

# CIFX API

The CIFX-API is the common *CIFX/COMX/netX Application Programming Interface (CIFX API)* and describes the containing functions, offered for all Hilscher standard devices based on netX controller hardware.

Aim of the API is to provide applications a target and fieldbus independent programming interface to netX based hardware running a standard Hilscher fieldbus protocol or firmware which meet the Hilscher netX dual port memory (netX DPM) definitions, described in the '*netX Dual Port Memory Interface*' manual).

The API is designed to give the user easy access to all of the communication board functionalities.

## API Function Overview

Function Group	Function	Description	
Driver	xDriverOpen	Opens the driver, allowing access to every driver function	
	xDriverClose	Closes an open connection to the driver	
	xDriverGetInformation	Retrieves driver information (e.g. Version)	
	xDriverGetErrorDescription	Retrieves an English description of a cifX driver error code	
	xDriverEnumBoards	Enumerate through all boards/devices the driver is managing	
	xDriverEnumChannels	Enumerate through all channels located on a specific board	
	xDriverRestartDevice	Restart a device	
	xDriverMemoryPointer	Get/Release a pointer to the dual port memory. <b>This function should only be used for debugging purpose</b>	
Function Group	Function	Description	
System Device	xSysdeviceOpen	Opens a connection to a boards system device	
	xSysdeviceClose	Closes a connection to a system device	
	<b>Device Administrative/Informational functions</b>		
	xSysdeviceInfo	Get System device specific information (e.g. mailbox size)	
	xSysdeviceReset	Perform a device reset	
	xSysdeviceBootstart	Perform a device boot start. This will activate the 2 <sup>nd</sup> Stage bootloader. An available firmware will not be started. <b>Note:</b> Only possible on FLASH based devices.	
	xSysdeviceDownload	Downloads a file/configuration/firmware to the device	
	xSysdeviceUpload	Uploads a file/configuration/firmware from the device	
	xSysdeviceFindFirstFile	Find the first file entry in the given directory	
	xSysdeviceFindNextFile	Find the next file entry in the given directory	
	xSysdeviceExtendedMemory	Get a pointer to an available extended memory area	
	<b>Asynchronous services (Packets)</b>		
	xSysdeviceGetMBXState	Retrieves the system mailbox state	
	xSysdeviceGetPacket	Retrieves a pending packet from the system mailbox	
	xSysdevicePutPacket	Send a packet to the system mailbox	
	Function Group	Function	Description

Communication Channel	xChannelOpen	Opens a connection to a communication channel
	xChannelClose	Closes a connection
	<b>Asynchronous services (Packets)</b>	
	xChannelGetMBXState	Retrieve the channels mailbox state
	xChannelGetPacket	Retrieve a pending packet from the channel mailbox
	xChannelPutPacket	Send a packet to the channel mailbox
	xChannelGetSendPacket	Read back the last sent packet
	<b>Device Administrational/Informational functions</b>	
	xChannelDownload	Download a file/configuration to the channel
	xChannelReset	Reset the channel
	xChannelInfo	Retrieve channel specific information
	xChannelWatchdog	Activate/Deactivate/Trigger the channel Watchdog
	xChannelHostState	Set the application state flag in the application COS flags, to signal the hardware if an application is running or not
	xChannelBusState	Set the bus state flag in the application COS state flags, to start or stop fieldbus communication.
	xChannelControlBlock	Access the channel control block
	xChannelCommonStatusBlock	Access to the common status block
	xChannelExtendedStatusBlock	Access to the extended status block
	xChannelUserBlock	Access user block (not implemented yet!)
	<b>Cyclic Data services (I/O's)</b>	
	xChannelIORead	Instructs the device to place the latest data into the DPM and passes them to the user
	xChannelIOWrite	Copies the data to the DPM and waits for the firmware to retrieve them
	xChannelIOReadSendData	Reads back the last send data
	<b>Cyclic Data services (I/O's, PLC optimized)</b>	
	xChannelPLCMemoryPtr	Get a pointer to the I/O memory block
	xChannelPLCActivateRead	Instruct the firmware to place the latest input data into the I/O memory block (no wait for completion)
	xChannelPLCActivateWrite	Instruct the firmware to retrieve the latest output data from the I/O memory block (no wait for completion)
	xChannelPLCIsReadReady	Checks if the last read activation has finished
	xChannelPLCIsWriteReady	Checks if the last write activation has finished
	<b>DMA services</b>	
	xChannelDMAState	Activate/Deactivate DMA mode
	<b>Bus synchronous operation</b>	
	xChannelSyncState	Wait for a synchronization event or trigger / acknowledge a sync event
	<b>Notification services (only available in Interrupt mode)</b>	
xChannelRegisterNotification	Register a notification callback	
xChannelUnregisterNotification	Un-register a notification callback	

In addition, Hilscher also offers a free of charge *cifX Toolkit* (C-source code based) which allows to write own drivers based on the Hilscher netX DPM definitions including the *CIFX API* functions (the toolkit is described in a separate *cifX/netX Toolkit* manual, see NXDRV-TKIT).

Documentation						
Page	Document type	Document title	Content	Date	Language	File type
<a href="#">cifX API (Revision 6)</a>	Programming reference guide	<a href="#">cifX API</a>	Description and usage of the standard cifX API.	2019-04	English	PDF