ct, from there run the command: "ctlist flntsys.ct". The output can be redirected to a file, e.g. appending: "> d:\flntsys.txt" to the commandline. Contact the support desk for more help.

6 License Agreement

The use of this program is subject to the following terms and conditions.

Title To The Licensed Software
Title to the licensed software is NOT transferred or sold to the end user. The end user is granted a non-exclusive license to use the software on a SINGLE computer or computer work station. EACH computer or computer workstation must have its own licensed copy of the software.

Copyright Protection
This software is copyrighted material. It is protected by the copyright laws of the Netherlands, and other proprietary rights of RLD Automation. You may not make any changes or modifications to the program or the manual. You may not decompile, disassemble, or otherwise reverse-engineer the software in any way.

Limited Warranty
RLD Automation does not warrant that the licensed software will meet your requirements or that the operation of the software will be uninterrupted or error free. The warranty does not cover any media or documentation which has been subjected to damage or abuse by you or others. The software warranty does not cover any copy of the licensed software which has been altered or changed in any way. In other words there is no warranty either implied or expressed.

ANY IMPLIED WARRANTIES INCLUDING ANY WARRANTIES OF MERCHANTABILITY OR FITNESS OR A PARTICULAR PURPOSE ARE LIMITED TO THE TERM OF THE EXPRESS WARRANTIES. Some states do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you.

Other Warranties
The warranties set forth above are in lieu of any and all other express or implied warranties, whether oral, written, or implied, and the remedies set forth above are the sole and exclusive remedies.

Limitation Of Liability
RLD Automation is not responsible nor liable in anyway for any problems or damage caused by the licensed software that may result from using the licensed software. This includes, but is not limited to, computer hardware, computer software, operating systems, and any computer or computing accessories. End user agrees to hold RLD Automation harmless for any problems arising from the use of the software.
RLD Automation SHALL NOT IN ANY CASE BE LIABLE FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, INDIRECT OR OTHER SIMILAR DAMAGES ARISING FROM ANY BREACH OF THESE WARRANTIES EVEN IF RLD Automation OR ITS AGENTS OR DISTRIBUTORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

In no case shall RLD Automation's liability exceed the license fees paid for the right to use the licensed software.

The above constitutes the license agreement for this program. It supersedes any and all previous license agreements.