# Table of Contents

1 Introduction ........................................................................................................................................ 3
  1.1 About this Document .................................................................................................................... 3
  1.2 List of Revisions ............................................................................................................................ 3
  1.3 Overview ....................................................................................................................................... 4
  1.4 Requirement ................................................................................................................................. 5
  1.5 Supported Hardware ...................................................................................................................... 5
  1.6 Features ....................................................................................................................................... 5
  1.7 Limitations .................................................................................................................................... 5
  1.8 CD Contents ................................................................................................................................ 6
  1.9 Terms and Abbreviations .............................................................................................................. 7
  1.10 References to Documents ......................................................................................................... 7

2 Installation ......................................................................................................................................... 8
  2.1 Manual registration of the cifX device driver .............................................................................. 10
  2.2 Annotations to Interrupt Support ............................................................................................... 11
  2.3 Device Configuration .................................................................................................................. 12
    2.3.1 Firmware and Configuration File Storage ............................................................................. 12
    2.3.2 Guideline to Create a Device Configuration ........................................................................ 14
    2.3.3 Providing the Configuration for RTX2009/2011/2012/RTX64 ........................................... 15
    2.3.4 Providing the Device Configuration for RTX 8.1 ............................................................... 15
    2.3.5 Description of the Device Configuration File ...................................................................... 16

3 Shared Memory Extension .............................................................................................................. 18
  3.1 Restrictions ................................................................................................................................ 20
  3.2 Installation .................................................................................................................................. 21
  3.3 Using the Shared Memory Service ............................................................................................. 22

4 Driver Specific Information ............................................................................................................ 23
  4.1 Additional Structures .................................................................................................................. 23
    4.1.1 Structure RTX_CIFXDRV_PARAMETERS_T ...................................................................... 23
    4.1.2 Structure RTX_CIFXDRV_DEVICEENTRY_T ..................................................................... 24
    4.1.3 Structure RTX_CIFXDRV_DEVICEBUSINFO_T ................................................................. 24
  4.2 Additional functions .................................................................................................................... 25
    4.2.1 cifXInitDriver() .................................................................................................................... 25
    4.2.2 cifXDeinitDriver() ................................................................................................................ 26
    4.2.3 cifXFindDevice() .................................................................................................................. 27
    4.2.4 cifXGetDriverVersion() ....................................................................................................... 28
  4.3 Driver Startup Procedure .......................................................................................................... 29
  4.4 Driver Log File ............................................................................................................................ 30

5 Programming with the Driver ....................................................................................................... 31

6 Frequently Asked Questions ........................................................................................................ 32

7 Appendix ......................................................................................................................................... 33
  7.1 Legal Notes .................................................................................................................................. 33
  7.2 Registered Trademarks ............................................................................................................... 36
  7.3 List of Tables ............................................................................................................................... 37
  7.4 List of Figures .............................................................................................................................. 37
  7.5 Contact ....................................................................................................................................... 38
1 Introduction

1.1 About this Document

This manual describes the Hilscher cifX Device Driver for IntervalZero RTX® and its architecture. The driver offers access to the Hilscher netX based hardware with the same functional API as the cifX device driver for Windows® and offers transparent access to the different devices. IntervalZero's RTX® software is a Real Time eXtension for the Windows operating system.

The Hilscher RTX® driver is represented as a dynamic loadable library, which supports different netX based hardware designs. First design is a cifX PCI/PCIe bus based PC card. This card is not using any FLASH memory and the driver is responsible to download the necessary firmware and configuration files during the startup phase of the hardware, to get the boards in operational mode. The second type of hardware, which is supported by the driver, is the NXSB-PCA and NX-PCA-PCI boards. These boards enable the connection of a NXSB 100 / NXHX board to the PCI bus. In addition, to the PCI based hardware, the driver also supports devices, connected via the ISA bus.

1.2 List of Revisions

<table>
<thead>
<tr>
<th>Rev</th>
<th>Date</th>
<th>Name</th>
<th>Chapter</th>
<th>Revisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2009-07-23</td>
<td>SS</td>
<td>6</td>
<td>created</td>
</tr>
<tr>
<td>2</td>
<td>2009-09-08</td>
<td>SS</td>
<td>2.3.1</td>
<td>Chapter ‘Frequently Asked Questions’ added</td>
</tr>
<tr>
<td>3</td>
<td>2010-06-09</td>
<td>SS</td>
<td>2.3</td>
<td>Support for DMA mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.3.1</td>
<td>Support for loadable modules</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.3.5</td>
<td>Note about use of registry file added</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.2</td>
<td>Return types of additional functions adapted to stdint data types</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.3.1.6</td>
<td>Slot number support</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.3</td>
<td>NXPCA-PCI Timings parameter support</td>
</tr>
<tr>
<td>4</td>
<td>2011-10-21</td>
<td>SS/RM</td>
<td>1.5</td>
<td>Support for netJACK 100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>Guideline for device configuration revised</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.3.1.x</td>
<td>Support for RTX 2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Device configuration is obtained directly from windows registry</td>
</tr>
<tr>
<td>5</td>
<td>2013-01-28</td>
<td>SS/RM</td>
<td>1.10</td>
<td>Added information about CIFX API manual</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.4, 1.8, 4.1</td>
<td>Information about cifX driver for RTX 2012 included</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.3</td>
<td>cifX driver for RTX2009 now supports direct registry access</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Add description for new setup application</td>
</tr>
<tr>
<td>6</td>
<td>2013-12-13</td>
<td>SS</td>
<td>1.2</td>
<td>Support for RTX64 (2013)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>New chapter: Shared Memory Extension introduced (Windows access)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.4</td>
<td>Log file support added</td>
</tr>
<tr>
<td>7</td>
<td>2015-08-06</td>
<td>RM</td>
<td>1.2</td>
<td>Support for RTX64 2014</td>
</tr>
<tr>
<td>8</td>
<td>2017-10-11</td>
<td>RM/LC</td>
<td>1.7</td>
<td>Interrupt handling under RTX only possible with none shared interrupt</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.2, 6</td>
<td>Support for RTX64 3.x</td>
</tr>
</tbody>
</table>

Table 1: List of Revisions
1.3 Overview

The cifX Device Driver for IntervalZero RTX® is available as a dynamic library built around the cifX Toolkit. Any application which needs to access a cifX device can use the device specific functions provided by this driver library. The dynamic library is implemented as a RTDLL, which is the analog of an explicitly loaded windows DLL. User processes on RTX till version 2012 gain access to the cifX driver functions by using the `LoadLibrary()` and `GetProcAddress()` calls (explicit library load). As RTX64 provides its own image loader, the driver library can also be loaded implicitly here. The concept of the cifX device driver is illustrated in the following figure.

![Figure 1: cifX Device Driver Architecture](image-url)
1.4 Requirement

- IntervalZero RTX® version 8.1, 2009(SP2 Update 1 or higher), 2011, 2012, RTX64 (2013), RTX64 2014 (SP2), RTX64 3.x
- cifX Device Driver for Windows® 8/7/Vista/XP/2000
- Microsoft Visual 2005 or higher for driver test /cifX TCP Server application
- Microsoft Visual 2010 for driver test /cifX TCP Server application on RTX64 (2013)
- Microsoft Visual 2013 for driver test /cifX TCP Server application on RTX64 2014
- Microsoft Visual 2017 for driver test /cifX TCP Server application on RTX64 3.x

1.5 Supported Hardware

- cifX board (PCI / PCIe)
- netPLC
- NXSB-PCA / NXSB100 / NXHX board
- NX-PCA-PCI / NXHX
- CIFX 104 (ISA)
- netJACK 100

1.6 Features

- Based on the cifX Toolkit source (V1.4.0.0)
- Interrupt support for cifX boards
- DMA data transfer for I/O data
- Support for loadable modules
- Interrupt notification for applications
- 64 bit support on RTX64 only

1.7 Limitations

- Only one RTX® process can use the driver at the same time
- Only little-endian hosts supported
- No DMA support for NXSB-PCA, NX-PCA-PCI and CIFX104
- Interrupt handling under RTX only possible if interrupt lines not shared with Windows devices (Interrupt must be available for exclusive use of RTX)
- CIFX50e (PCIexpress) does not support MSI
## 1.8 CD Contents

<table>
<thead>
<tr>
<th>Folder</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>cifX Driver API</td>
</tr>
<tr>
<td>Documentation</td>
<td>Driver documentation</td>
</tr>
<tr>
<td>Driver</td>
<td>Driver runtimes</td>
</tr>
<tr>
<td>RTX 8.1</td>
<td>Driver runtimes for RTX 8.1</td>
</tr>
<tr>
<td>RTX 2009</td>
<td>Driver runtimes for RTX 2009</td>
</tr>
<tr>
<td>RTX 2011</td>
<td>Driver runtimes for RTX 2011</td>
</tr>
<tr>
<td>RTX 2012</td>
<td>Driver runtimes for RTX 2012</td>
</tr>
<tr>
<td>RTX64 (2013)</td>
<td>Driver runtimes and setup information file for RTX64 (2013)</td>
</tr>
<tr>
<td>RTX64 2014</td>
<td>Driver runtimes and setup information file for RTX64 2014</td>
</tr>
<tr>
<td>RTX64 3.x</td>
<td>Driver runtimes and setup information file for RTX64 3.x</td>
</tr>
<tr>
<td>Examples</td>
<td>Examples</td>
</tr>
<tr>
<td>Example Configuration File</td>
<td>Example card configuration registry file</td>
</tr>
<tr>
<td>Test Application</td>
<td>Source code for driver test application</td>
</tr>
<tr>
<td>cifXTCPServer</td>
<td>Source code for cifX TCP Server application</td>
</tr>
<tr>
<td>Installation</td>
<td>Driver setup</td>
</tr>
<tr>
<td>SharedMemory Extension</td>
<td>Shared Memory Extension for accesses from windows environment</td>
</tr>
</tbody>
</table>

*Table 2: CD Contents*
1.9 Terms and Abbreviations

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cifX</td>
<td>Communication Interface based on netX</td>
</tr>
<tr>
<td>comX</td>
<td>Communication Module based on netX</td>
</tr>
<tr>
<td>PCI</td>
<td>Peripheral Component Interconnect</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>DPM</td>
<td>Dual-Port-Memory Physical interface to all communication board (DPM is also used for PROFIBUS-DP Master).</td>
</tr>
</tbody>
</table>

Table 3: Terms and Abbreviations

1.10 References to Documents

This manual is based on the following documents and specifications:


Table 4: References to Documents
2 Installation

To use a cifX device in the RTX® environment, it must be explicitly removed from Windows® Device Manager and prepared for RTX® support.

With release of version 1.0.3.0 of the cifX device driver, installation is performed via a dedicated setup application (see step 5).

Several steps must be done to install the Hilscher RTX® driver:

The installation is provided by the system CD which comes with the hardware or can be downloaded from the Hilscher WEB site (www.hilscher.com).

Step 2: Configure your cifX device (see section Device Configuration on page 12) with the cifX Setup Tool, included with the cifX device driver for Windows®.

Note: An online connection to a cifX device is not available if the cifX device is already converted to a RTX® device. Thus, configuration of the cifX device should be done in advance.

Note: The device conversion process has changed in RTX64 SP1. Please consult reference [6] for further details. The required setup information file is installed automatically on your system during installation of the driver. For manual installation, the setup information file is also located on the installation CD.

Step 3: Add RTX® support for a cifX device (not required on RTX64).
- Start the RTX Properties dialogue from Windows® start menu
- Open the 'Hardware' tab and enter the Plug and Play configuration via the 'Settings' button.
- Select the cifX device which should be controlled with the RTX® software, choose 'Add RTX INF support' and select 'Apply'.

Figure 2: Add RTX® Support for a cifX Device

© Hilscher 2009-2017
Step 4: Converting a Windows® Device to a RTX® Device (Windows® XP)

- Open the Windows® device manager
- Select the cifX device which should be converted and choose 'Update driver'
- Select 'Let me pick from a list of device drivers on my computer'
- To device drivers are listed: The standard cifX driver for Windows® and the cifX driver with RTX support. Select the cifX driver with RTX support to convert the selected cifX device to a RTX® device.

![Figure 3: Convert the Windows® Device to a RTX® Device](image)

Note: A detailed description of the conversion process for other Windows® systems is given in reference [5] and [6].

Step 5: Installation of the cifX device driver for RTX via the setup application

- Execute the setup application cifX RTX Driver Setup.exe from the installation CD. The setup will copy the adequate driver to the given destination directory and register the driver library to the RTX subsystem automatically.

![Setup application](image)

- After the installation has completed the cifX device can be accessed via RTX subsystem!
2.1 Manual registration of the cifX device driver

Attention: - Only required for cifX driver V1.0.2.0 and older versions
- Not needed for RTX64 (RTX64 provides its own image loader, registration is not used anymore)

Step 1: Copy the cifX RTX® driver
- Copy the Hilscher RTX® driver DLL \textit{cifXRTXDrv.rtdll} for your RTX® version from the installation CD to the local hard disk.

Step 2: Register the driver DLL (not required on RTX64)
- The driver DLL \textit{cifXRTXDrv.rtdll} must be registered within the RTX® environment. Therefore the RTX® program \textit{RTSSrun} can be used. The RTX® environment will generate a local copy of the driver DLL and makes the driver DLL available for RTSS applications.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{RTSSrun_program_example.png}
\caption{RTSSrun Program Example}
\end{figure}

\textbf{Note:} You have to repeat this step if you getting an updated driver DLL

\textbf{Note:} As the cifX driver DLL cannot be loaded by more than one RTX® process simultaneously, make sure that the option ‘Share between processes‘ is unchecked.
2.2 Annotations to Interrupt Support

The cifX device can be used in polling or interrupt mode.
If interrupt mode should be used, IRQ resource limitations must be considered.
As RTX® does not support sharing IRQ lines with Windows® devices, the interrupt line which RTX® uses must be available for exclusive use.
Finding an exclusive IRQ often requires physically moving hardware around in the system or disabling other Windows® devices.

If you have difficulty configuring your hardware for use with RTX® due to IRQ resource limitations, please consult reference [4] for further information.

Note: Actual Hilscher PCIe cards are not supporting MSI interrupts
2.3 Device Configuration

Each device has several parameters which can be configured, e.g. Alias Name or Interrupt Support. Some of the parameters are also important to get the device working, e.g. Bootloader and firmware and fieldbus configuration files.

Note: Perform the initial device configuration before assigning your cifX device to the RTX® environment, as the cifX device is no longer accessible via a Windows® process.

The device configuration will be done with the cifX Driver Setup Utility, included with the cifX device driver for Windows® and the complete device configuration information is stored in the Windows® registry.

Also the cifX device driver for RTX® needs these configuration and depending on the RTX® runtime version, access to the Windows® registry and therewith to the device configuration is available or not.

For RTX® versions without registry access (RTX® version prior RTX® 2009) the registry data must be exported into a file to be accessible by the cifX device driver for RTX®.

If the cifX device is already converted to a RTX® device, the online connection to the cifX device is no longer available. Because of this the initial device configuration must be done in advance.

2.3.1 Firmware and Configuration File Storage

cifX PCI cards are not using any FLASH memory to store a firmware or configuration on the card. Every time the card is powered-up the firmware and configuration must be downloaded to the hardware.

Note: Firmware and configuration files are not stored on the hardware and must be downloaded each time the card is powered-up.

It is the task of the driver to initialize the card and therefore the driver has to know which files must be loaded to the hardware.

To allow device specific configuration, every file that needs to be downloaded must reside in the folder tree which is created by the cifX device driver for Windows® (or manually generates). The location of this folder tree is passed to the cifX RTX® driver via a configuration file (see section Device Configuration on page 12).

- Use the Slotnumber (hardware rotary switch)
  The Slotnumber serves to distinguish cifX cards from each other clearly, especially if more cifX cards are installed in one PC. The Slotnumber must be set at the cifX card using the Rotary Switch Slotnumber. While Slotnumber 0 means, that the cifX card is identified via its device and serial number, values from 1 to 9 corresponds to the Slotnumber 1 to 9. The firmware and configuration file must reside in the subdirectory Slot_<1..9>.

- Use the device and serial number (default)
  If the cifX device is not equipped with a rotary switch or the Slotnumber should not be used, the device is identified by its device and serial number. The firmware and configuration file must reside in the subdirectory /<Device Number>_/<Serial Number>/.
The following table describes the different subdirectory levels created by the cifX device driver for Windows®:

<table>
<thead>
<tr>
<th>Subdirectory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;InstallDir&gt;</code></td>
<td>Installation directory of the cifX device driver for Windows®.</td>
</tr>
<tr>
<td>• <code>&lt;Device Nr.&gt;_&lt;_Serial Nr.&gt;</code></td>
<td>Device and serial number of the device or slot number if the device provides a rotary switch. If the slot number is 0 the device and serial number is always used to identify the device.</td>
</tr>
<tr>
<td>• Slot_&lt;1..9&gt;</td>
<td>Note: This directory must contain the second stage PCI bootloader (e.g. NETX100-BSL.bin)</td>
</tr>
<tr>
<td>Channel&lt;#&gt;</td>
<td>Channel specific files (loadable modules, monolithic firmware files, fieldbus database files)</td>
</tr>
<tr>
<td></td>
<td>Note: Currently only channel 0 is supported</td>
</tr>
</tbody>
</table>

Table 5: Firmware and Configuration File Storage

Sample file structure for a cifX device with device number 1250100 and serial number 20217:

```
+ `<InstallDir`
  |-- NETX100-BSL.BIN (second stage PCI bootloader)
  |--- 1250100_20217
     -- Channel0
     |--- cifXdps.nxf (monolithic firmware)
     |--- config.nxd (fieldbus database)
     -- Channel1
     -- Channel2
     -- Channel3
     -- Channel4
     -- Channel5
```

Sample file structure for a cifX device identified by Slot number 2 and loadable module support:

```
+ `<InstallDir`
  |-- NETX100-BSL.BIN (second stage PCI bootloader)
  |--- Slot_2
     -- Channel0
     |--- nx100dpm.nxo (loadable module)
     |--- config.nxd (fieldbus database)
     -- Channel1
     -- Channel2
     -- Channel3
     -- Channel4
     -- Channel5
     -- cifXrcX.nxf (rcX base firmware)
```
2.3.2 Guideline to Create a Device Configuration

A cifX device needs a configuration defining the firmware and fieldbus configuration files for each cifX device. This chapter describes how to create a card configuration and how to make it available to the cifX device driver for RTX®.

**Note:** Creating a device configuration is done under the Windows® operating system.

- Configure your cifX device with the cifX Driver Setup Utility.
  - A configuration instruction can be found in the CIFX Device Driver Installation and Operation manual (cifX Device Driver_usermanual_en.pdf).
  - Start the utility from the Windows® Control Panel and insert the necessary information.
  - Press the Apply button and confirm the request to restart the device.
  - This will store the device configuration in the Windows® registry and create the required directory tree for the device configuration and firmware file storage (see section Firmware and Configuration File Storage on page 12).

![cifX Driver Setup Utility](image)
2.3.3 Providing the Configuration for RTX2009/2011/2012/RTX64

RTX 2009 / 2011 / 2012 / RTX64 offering direct access to the Windows® registry via a real-time application. This new feature is used by cifX RTX® driver to access the device configuration and no further initialization steps are necessary.

2.3.4 Providing the Device Configuration for RTX 8.1

The Windows® registry is not accessible via the RTX application, thus the device configuration must be exported to a ‘.reg’ file. For this purpose the export function of the Windows® registry editor must be used.

The location and the filename of this registry file must be supplied to the cifX RTX® driver via the driver parameters (see section Structure RTX_CIFXDRV_PARAMETERS_T on page 23).

Exporting the device configuration from the Windows® registry:

- Open the Windows® Start menu and select 'Run'
- Execute the Windows® Registry Editor by entering the command 'regedit' and confirm via the OK button.
- In the Windows® Registry Editor, navigate to the registry key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CIFxDrv
- In the file menu select Export….
  Make sure that the file type is Registration Files (.reg) or Win9x/NT4 Registration files (.reg). Enter an arbitrary filename and select Save. Remember the file name and the file location!
- In your RTX application you have to pass the path to the exported registry file to the driver initialization routine cifXInitDriver() via the driver parameter szRegFile (see section Structure RTX_CIFXDRV_PARAMETERS_T on page 23).

A sample configuration file is also located on the installation CD and can also be manually modify. Relevant configuration keys are listed and described in the following chapter.
2.3.5 Description of the Device Configuration File

A sample configuration file is located on the installation CD. To modify the configuration file manually, the relevant configuration keys are listed below.

### Key Value

<table>
<thead>
<tr>
<th>Key Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![HKLM\SYSTEM\CurrentControlSet\Services\CIFxDrv]</td>
<td>Path to installation directory, created by the cifX device driver for Windows®</td>
</tr>
<tr>
<td>![HKLM\SYSTEM\CurrentControlSet\Services\CIFxDrv\DeviceConfig&lt;Device Nr.&gt;&lt;Serial Nr.&gt;]</td>
<td></td>
</tr>
<tr>
<td>![HKLM\SYSTEM\CurrentControlSet\Services\CIFxDrv\DeviceConfig\Slot&lt;Slot Nr.&gt;]</td>
<td></td>
</tr>
<tr>
<td>Alias</td>
<td>Statically assigned alias name</td>
</tr>
</tbody>
</table>
| InterruptEnable | Device interrupt handling enable/disable  
0 = disable (default)  
1 = enabled |
| DMAEnable | Direct memory access enable/disable  
0 = disable (default)  
1 = enable |
| OSFile | To use loadable modules, a rcX base firmware is required. This entry specifies the path to the rcX base firmware file relative to:  
- <InstallationDir>\<Device Number>_\<Serial Number>  
- <InstallationDir>\Slot_\<Slot Nr.> |
| ![HKLM\SYSTEM\CurrentControlSet\Services\CIFxDrv\DeviceConfig\<Device Nr.>\<Serial Nr.>\Channel<0..6>] | | |
| ![HKLM\SYSTEM\CurrentControlSet\Services\CIFxDrv\DeviceConfig\Slot\<Slot Nr.>\Channel<0..6>] | | |
| Note: | Currently the firmware only supports channel 0 |
| ModuleCount | Number of configured loadable modules / monolithic firmware files |
| Module<0..ModuleCount> | Each loadable module or monolithic firmware file gets an own entry with an index as suffix. This specifies the path to the module file relative to:  
- <InstallationDir>\<Dev. Number>_\<Serial Number>\Channel<0..6>  
- <InstallationDir>\Slot_\<Slot Nr.>\Channel<0..6> |
| ConfigCount | Number of configured fieldbus databases (configuration files) |
| Config<0..ModuleCount> | Each database gets an own entry with an index as suffix. This specifies the path to the database file relative to:  
- <InstallationDir>\<Dev. Number>_\<Serial Number>\Channel<0..6>  
- <InstallationDir>\Slot_\<Slot Nr.>\Channel<0..6> |
| WarmstartFile | Full file name to warmstart parameter file residing in directory:  
- <InstallationDir>\<Dev. Number>_\<Serial Number>\Channel<0..6>  
- <InstallationDir>\Slot_\<Slot Nr.>\Channel<0..6>  
**Note:** Warmstart file support depends on the used firmware and is maybe not supported |
| ![HKLM\SYSTEM\CurrentControlSet\Services\CIFxDrv\Parameters] | Driver log trace level (See section 4.4 for details) |
| ![HKLM\SYSTEM\CurrentControlSet\Services\CIFxDrv\Parameters\NX-PCA-PCI] | NXPCA-PCI Timings parameter for 8 bit mode.  
DPM_8_Bit |
| DPM_16_Bit | NXPCA-PCI Timings parameter for 16 bit mode. |
| DPM_32_Bit | NXPCA-PCI Timings parameter for 32 bit mode. |

Table 6: Device Configuration File Keys
Sample configuration registry file:

<table>
<thead>
<tr>
<th>Registry Key</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet</td>
<td>Services\CIFxDrv]</td>
</tr>
<tr>
<td>&quot;InstallDir&quot;=&quot;C:\Programme\CifX Device Driver&quot;</td>
<td></td>
</tr>
<tr>
<td>[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet</td>
<td>Services\CIFxDrv\DeviceConfig]</td>
</tr>
<tr>
<td>[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet</td>
<td>Services\CIFxDrv\DeviceConfig\1250100]</td>
</tr>
<tr>
<td>&quot;Alias&quot;=&quot;&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;InterruptEnable&quot;=dword:00000001</td>
<td></td>
</tr>
<tr>
<td>&quot;DMAEnable&quot;=dword:00000001</td>
<td></td>
</tr>
<tr>
<td>&quot;OSFile&quot;=&quot;cifXrcX.nxf&quot;</td>
<td></td>
</tr>
<tr>
<td>[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet</td>
<td>Services\CIFxDrv\DeviceConfig\1250100\20217\Channel0]</td>
</tr>
<tr>
<td>&quot;ModuleCount&quot;=dword:00000001</td>
<td></td>
</tr>
<tr>
<td>&quot;Module0&quot;=&quot;cifXdps.nxf&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;ConfigCount&quot;=dword:00000001</td>
<td></td>
</tr>
<tr>
<td>&quot;Config0&quot;=&quot;config.nxd&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;WarmstartFile&quot;=&quot;warmstart.dat&quot;</td>
<td></td>
</tr>
<tr>
<td>[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet</td>
<td>Services\CIFxDrv\DeviceConfig\1250100\20217\Channel1]</td>
</tr>
<tr>
<td>&quot;ModuleCount&quot;=dword:00000000</td>
<td></td>
</tr>
<tr>
<td>&quot;ConfigCount&quot;=dword:00000000</td>
<td></td>
</tr>
<tr>
<td>[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet</td>
<td>Services\CIFxDrv\DeviceConfig\1250100\20217\Channel2]</td>
</tr>
<tr>
<td>&quot;ModuleCount&quot;=dword:00000000</td>
<td></td>
</tr>
<tr>
<td>&quot;ConfigCount&quot;=dword:00000000</td>
<td></td>
</tr>
<tr>
<td>[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet</td>
<td>Services\CIFxDrv\DeviceConfig\1250100\20217\Channel3]</td>
</tr>
<tr>
<td>&quot;ModuleCount&quot;=dword:00000000</td>
<td></td>
</tr>
<tr>
<td>&quot;ConfigCount&quot;=dword:00000000</td>
<td></td>
</tr>
<tr>
<td>[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet</td>
<td>Services\CIFxDrv\DeviceConfig\1250100\20217\Channel4]</td>
</tr>
<tr>
<td>&quot;ModuleCount&quot;=dword:00000000</td>
<td></td>
</tr>
<tr>
<td>&quot;ConfigCount&quot;=dword:00000000</td>
<td></td>
</tr>
<tr>
<td>[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet</td>
<td>Services\CIFxDrv\DeviceConfig\1250100\20217\Channel5]</td>
</tr>
<tr>
<td>&quot;ModuleCount&quot;=dword:00000000</td>
<td></td>
</tr>
<tr>
<td>&quot;ConfigCount&quot;=dword:00000000</td>
<td></td>
</tr>
</tbody>
</table>
3 Shared Memory Extension

Windows applications may access cifX devices which are under RTX control via a special shared memory extension.

**Note:** Due to the limited performance of the shared memory interface, the shared memory interface should be used for diagnostic purposes only.

---

**General Overview:**

![General Overview of the Shared Memory Service](image)

*Figure 7: General Overview of the Shared Memory Service*
This extension consists of a RTX process “cifXd” and two libraries for the RTX (“cifX32Dll.rtdll”) and Windows environment (“cifX32Dll.dll”). The "cifXd" process has exclusive control over the cifX devices and provides a shared memory interface to enable access from various RTX and Windows applications.

A RTX and a Windows version of the cifX32Dll library allow any CIFX API compatible application to access the "cifXd"-controlled devices via the shared memory interface.

The concept of the shared memory interface is illustrated in the following figure.

![Shared Memory Extension Diagram](image-url)

*Figure 8: Shared Memory Extension*
3.1 Restrictions

Although cifX32Dll.rtdll and cifXRTXDrv.rtdll are interchangeable in terms of its API, some extended functionalities are restricted or may not work at all via the shared memory interface.

Affected CIFX API functions:

- xSysdeviceBootstart()
- xSysdeviceExtendedMemory()
- xChannelDMAState()
- xChannelRegisterNotification(), xChannelUnregisterNotification()
- xChannelSyncState()
- xChannelPLCMemoryPtr(), xChannelPLCIsReadReady(), xChannelPLCIsWriteReady(), xChannelPLCActivateWrite(), xChannelPLCActivateRead()

Although linking against those API functions succeeds, they will always return with the error code CIFX_FUNCTION_NOT_AVAILABLE.
3.2 Installation

Installation of the shared memory extension is performed automatically during installation of the cifX RTX Device Driver.

This feature can also be selected/deselected in the custom setup dialog:

**ATTENTION:** Re-installation of the original cifX Device Driver for Windows will override any cifX32dll.dll exchanged by the RTX driver installation in the .\Windows\System32 and .\Windows\SysWOW64 directory.

If you want to use the Shared Memory Service again, it is necessary to manually copy the shared memory enabled cifX32DII files again to .\Windows\System32 and .\Windows\SysWOW64 directory!

Selecting the Shared Memory Service will exchange the cifX32dll.dll in your .\Windows\System32 directory (typically C:\Windows\System32) with the shared memory enabled cifX32DII variant enclosed with this setup application.

On 64bit Windows systems the installer also exchanges the cifX32Dll.dll in the .\Windows\SysWOW64 directory (RTX64 only).

*Figure 9: Install shared memory extension to enable access from windows application*
3.3 Using the Shared Memory Service

The shared memory extension demon process ("cifXd") has to be running in background as long as access to the cifX devices from a Windows application via the cifX32Dll is required.

To start the "cifXd" process, navigate to the Windows "Start" menu and execute the "Start Shared Memory Service (cifXd)" shortcut (see figure below) or execute the "cifXd" process manually via a command line.

Command line to start Shared Memory Service:

RTX8 / 2009 / 2011 / 2012: rtssrun cifXd.rtss start
RTX64: rtssrun cifXd64.rtss start

Figure 1: cifXd Process Shortcuts in the Windows Start Menu

The CIFX API compatible cifX32dll.dll (for Windows applications) and the cifX32dll.rtdll (for RTX applications) should be used now to access cifX devices via the "cifXd" shared memory process.

To stop the "cifXd" process, use either the appropriate shortcut in the Windows "Start" menu or stop the process manually by using a command line.

Command line to stop Shared Memory Service:

RTX64: rtssrun cifXd64.rtss stop

Although concurrent access from RTX and Windows is possible, it is not recommended to run several applications on one and the same cifX device at the same time.

Process based packet routing is not supported by the "cifXd" process, i.e. packet data transfers may fail as it can not be guaranteed that a packet reply is only delivered to that application which has dispatched the request packet before.

Note: As long as the "cifXd" process is running, RTX applications must not access cifX devices via the origin driver library (cifXRTXDrv.rtdll).

The CIFX API compatible cifX32dll.rtdll can be used instead to access cifX devices via shared memory from the RTX environment.
4 Driver Specific Information

The cifX driver for RTX® needs some special initialization functions and structures.

4.1 Additional Structures

Some of the RTX® specific functions need parameters provided through structures. The structures and the meaning of the internal data are described in the following chapter.

4.1.1 Structure RTX_CIFXDRV_Parameters_T

This structure is used to initialize the cifX driver.

<table>
<thead>
<tr>
<th>Element</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| fScanPCI        | int                        | Driver Initialization options:  
|                 |                            | 0 = FALSE  
|                 |                            | Driver does not scan for PCI cards. Cards can be added manually by ulUserDevCount and ptUserDevList parameters.  
|                 |                            | 1 = TRUE  
|                 |                            | Driver scans for all available PCI cards and adds them to the application.                                                                      |
| ulTraceLevel    | unsigned long              | **RTX 2009/2011/2012/RTX64:**  
|                 |                            | not used (obtained via registry, see section 2.3.5)                                                                                           |
|                 |                            | **RTX 8.1:**  
|                 |                            | Set the trace level of the driver:  
|                 |                            | 1 = TRACE_LEVEL_DEBUG  
|                 |                            | 2 = TRACE_LEVEL_INFO  
|                 |                            | 4 = TRACE_LEVEL_WARNING  
|                 |                            | 8 = TRACE_LEVEL_ERROR  
| ulPollInterval  | unsigned long              | Polling interval in milliseconds [ms] for non-interrupt driven cards (used for Change of State (COS) signaling, see reference [3])  
|                 |                            | 0 = 500ms default                                                                                                                             |
| szRegFile       | const char*                | **RTX 2009/2011/2012/RTX64:**  
|                 |                            | not used  
|                 |                            | **RTX 8.1:**  
|                 |                            | Set the path to the Windows® registry file. Can be NULL to use the default of 'c:\CIFxDrv.reg'. Creation of this registry file is explained in section Device Configuration on page 12.  
|                 |                            | NULL = 'c:\CIFxDrv.reg' default                                                                                                             |
| ulUserDevCount  | unsigned long              | Number of user cards entries in the ptUserDevList to add to the driver (e.g. if a card is connected via DPM).  
|                 |                            | 0 = none                                                                                                                                     |
| ptUserDevList   | RTX_CIFXDRV_DEVICEENTRY_T ** | Array of user added cards. Number of entries is defined by ulUserDevCount. See section Structure RTX_CIFXDRV_DEVICEENTRY_T on page 24. |

Table 7: Structure RTX_CIFXDRV_Parameters_T Definition
4.1.2 Structure RTX_CIFXDRV_DEVICEENTRY_T

This structure describes a cifX device which should be added to the driver. This structure can be acquired through `cifXFindDevice()` or filled by the user if a custom card should be added.

<table>
<thead>
<tr>
<th>Element</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ulPhysicalAddress</td>
<td>unsigned long</td>
<td>Physical address of the DPM (this value is used to detect the PCI card linked to the DPM)</td>
</tr>
<tr>
<td>bIrqNumber</td>
<td>unsigned char</td>
<td>Interrupt number</td>
</tr>
<tr>
<td>pvDPMAddress</td>
<td>void*</td>
<td>Virtual pointer to card DPM</td>
</tr>
<tr>
<td>ulDPMSize</td>
<td>unsigned long</td>
<td>Size of the DPM in bytes</td>
</tr>
<tr>
<td>fPCICard</td>
<td>int</td>
<td>0 = FALSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Device is connected via DPM.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = TRUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Device is connected to PCI bus</td>
</tr>
<tr>
<td>tBusInfo</td>
<td>RTX_CIFXDRV_DEVICEBUSINFO_T</td>
<td>Bus information, see section Structure RTX_CIFXDRV_DEVICEBUSINFO_T on page 24</td>
</tr>
</tbody>
</table>

Table 8: Structure RTX_CIFXDRV_DEVICEENTRY_T Definition

4.1.3 Structure RTX_CIFXDRV_DEVICEBUSINFO_T

Bus information structure used to store bus specific information for cifX devices connected via PCI bus.

<table>
<thead>
<tr>
<th>Element</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ulBusNumber</td>
<td>int</td>
<td>Bus number</td>
</tr>
<tr>
<td>ulSlotNumber</td>
<td>int</td>
<td>Slot number</td>
</tr>
</tbody>
</table>

Table 9: Structure RTX_CIFXDRV_DEVICEBUSINFO_T Definition

4.2 Additional functions

This chapter describes functions which are only available for the RTX® version of the driver. These functions are used to initialize the cifX device driver.

4.2.1 cifXInitDriver()

This function must be called before accessing any driver function. The cifX driver initialization includes discovering all available cifX PCI devices and downloading the firmware and configuration files.

Note: **cifXInitDriver()** calls **cifXFIndDevice()** internally to search for available cifX PCI cards (**fScanPCI** = TRUE). If the cards should be defined manually, **cifXFIndDevice()** can be used to add cards manually (**fScanPCI** = FALSE).

Note: For none PCI cards, the application has to create a RTX_CIFXDRV_DEVICEENTRY_T with the corresponding device information. Afterwards, the structure must be placed into the RTX_CIFXDRV_PARAMETERS_T structure (see parameters **ulUserDevCount**, **ptUserDevList**, **fScanPCI** = FALSE) before **cifXInitDriver()** is called.

Function call:

```
int32_t cifXInitDriver (RTX_CIFXDRV_PARAMETERS_T*  ptDriverParams);
```

Arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ptDevEntry</td>
<td>RTX_CIFXDRV_PARAMETERS_T*</td>
<td>Driver parameters, see section Structure RTX_CIFXDRV_PARAMETERS_T on page 23</td>
</tr>
</tbody>
</table>

Return Values:

<table>
<thead>
<tr>
<th>Return Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFX_NO_ERROR</td>
<td>Driver initialization successful</td>
</tr>
<tr>
<td>CIFX_DRV_INIT_ERROR</td>
<td>Driver initialization failed (no cifX device available)</td>
</tr>
</tbody>
</table>

Example:

```c
RTX_CIFXDRV_VTABLE tVTable = {0};
RTX_CIFXDRV_PARAMETERS_T tDriverParams = {0};
HANDLE hDll = NULL;
char* szRegFile = "C:\CIFxDrv.reg";

/* Set driver parameters */
tDriverParams.fScanPCI = TRUE;
tDriverParams.ulUserDevCount = 0;
tDriverParams.szRegFile = szRegFile;

/* Load cifXRTXDrv.rtdll */
hDll = LoadLibrary("cifXRTXDrv.dll");

/* Assign function pointer */
tVTable.cifXInitDriver = (RTXDRV_INIT) GetProcAddress( hDll, "cifXInitDriver");

/* Scan for all available cifX PCI devices and initialize the cifX device driver */
tVTable.cifXInitDriver (tDriverParams);
FreeLibrary(hDll);
```
### 4.2.2 cifXDeinitDriver()

Un-initialize the driver and remove all devices from the control of the cifX driver library. After calling this function the application must not access any cifX driver API function any more.

**Function call:**

```c
int32_t cifXDeinitDriver ( void );
```

**Arguments:**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Return Values:**

<table>
<thead>
<tr>
<th>Return Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFX_NO_ERROR</td>
<td>Driver deinitialization successful</td>
</tr>
<tr>
<td>CIFX_DEV_HW_PORT_IS_USED</td>
<td>Device is in use, so driver deinitialization is denied</td>
</tr>
<tr>
<td>CIFX_DRV_NOT_INITIALIZED</td>
<td>Driver was not initialized</td>
</tr>
</tbody>
</table>
4.2.3 **cifXFindDevice()**

This function scans for a cifX PCI device in the system and builds a RTX_CIFXDRV_DEVICEENTRY_T structure for each discovered device.

The function is used internally by *cifXInitDriver()* if PCI cards should be detected automatically (RTX_CIFXDRV_PARAMETERS_T, fScanPCI parameter = TRUE).

cifXFindDevice() can also be used by an application to manually define a cifX card. Therefore the function can be called with a fixed device number (*iNum*). If the given device is available, the function returns the RTX_CIFXDRV_DEVICEENTRY_T structure for the device.

Afterwards the application has to insert this structure into the RTX_CIFXDRV_PARAMETERS_T structure (see section Structure RTX_CIFXDRV_PARAMETERS_T on page 23), setting the *fScanPCI* flag to FALSE and calling *cifXInitDriver()*.

**Function call:**

```c
BOOL cifXFindDevice ( RTX_CIFXDRV_DEVICEENTRY_T* ptDevEntry, int iNum);
```

**Arguments:**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ptDevEntry</td>
<td>RTX_CIFXDRV_DEVICEENTRY_T*</td>
<td>Pointer to a RTX_CIFXDRVDEVICEENTRY_T structure, to place returned values in</td>
</tr>
<tr>
<td>iNum</td>
<td>int</td>
<td>Number of the device in the system</td>
</tr>
</tbody>
</table>

**Return Values:**

<table>
<thead>
<tr>
<th>Return Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>A device with number <em>iNum</em> was found</td>
</tr>
<tr>
<td>FALSE</td>
<td>A device with number <em>iNum</em> could not be found</td>
</tr>
</tbody>
</table>

**Example:**

```c
RTX_CIFXDRV_VTABLE   tVTable = {0};
RTX_CIFXDRV_PARAMETERS_T    tDriverParams = {0};
RTX_CIFXDRV_DEVICEENTRY_T tDevEntry = {0};
HANDLE hDll = NULL;
/* Load cifXRTXDrv.rtdll */
hDll = LoadLibrary("cifxRTXDrv.dll");
/* Assign function pointer */
tVTable.cifXInitDriver = (RTXDRV_INIT) GetProcAddress( hDll, "cifXInitDriver");
tVTable.cifXFindDevice = (RTXDRV_FIND_DEVICE) GetProcAddress( hDll, "cifXFindDevice");
/* Find the first cifX PCI device */
if (tVTable.cifXFindDevice (&tDevEntry, 0)) {
    tDriverParams.fScanPCI       = FALSE;
    tDriverParams.ulUserDevCount = 1;
    tDriverParams.ptUserDevList  = &tDevEntry;
    /* initialize the cifX device driver */
    tVTable.cifXInitDriver (tDriverParams);
}
FreeLibrary(hDll);
```
4.2.4 cifXGetDriverVersion()

This function requests the version of the cifX driver for RTX®.

**Function call:**

```c
int32_t cifXGetDriverVersion ( char* szVersion, uint32_t ulSize )
```

**Arguments:**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>szVersion</td>
<td>char*</td>
<td>String buffer to return the version of the cifX driver</td>
</tr>
<tr>
<td>ulSize</td>
<td>uint32_t</td>
<td>Maximum buffer size for the version string</td>
</tr>
</tbody>
</table>

**Return Values:**

<table>
<thead>
<tr>
<th>Return Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFX_NO_ERROR</td>
<td>Driver version successfully copied to string buffer</td>
</tr>
<tr>
<td>CIFX_INVALID_BUFFERSIZE</td>
<td>Supplied buffer too small to hold the driver version string</td>
</tr>
</tbody>
</table>

**Example:**

```c
cifX_CIFXDRV_VTABLE tVTable   = {0};
char    szDrvVersion[20] = "";
HANDLE   hDll    = NULL;
/* Load cifXRTXDrv.rtdll */
hDll = LoadLibrary("cifxRTXDrv.dll");
/* Assign function pointer */
tVTable.cifXGetDriverVersion = (RTXDRV_GET_VERSION)
    GetProcAddress( hDll, "cifXGetDriverVersion" );
/* Get driver version */
tVTable.cifXGetDriverVersion( szDrvVersion,
    sizeof(szDrvVersion)/sizeof(*szDrvVersion) );
/* Print driver version to screen */
RtPrintf("%s", szDrvVersion);
FreeLibrary(hDll);
```
4.3 Driver Startup Procedure

The driver startup procedure can be controlled by the user. The following two use cases are available:

- Automatically add all found cifX PCI devices and optionally add user specific devices
- Skip cifX PCI device scan and add user specific device manually

```
Application  cifXRTXDrv.rtdll  cifX Toolkit

LoadLibrary("cifXRTXDrv.dll")

cifXInitDriver()  (fScanPCI = TRUE)

This will be done for every available cifX device

return CIFX_NO_ERROR

Figure 10: Initialization of the cifX Driver with fScanPCI = TRUE

Application  cifXRTXDrv.rtdll  cifX Toolkit

LoadLibrary("cifXRTXDrv.dll")

Set ptUserDevList and ptUserDevCount of driver parameter
cifXInitDriver()  (fScanPCI = FALSE)

cifX Toolkit

Set ptUserDevList and ptUserDevCount of driver parameter
cifX Toolkit

cifX Toolkit

return CIFX_NO_ERROR

Figure 11: Initialization of the cifX Driver with fScanPCI = FALSE
```
4.4 Driver Log File

The cifX driver for RTX is able to create log files on the startup which can be used for debug and support purposes.

Note: This feature was added in Version 1.0.4.0! Older versions reporting log traces on the RtxServer console only.

For each cifX device in the system, a separate log file is created using the naming convention "cifX<0..n>.log".

Log files are stored in the installation directory of the cifX Device Driver for Windows(!) (typically "C:\Program Files\cifX Device Driver").

The driver installation directory is also stored in the Windows registry (see section 2.3.5 for further details).

Trace granularity can be adjusted by setting the trace level accordingly:

- TRACE_LEVEL_ERROR
- TRACE_LEVEL_WARNING
- TRACE_LEVEL_INFO
- TRACE_LEVEL_DEBUG

TRACE_LEVEL is also stored in the Windows registry and can be adjusted via the cifXSetup utility or manually via the Windows registry editor (regedit).
5 Programming with the Driver

The Application Programming Interface (API) of the Hilscher RTX® driver is based on the already known CIFX API (reference [1]). The CIFX API is documented in the 'CIFX API - Application Programming Interface' manual, describing the API functions, error codes and shows some program examples using the API.

The installation CD also includes a 'Test Application' directory with a RTX® specific example.

The following C application demonstrates the minimum functions which must be called to enable an application to work with a cifX device.

**Note:** RTX64 3.x requires the ‘cifxRTX64Drv.rtdll’ to be loaded.

```c
#include <windows.h>
#include <rtapi.h>
#include "cifXUser.h"
#include "cifXErrors.h"
#include "cifxRTXDrv.h"

/*******************************************************************************/
/*! The main function */
* \return 0 on success                                                     */
/*******************************************************************************/
int main(int argc, char* argv[])
{
    HANDLE     hDll   = NULL;
    CIFXHANDLE hDriver  = NULL;
    RTX_CIFXDRV_VTABLE tVTable  = {0};
    RTX_CIFXDRV_PARAMETERS_T tDriverParams  = {0};
    long     lRet   = CIFX_NO_ERROR;
    char*     szRegFile  = "C:\CIFxDrv.reg";

    /* Set driver parameters */
    tDriverParams.fScanPCI       = TRUE;
    tDriverParams.ulUserDevCount = 0;
    tDriverParams.szRegFile      = szRegFile;
    tDriverParams.ulTraceLevel   = TRACE_LEVEL_ERROR   | TRACE_LEVEL_WARNING |
                                TRACE_LEVEL_INFO    | TRACE_LEVEL_DEBUG;

    /* Load cifXRTXDrv.rtdll */
    hDll = LoadLibrary("cifxRTXDrv.dll");

    /* Assign function pointer */
    tVTable.cifXInitDriver = (RTXDRV_INIT) GetProcAddress(hDll, "cifXInitDriver");
    tVTable.cifXDeinitDriver = (RTXDRV_DEINIT) GetProcAddress(hDll, "cifXDeinitDriver");
    tVTable.xDriverOpen = (X_DRIVER_OPEN) GetProcAddress(hDll, "xDriverOpen");
    tVTable.xDriverClose = (X_DRIVER_CLOSE) GetProcAddress(hDll, "xDriverClose");
    /* TODO: assign further api function */

    /* Initialize and open the cifX device driver */
    if( (CIFX_NO_ERROR == (lRet = tVTable.cifXInitDriver(&tDriverParams))) &&
        (CIFX_NO_ERROR == (lRet = tVTable.xDriverOpen(&hDriver))) )
    {
        /* Work with the cifX API */

        /* Close the cifX driver */
        tVTable.xDriverClose(hDriver);
    } else
    {
        RtPrintf("Error opening driver. lRet=0x%08X\r\n", lRet);
    }

    /* Deinit the cifX device driver */
    tVTable.cifXDeinitDriver(hDriver);
    FreeLibrary(hDll);
    return 0;
}
```
6 Frequently Asked Questions

Q: The driver presents the following warning at startup: 'OS_Memalloc: Not enough free memory in RTSS memory pool! Request from windows memory pool'. What does that mean?

A: To fulfill RTSS applications memory requests, RTX deterministically allocates memory from a special memory pool (Local memory pool). RTSS applications that allocate memory from that pool don't have to initiate a Service Request Interrupt to request memory from Windows (This request would be non-deterministic). If the memory needs of the RTSS application exceeds the initial size of the local memory pool, RTX initiate a non-deterministic Service Request Interrupt to request memory from Windows. This action is signaled to the user via the drivers warning message. This scenario can be avoided, by increasing the initial local memory pool size to the memory needs of the driver and the RTSS application (see memory settings in the RTX properties application). If the creation of the local memory pool is not done at startup, but with the first call to a local memory pool allocation function the drivers warning message is also signaled. The creation of the local memory pool at startup is initiated, a) by executing the RTSS application with the parameter /local or b) by modifying the memory settings in the RTX properties application (see RTX Manual).

Q: Is it possible to configure a cifX card, independently of its serial and device number?

A: The cifX Device Driver for RTX from version 1.0.1.0 on identifies cifX cards alternatively via its slot number. Please note that your cifX device must provide a rotary switch to use slot number support.

Q: What versions of Visual Studio are used for the driver examples?

A: This depends on the used RTX version and each example contains one or more Visual Studio project files for different versions of RTX and Visual Studio. If the version of your Visual Studio is not included, create a new VS project and add the source and header files to the newly created project. IntervalZero discontinues support for outdated Visual Studio versions with each new release of RTX. Please refer to the RTX/RTX64 documents given by IntervalZero: http://intervalzero.com/assets/RTX2012/OSCompatMatrixRTX.pdf and http://intervalzero.com/assets/RTX642014/RTX64OSCompatibilityMatrix.pdf for a complete overview!

Q: The compiled test application crashes the system. What's wrong?

A: The example applications delivered with the driver CD are created with different versions of Visual Studio for different RTX versions. If you have problems to compile or run an example make sure to adjust the linker path of the Visual Studio specific RTX libraries.
7 Appendix

7.1 Legal Notes

Copyright

© Hilscher Gesellschaft für Systemautomation mbH
All rights reserved.
The images, photographs and texts in the accompanying materials (in the form of a user's manual, operator's manual, Statement of Work document and all other document types, support texts, documentation, etc.) are protected by German and international copyright and by international trade and protective provisions. Without the prior written consent, you do not have permission to duplicate them either in full or in part using technical or mechanical methods (print, photocopy or any other method), to edit them using electronic systems or to transfer them. You are not permitted to make changes to copyright notices, markings, trademarks or ownership declarations. Illustrations are provided without taking the patent situation into account. Any company names and product designations provided in this document may be brands or trademarks by the corresponding owner and may be protected under trademark, brand or patent law. Any form of further use shall require the express consent from the relevant owner of the rights.

Important notes
Utmost care was/is given in the preparation of the documentation at hand consisting of a user's manual, operating manual and any other document type and accompanying texts. However, errors cannot be ruled out. Therefore, we cannot assume any guarantee or legal responsibility for erroneous information or liability of any kind. You are hereby made aware that descriptions found in the user's manual, the accompanying texts and the documentation neither represent a guarantee nor any indication on proper use as stipulated in the agreement or a promised attribute. It cannot be ruled out that the user's manual, the accompanying texts and the documentation do not completely match the described attributes, standards or any other data for the delivered product. A warranty or guarantee with respect to the correctness or accuracy of the information is not assumed.

We reserve the right to modify our products and the specifications for such as well as the corresponding documentation in the form of a user's manual, operating manual and/or any other document types and accompanying texts at any time and without notice without being required to notify of said modification. Changes shall be taken into account in future manuals and do not represent an obligation of any kind, in particular there shall be no right to have delivered documents revised. The manual delivered with the product shall apply.
Under no circumstances shall Hilscher Gesellschaft für Systemautomation mbH be liable for direct, indirect, ancillary or subsequent damage, or for any loss of income, which may arise after use of the information contained herein.
Liability disclaimer

The hardware and/or software was created and tested by Hilscher Gesellschaft für Systemautomation mbH with utmost care and is made available as is. No warranty can be assumed for the performance or flawlessness of the hardware and/or software under all application conditions and scenarios and the work results achieved by the user when using the hardware and/or software. Liability for any damage that may have occurred as a result of using the hardware and/or software or the corresponding documents shall be limited to an event involving willful intent or a grossly negligent violation of a fundamental contractual obligation. However, the right to assert damages due to a violation of a fundamental contractual obligation shall be limited to contract-typical foreseeable damage.

It is hereby expressly agreed upon in particular that any use or utilization of the hardware and/or software in connection with

- Flight control systems in aviation and aerospace;
- Nuclear fission processes in nuclear power plants;
- Medical devices used for life support and
- Vehicle control systems used in passenger transport

shall be excluded. Use of the hardware and/or software in any of the following areas is strictly prohibited:

- For military purposes or in weaponry;
- For designing, engineering, maintaining or operating nuclear systems;
- In flight safety systems, aviation and flight telecommunications systems;
- In life-support systems;
- In systems in which any malfunction in the hardware and/or software may result in physical injuries or fatalities.

You are hereby made aware that the hardware and/or software was not created for use in hazardous environments, which require fail-safe control mechanisms. Use of the hardware and/or software in this kind of environment shall be at your own risk; any liability for damage or loss due to impermissible use shall be excluded.
Warranty

Hilscher Gesellschaft für Systemautomation mbH hereby guarantees that the software shall run without errors in accordance with the requirements listed in the specifications and that there were no defects on the date of acceptance. The warranty period shall be 12 months commencing as of the date of acceptance or purchase (with express declaration or implied, by customer's conclusive behavior, e.g. putting into operation permanently).

The warranty obligation for equipment (hardware) we produce is 36 months, calculated as of the date of delivery ex works. The aforementioned provisions shall not apply if longer warranty periods are mandatory by law pursuant to Section 438 (1.2) BGB, Section 479 (1) BGB and Section 634a (1) BGB [Bürgerliches Gesetzbuch; German Civil Code] If, despite of all due care taken, the delivered product should have a defect, which already existed at the time of the transfer of risk, it shall be at our discretion to either repair the product or to deliver a replacement product, subject to timely notification of defect.

The warranty obligation shall not apply if the notification of defect is not asserted promptly, if the purchaser or third party has tampered with the products, if the defect is the result of natural wear, was caused by unfavorable operating conditions or is due to violations against our operating regulations or against rules of good electrical engineering practice, or if our request to return the defective object is not promptly complied with.

Costs of support, maintenance, customization and product care

Please be advised that any subsequent improvement shall only be free of charge if a defect is found. Any form of technical support, maintenance and customization is not a warranty service, but instead shall be charged extra.

Additional guarantees

Although the hardware and software was developed and tested in-depth with greatest care, Hilscher Gesellschaft für Systemautomation mbH shall not assume any guarantee for the suitability thereof for any purpose that was not confirmed in writing. No guarantee can be granted whereby the hardware and software satisfies your requirements, or the use of the hardware and/or software is uninterruptable or the hardware and/or software is fault-free.

It cannot be guaranteed that patents and/or ownership privileges have not been infringed upon or violated or that the products are free from third-party influence. No additional guarantees or promises shall be made as to whether the product is market current, free from deficiency in title, or can be integrated or is usable for specific purposes, unless such guarantees or promises are required under existing law and cannot be restricted.
Confidentiality

The customer hereby expressly acknowledges that this document contains trade secrets, information protected by copyright and other patent and ownership privileges as well as any related rights of Hilscher Gesellschaft für Systemautomation mbH. The customer agrees to treat as confidential all of the information made available to customer by Hilscher Gesellschaft für Systemautomation mbH and rights, which were disclosed by Hilscher Gesellschaft für Systemautomation mbH and that were made accessible as well as the terms and conditions of this agreement itself.

The parties hereby agree to one another that the information that each party receives from the other party respectively is and shall remain the intellectual property of said other party, unless provided for otherwise in a contractual agreement.

The customer must not allow any third party to become knowledgeable of this expertise and shall only provide knowledge thereof to authorized users as appropriate and necessary. Companies associated with the customer shall not be deemed third parties. The customer must obligate authorized users to confidentiality. The customer should only use the confidential information in connection with the performances specified in this agreement.

The customer must not use this confidential information to his own advantage or for his own purposes or rather to the advantage or for the purpose of a third party, nor must it be used for commercial purposes and this confidential information must only be used to the extent provided for in this agreement or otherwise to the extent as expressly authorized by the disclosing party in written form. The customer has the right, subject to the obligation to confidentiality, to disclose the terms and conditions of this agreement directly to his legal and financial consultants as would be required for the customer's normal business operation.

Export provisions

The delivered product (including technical data) is subject to the legal export and/or import laws as well as any associated regulations of various countries, especially such laws applicable in Germany and in the United States. The products / hardware / software must not be exported into such countries for which export is prohibited under US American export control laws and its supplementary provisions. You hereby agree to strictly follow the regulations and to yourself be responsible for observing them. You are hereby made aware that you may be required to obtain governmental approval to export, reexport or import the product.

7.2 Registered Trademarks

Please notice: RTX® is a registered trademark of IntervalZero.


All other mentioned trademarks are property of their respective legal owners.
7.3 List of Tables

Table 1: List of Revisions .................................................................................................................. 3
Table 2: CD Contents ......................................................................................................................... 6
Table 3: Terms and Abbreviations ..................................................................................................... 7
Table 4: References to Documents .................................................................................................... 7
Table 5: Firmware and Configuration File Storage .......................................................................... 13
Table 6: Device Configuration File Keys ............................................................................................ 16
Table 7: Structure RTX_CIFXDRV_PARAMETERS_T Definition .................................................... 23
Table 8: Structure RTX_CIFXDRV_DEVICEENTRY_T Definition ..................................................... 24
Table 9: Structure RTX_CIFXDRV_DEVICEBUSINFO_T Definition .................................................. 24

7.4 List of Figures

Figure 1: cifX Device Driver Architecture ......................................................................................... 4
Figure 2: Add RTX® Support for a cifX Device ................................................................................ 8
Figure 3: Convert the Windows® Device to a RTX® Device .............................................................. 9
Figure 4: RTSSrun Program Example ................................................................................................ 10
Figure 5: cifX Driver Setup Utility .................................................................................................... 14
Figure 6: Export the Device Configuration with the Windows® Registry Editor .............................. 15
Figure 7: General Overview of the Shared Memory Service ............................................................. 18
Figure 8: Shared Memory Extension ................................................................................................. 19
Figure 9: Install shared memory extension to enable access from windows application .................. 21
Figure 10: Initialization of the cifX Driver with fScanPCI = TRUE .................................................. 29
Figure 11: Initialization of the cifX Driver with fScanPCI = FALSE ................................................... 29
7.5 Contact

Headquarters

Germany
Hilscher Gesellschaft für Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

Support
Phone: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

Subsidiaries

China
Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support
Phone: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

France
Hilscher France S.a.r.l.
69500 Bron
Phone: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support
Phone: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

India
Hilscher India Pvt. Ltd.
Pune, Delhi, Mumbai
Phone: +91 8888 750 777
E-Mail: info@hilscher.in

Italy
Hilscher Italia S.r.l.
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-Mail: info@hilscher.it

Support
Phone: +39 02 25007068
E-Mail: it.support@hilscher.com

Japan
Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support
Phone: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Korea
Hilscher Korea Inc.
Seongnam, Gyeonggi, 463-400
Phone: +82 (0) 31-789-3715
E-Mail: info@hilscher.kr

Switzerland
Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support
Phone: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA
Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-Mail: info@hilscher.us

Support
Phone: +1 630-505-5301
E-Mail: us.support@hilscher.com