JBC

www.jbctools.com

Connect Library

Programmer's Guide

0011580/0114

DLL Version: 1.66.76.0

Contents

1 Introduction

The JBC Connect Library provides complete functionality to monitor and control JBC stations. In order to use this library and, in a more general view, to connect JBC stations to a PC, the proper USB to UART bridge driver must be installed. This driver is provided with the library distribution.

The library has been developed using Microsoft Visual Studio 2010 and is provided as a Dynamic Link Library (DLL); this means that it only can be used in Windows OS. The library is implemented in VisualBasic using .NET Framework 3.5.

The library (DLL), two test applications with source code and the driver for the USB to UART bridge are included in the library distribution set.

1.1 System Requirements

Operating System: Windows XP / Vista / 7

Processor: Intel i3

RAM: at least 1GB, 2GB recommended

Hard disk: 2MB available hard disk space

1.2 Supported Stations

The JBC stations supported by the library are listed below. The minimum program version needed is showed for each station model.

Station	Minimum program version needed
DM	9996761
DD	9996762
DDR	9996779
DI	9996763
HD	9996764
HDR	9996780
CD/CF	9996765
CS/CV	9996766
CP	9996767
NA	9996768

2 Library Description

The library is composed by one main class and two tool classes. The main class is called **JBC_API**. Inside this class are all the public constants, data structures, events and functions required by the programmer or, in other words, served by the library. The other two classes are **Ctemperature**, to store a temperature, and **Cerror**, to indicate errors to the user. These three classes are stored in the namespace **JBC_Connect**.

2.1 JBC_API Class

This is the main class of the library. It must be created in order to use the library. In this class, constant lists (enumerations), data structures (structure), events and methods are found:

- ***Constant lists**:* These lists of constants are used as a variable type for some of the parameters of the methods in the class. For example, a list of possible station tools. All of the constant lists start with the character "c", for example "cGenericStationTools".

- ***Data structures**:* There are some data structures defined to be used as a parameter of some methods, typically the continuous mode methods. All of these structures start with the character "t", for example "tContinuousModeData".

- ***Events**:* There are three events in the class. Two of them are used to detect the connection and disconnection of JBC stations to the PC and the third one is used to pass errors (using the Cerror class previously indicated).

- ***Methods**:* There are several methods in this class. Those methods are the API and are designed in a C-style library, usually starting with "Get" or "Set".

This class can be created indicating if the station will be set to Control Mode when it connects.

jbc_api = **New** JBC_Connect.JBC_API(**bStationsInitialControlMode**)

If **bStationsInitialControlMode** is false, the library sets the station to Monitor Mode when it connects.

If **bStationsInitialControlMode** is true, the library sets the station to Control Mode when it connects.

By default, if no parameter is passed, the library sets the station to Control Mode.

2.1.1 Constants

- ***cStationError**:* List of internal station errors.
- ***cTemperatureLimits**:* List of maximum and minimum temperatures supported by the station.
- ***cTemperatureUnits**:* List of temperature units available for the station.
- ***cOnOff**:* Used in some methods as ON or OFF status value.
- ***cGenericStationTools**:* List of general tool models.
- ***cToolError**:* List of internal station port errors.
- ***cToolTemperatureLevels**:* List of temperature levels available for the tools.
- ***cToolTimeSleep**:* List of possible sleep delay times.
- ***cToolTimeHibernation**:* List of possible hibernation delay times.
- ***cToolFutureMode**:* List of modes automatically reached by the station.
- ***cPowerLimits**:* List of station power limits in Watts. It depends on the station model. A value of 0 means no power limit for that model.
- ***cSpeedContinuousMode**:* List of available speeds (periods) for the continuous mode.
- ***cToolStatus**:* List of tool status.
- ***cPort**:* General list of station ports.

2.1.2 Structures

***tContinuousModeStatus**:* Defines the status of the continuous mode, with selected ports and speed.

1. **port1**: used to indicate if port 1 is active in continuous mode.
2. **port2**: used to indicate if port 2 is active in continuous mode.
3. **port3**: used to indicate if port 3 is active in continuous mode.
4. **port4**: used to indicate if port 4 is active in continuous mode.
5. **speed**: used to indicate the continuous mode speed (period).

***tContinuousModePort**:* Defines the data received from a port in continuous mode.

1. **port***:* used to indicate from which port this data is coming.
2. **temperature***:* used to pass the port current tool tip temperature.
3. **power***:* used to pass the port current tool power consumption in per thousand units.
4. **status***:* used to pass the current port tool status.

***tContinuousModeData**:* Defines the data returned by the continuous mode.

1. **data***:* a vector of tContinuousModePort structures that contains the information reported by the station for all the ports activated in the continuous mode.
2. **sequence***:* used to pass the corresponding sequence ordering value for the current data transmission.

2.1.3 Events

NewStationConnected:

- *Description:* This event is launched when a new station has been connected to the PC.
- *Prototype:* Public **Event** NewStationConnected(ByVal stationID as **Ulong**)
- *Params:* **stationID** – The identifier of the detected station. User must use this ID in order to use the API methods to manage the station.

***StationDisconnected**:*

- *Description:* This event is launched when a station has been disconnected from the PC.
- *Prototype:* Public **Event** StationDisconnected (ByVal stationID as **Ulong)**
- *Params:* **stationID** – The identifier of the disconnected station. User must not use this ID anymore.

***UserError**:*

- *Description:* This event is launched when a method detects an error on its input parameters. User must use this event in order to catch these errors. In the methods descriptions that follow these errors are specified.
- *Prototype:* Public **Event** UserError(ByVal err as **Cerror)**
- *Params:* **err** – The error occurred as a Cerror object. See Cerror class description for more information.

***TransactionFinished**:*

- *Description:* This event is launched when the library receives the response for the SetTransaction function. You can use this function to know that all previous functions have been processed by the station.
- *Prototype:* Public **Event** TransactionFinished(ByVal stationID as **Ulong** , ByVal transactionID as **UInteger)**
- *Params:* **stationID** – The identifier of the station.

  transactionID – The transaction number returned when the SetTransaction function was executed.

2.1.4 Methods

Following is the list of all the methods for this class that is that main one. Each of these methods is described with five parameters:

- Description: General method behaviour description.
- Prototype: VisualBasic method prototype code.
- Parameters: Explanation of the parameters of the method.
- Returns: If the method is a function type that returns a value here is described this value.
- Errors: The possible errors thrown by the event "UserError" as Cerror objects for the method and its cause.

An important thing to note is that when a user error is thrown by the event those methods that return values will return a non valid value.

2.1.4.1 DefaultStationParameters

Description: Restores default values for all the parameters of the indicated station.

Prototype: Public **sub** DefaultStationParameters(ByVal stationID as **ULong**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.2 SetControlMode

Description: Activates or deactivates the control mode. When the control mode is activated the PC can change or set the station parameters; when deactivated, PC can only monitor or get the station parameters but never set them. It is also important to note that when the control mode is active the station only allow the PC to set its parameters.

Prototype: Public **sub** SetControlMode(ByVal stationID as **Ulong**, ByVal OnOff as **cOnOff)**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

OnOff – The desired ON or OFF status for the control mode.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.3 GetControlMode

Description: Returns the current control mode status. When the control mode is activated the PC can change or set the station parameters, when deactivated PC can only monitor or get the station parameters but never set them. It also is important to note that when the control mode is active the station only allow the PC to set its parameters.

Prototype: Public **function** GetControlMode(ByVal stationID as **Ulong)** as **cOnOff**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The current ON or OFF status of the control mode.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.4 SetRemoteMode

Description: Activates or deactivates the remote mode. When the remote mode is activated the PC can change the different station tools status (Sleep, Hibernation, Desolder, Extractor) without the need of manually put the tool on its stand. This method must be called in order to use the methods that change tool's status, which are:

SetPortToolStandStatus()

SetPortToolExtractorStatus()

SetPortToolDesolderStatus()

It is important to note that when remote mode is active the tool only changes its status by a PC order (using the mentioned methods).

Prototype: Public **sub** SetRemoteMode(ByVal stationID as **Ulong**, ByVal OnOff as **cOnOff)**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

OnOff – The desired ON or OFF status for the remote mode.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.5 GetRemoteMode

Description: Returns the current remote mode status. When the remote mode is activated the PC can change the different station tools status (Sleep, Hibernation, Desolder, Extractor) without the need of manually put the tool on its stand. This method must be called in order to use the methods that change tool's status, which are:

SetPortToolStandStatus()

SetPortToolExtractorStatus()

SetPortToolDesolderStatus()

It is important to note that when remote mode is active the tool only changes its status by a PC order (using the mentioned methods).

Prototype: Public **function** GetRemoteMode(ByVal stationID as **Ulong)** as **cOnOff**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The current ON or OFF status of the remote mode.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.6 SetPortToolStandStatus

Description: Sets a tool at stand status to ON or OFF when the remote mode is ON, see SetRemoteMode(). The stand status is the one you get when the tool is put in the stand.

Prototype: Public **sub** SetPortToolStandStatus(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal OnOff as **cOnOff)**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

OnOff – The desired ON or OFF status for the stand status.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.7 SetPortToolExtractorStatus

Description: Sets a tool at extractor status to ON or OFF when the remote mode is ON, see SetRemoteMode().

Prototype: Public **sub** SetPortToolExtractorStatus(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal OnOff as **cOnOff)**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

OnOff – The desired ON or OFF status for the extractor status.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.8 SetPortToolDesolderStatus

Description: Sets a tool at desolder status to ON or OFF when the remote mode is ON, SetRemoteMode().

Prototype: Public **sub** SetPortToolDesolderStatus(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal OnOff as **cOnOff)**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

OnOff – The desired ON or OFF status for the desolder status.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.9 GetPortToolStandStatus

Description: Returns the current stand ON or OFF status of the tool at the indicated port.

Prototype: Public **function** GetPortToolStandStatus(ByVal stationID as **Ulong**,
ByVal port as **cPort**) as **cOnOff**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The current ON or OFF status of the stand mode.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.10 GetPortToolSleepStatus

Description: Returns the current sleep ON or OFF status of the tool at the indicated port.

Prototype: Public **function** GetPortToolSleepStatus(ByVal stationID as **Ulong**,
ByVal port as **cPort**) as **cOnOff**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The current ON or OFF status of the sleep mode.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.11 GetPortToolHibernationStatus

Description: Returns the current hibernation ON or OFF status of the tool at the indicated port.

Prototype: Public **function** GetPortToolHibernationStatus(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **cOnOff**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The current ON or OFF status of the hibernation mode.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.12 GetPortToolExtractorStatus

Description: Returns the current extractor ON or OFF status of the tool at the indicated port.

Prototype: Public **function** GetPortToolExtractorStatus(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **cOnOff**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The current ON or OFF status of the extractor mode.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.13 GetPortToolDesolderStatus

Description: Returns the current desolder ON or OFF status of the tool at the indicated port.

Prototype: Public **function** GetPortToolDesolderStatus(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **cOnOff**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The current ON or OFF status of the desolder mode.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.14 GetStationCOM

Description: Returns the COM port name of the indicated station.

Prototype: Public **function** GetStationCOM(ByVal stationID as **Ulong)** as **String**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The COM port name of the station. Ex: "COM5"

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.15 GetStationProtocol

Description: Returns the communications protocol version of the indicated station.

Prototype: Public **function** GetStationProtocol(ByVal stationID as **Ulong)** as **String**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The communications protocol version of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.16 GetStationModel

Description: Returns the model name of the indicated station.

Prototype: Public **function** GetStationModel(ByVal stationID as **Ulong)** as **String**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The model name of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.17 GetStationHWversion

Description: Returns the hardware version of the indicated station.

Prototype: Public **function** GetStationHWversion(ByVal stationID as **Ulong)** as **String**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The hardware version of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.18 GetStationSWversion

Description: Returns the software version of the indicated station.

Prototype: Public **function** GetStationSWversion(ByVal stationID as **Ulong)** as **String**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The software version of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.19 GetStationError

Description: Returns the current station error code of the indicated station. See the cStationError constant description to more knowledge in the station error codes.

Prototype: Public **function** GetStationError(ByVal stationID as **Ulong)** as **cStationError**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The station error code of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.20 GetStationTransformerTemp

Description: Returns the current transformer temperature of the indicated station.

Prototype: Public **function** GetStationTransformerTemp(ByVal stationID as **Ulong)** as **Ctemperature**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The transformer temperature of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.21 GetStationPluggedMinutes

Description: Returns the current number of connected minutes of the indicated station.

Prototype: Public **function** GetStationPluggedMinutes(ByVal stationID as **Ulong)** as **Integer**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The number of connected minutes of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.22 GetPortCount

Description: Returns the number of ports of the indicated station. It depends on the model.

Prototype: Public **function** GetPortCount(ByVal stationID as **Ulong)** as **Integer**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The number of ports of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.23 GetPortToolID

Description: Returns the model of the tool at the indicated port.

Prototype: Public **function** GetPortToolID(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **cGenericStationTools**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The model of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.24 GetPortToolActualTemp

Description: Returns the current tip temperature of the tool at the indicated port.

Prototype: Public **function** GetPortToolActualTemp(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **Ctemperature**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The current tip temperature of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.25 GetPortToolActualPower

Description: Returns the current power consumption of the tool at the indicated port in per thousand units.

Prototype: Public **function** GetPortToolActualPower(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **Integer**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The current power consumption of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.26 GetPortToolError

Description: Returns the current error code of the tool at the indicated port. See cToolError constant definitions for more information on tool error codes.

Prototype: Public **function** GetPortToolError(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **cToolError**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The current error code of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.27 GetPortToolMOStemp

Description: Returns the current MOSFET temperature of the indicated port.

Prototype: Public **function** GetPortToolMOStemp(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **Ctemperature**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The current MOSFET temperature of the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.28 GetPortToolFutureMode

Description: Returns the future mode of the tool at the indicated port. The future mode can be sleep or hibernation. This function indicates the next mode the station will achieve if the delay time is not interrupted.

Prototype: Public **function** GetPortToolFutureMode(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **cToolFutureMode**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The future mode of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.29 GetPortToolTimeToFutureMode

Description: Returns the remaining time to the future mode of the tool at the indicated port in seconds. See GetPortToolFutureMode().

Prototype: Public **function** GetPortToolTimeToFutureMode(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **integer**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The remaining time to the future mode of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.30 GetPortToolWorkMinutes

Description: Returns the number of working minutes of the tool at the indicated port.

Prototype: Public **function** GetPortToolWorkMinutes(ByVal stationID as **Ulong**,
ByVal port as **cPort**) as **Integer**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The working minutes of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.31 GetPortToolSleepMinutes

Description: Returns the number of sleep minutes of the tool at the indicated port.

Prototype: Public **function** GetPortToolSleepMinutes(ByVal stationID as **Ulong**,
ByVal port as **cPort**) as **Integer**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The sleep minutes of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.32 GetPortToolHibernationMinutes

Description: Returns the number of hibernation minutes of the tool at the indicated port.

Prototype: Public **function** GetPortToolHibernationMinutes(ByVal stationID as **Ulong**,
ByVal port as **cPort**) as **Integer**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The hibernation minutes of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.33 GetPortToolIdleMinutes

Description: Returns the number of idle minutes of the tool at the indicated port. Idle means no tool connected at the port.

Prototype: Public **function** GetPortToolIdleMinutes(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **Integer**

Parameters **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The idle minutes of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.34 GetPortToolSleepCycles

Description: Returns the number of sleep cycles of the tool at the indicated port.

Prototype: Public **function** GetPortToolSleepCycles(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **Integer**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The sleep cycles of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.35 GetPortToolDesolderCycles

Description: Returns the number of desolder cycles of the tool at the indicated port.

Prototype: Public **function** GetPortToolDesolderCycles(ByVal stationID as **Ulong**, ByVal port as **cPort**) as **Integer**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The desolder cycles of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.36 GetStationName

Description: Returns the name of the indicated station.

Prototype: Public **function** GetStationName(ByVal stationID as **Ulong)** as **String**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The name of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.37 SetStationName

Description: Sets the name of the indicated station.

Prototype: Public **sub** SetStationName(ByVal stationID as **Ulong**, ByVal newName as **String)**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

newName – The new desired name for the station.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

INVALID_STATION_NAME – if the station name passed is not valid.

2.1.4.38 GetStationPIN

Description: Returns the PIN code of the indicated station.

Prototype: Public **function** GetStationPIN(ByVal stationID as **Ulong)** as **String**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The PIN code of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.39 SetStationPIN

Description: Sets the PIN code of the indicated station.

Prototype: Public **sub** SetStationPIN(ByVal stationID as **Ulong**, ByVal newPIN as **String)**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

newPIN – The new desired PIN for the station. A string of 4 numbers.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

INVALID_STATION_PIN – if the station PIN passed is not valid.

2.1.4.40 GetStationMaxTemp

Description: Returns the maximum temperature of the indicated station.

Prototype: Public **function** GetStationMaxTemp(ByVal stationID as **Ulong)** as **Ctemperature**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The maximum temperature of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.41 SetStationMaxTemp

Description: Sets the maximum temperature of the indicated station.

Prototype: Public **sub** SetStationMaxTemp(ByVal stationID as **Ulong**,
ByVal newTemp as **Ctemperature**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

newTemp – The desired new maximum temperature.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

TEMPERATURE_OUT_OF_RANGE – if the temperature passed is outside acceptable ranges.

2.1.4.42 GetStationMinTemp

Description: Returns the minimum temperature of the indicated station.

Prototype: Public **function** GetStationMinTemp(ByVal stationID as **Ulong)** as **Ctemperature**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The minimum temperature of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.43 SetStationMinTemp

Description: Sets the minimum temperature of the indicated station.

Prototype: Public **sub** SetStationMinTemp(ByVal stationID as **Ulong**,
ByVal newTemp as **Ctemperature**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

newTemp – The desired new minimum temperature.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

TEMPERATURE_OUT_OF_RANGE – if the temperature passed is outside acceptable ranges.

2.1.4.44 GetStationTempUnits

Description: Returns the current temperature units used by the indicated station.

Prototype: Public **function** GetStationTempUnits(ByVal stationID as **Ulong)** as
cTemperatureUnits

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The current temperature units used by the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.45 SetStationTempUnits

Description: Sets the temperature units used by the indicated station.

Prototype: Public **sub** SetStationTempUnits(ByVal stationID as **Ulong**,
ByVal newUnits as **cTemperatureUnits**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

newUnits – The desired new temperature units.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.46 GetStationN2Mode

Description: Returns the current nitrogen mode ON or OFF status of the indicated station.

Prototype: Public **function** GetStationN2Mode(ByVal stationID as **Ulong)** as **cOnOff**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The current nitrogen mode ON or OFF status of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.47 SetStationN2Mode

Description: Sets the nitrogen mode ON or OFF status of the indicated station.

Prototype: Public **sub** SetStationN2Mode(ByVal stationID as **Ulong**, ByVal newMode as **cOnOff**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

newMode – The desired new nitrogen mode status.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.48 GetStationHelpText

Description: Returns the current help text ON or OFF status of the indicated station.

Prototype: Public **function** GetStationHelpText(ByVal stationID as **Ulong)** as **cOnOff**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The current help text ON or OFF status of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.49 SetStationHelpText

Description: Sets the help text ON or OFF status of the indicated station.

Prototype: Public **sub** SetStationHelpText(ByVal stationID as **Ulong**, ByVal newHelp as **cOnOff**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

newHelp – The desired new help text status.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.50 GetStationBeep

Description: Returns the current Beep ON or OFF status of the indicated station.

Prototype: Public **function** GetStationBeep(ByVal stationID as **Ulong)** as **cOnOff**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The current Beep ON or OFF status of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.51 SetStationBeep

Description: Sets the Beep ON or OFF status of the indicated station.

Prototype: Public **sub** SetStationBeep(ByVal stationID as **Ulong**, ByVal beep as **cOnOff**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

beep – The desired new beep ON or OFF status.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.52 GetPortToolSelectedTemp

Description: Returns the current selected tip temperature of the tool at the indicated port.

Prototype: Public **function** GetPortToolSelectedTemp(ByVal stationID as **Ulong**,
ByVal port as **cPort)** as **Ctemperature**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

Returns: The current selected tip temperature of the tool at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.53 SetPortToolSelectedTemp

Description: Sets the selected tip temperature of the tool at the indicated port.

Prototype: Public **sub** SetPortToolSelectedTemp(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal newTemp as **Ctemperature**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

newTemp – The desired new selected temperature.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

TEMPERATURE_OUT_OF_RANGE – if the temperature passed is outside acceptable ranges.

2.1.4.54 GetPortToolFixTemp

Description: Returns the current selected fix temperature for the tool model at the indicated port.

Prototype: Public **function** GetPortToolFixTemp(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**) as
Ctemperature

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

Returns: The current selected fix temperature for the tool model at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT**_IN_RANGE** – if the port identifier is not in range in the current station model number of ports.

2.1.4.55 SetPortToolFixTemp

Description: Sets the selected fix temperature for the tool model at the indicated port.

Prototypes: Public **sub** SetPortToolFixTemp(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**,
ByVal newTemp as **Ctemperature**)

Public **sub** SetPortToolFixTemp(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**,
ByVal newOnOff as **cOnOff**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

newTemp – The desired new fix temperature.

newOnOff – Use **cOnOff._OFF** to deactivate fix temperature.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

TEMPERATURE_OUT_OF_RANGE – if the temperature passed is outside acceptable ranges.

2.1.4.56 GetPortToolSelectedTempLevels

Description: Returns the current selected temperature levels for the tool model at the indicated port.

Prototype: Public **function** GetPortToolSelectedTempLevels(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**)
as **cToolTemperatureLevels**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

Returns: The current selected temperature levels for the tool model at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.57 SetPortToolSelectedTempLevels

Description: Sets the temperature levels for the tool model at the indicated port.

Prototype: Public **sub** SetPortToolSelectedTempLevels(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**,
ByVal levels as **cToolTemperatureLevels**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

levels – The desired new temperature levels.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.58 GetPortToolTempLevel

Description: Returns the current temperature of the indicated level for the tool model at indicated port.

Prototype: Public **function** GetPortToolTempLevel(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**,
ByVal level as **cToolTemperatureLevels**) as **Ctemperature**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

level – The desired temperature level.

Returns: The current temperature of the indicated level for the tool model at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.59 SetPortToolTempLevel

Description: Sets the temperature of the indicated level for the tool model at the indicated port.

Prototype: Public **sub** SetPortToolTempLevel(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**,
ByVal level as **cToolTemperatureLevels**,
ByVal newTemp as **Ctemperature**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

level – The desired temperature level.

newTemp – The desired new temperature for the level.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

TEMPERATURE_OUT_OF_RANGE – if the temperature passed is outside acceptable ranges.

2.1.4.60 GetPortToolSleepDelay

Description: Returns the current sleep delay time for the tool model at the indicated port.

Prototype: Public **function** GetPortToolSleepDelay(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**) as
cToolTimeSleep

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

Returns: The current sleep delay time for the tool model at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.61 SetPortToolSleepDelay

Description: Sets the sleep delay time for the tool model at the indicated port.

Prototype: Public **sub** SetPortToolSleepDelay(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**,
ByVal delay as **cToolTimeSleep**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

delay – The desired sleep delay time.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.62 GetPortToolSleepTemp

Description: Returns the current sleep temperature for the tool model at the indicated port.

Prototype: Public **function** GetPortToolSleepTemp(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**) as
Ctemperature

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

Returns: The current sleep temperature for the tool model at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.63 SetPortToolSleepTemp

Description: Sets the sleep temperature for the tool model at the indicated port.

Prototype: Public **sub** SetPortToolSleepTemp(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**,
ByVal newTemp as **Ctemperature**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

newTemp – The desired sleep temperature.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT**_IN_RANGE** – if the port identifier is not in range in the current station model number of ports.

TEMPERATURE_OUT_OF_RANGE – if the temperature passed is outside acceptable ranges.

2.1.4.64 GetPortToolHibernationDelay

Description: Returns the current hibernation delay time for the tool model at the indicated port.

Prototype: Public **function** GetPortToolHibernationDelay(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**) as
cToolTimeSleep

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

Returns: The current hibernation delay time for the tool model at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT**_IN_RANGE** – if the port identifier is not in range in the current station model number of ports.

2.1.4.65 SetPortToolHibernationDelay

Description: Sets the hibernation delay time for the tool model at the indicated port.

Prototype: Public **sub** SetPortToolHibernationDelay(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**,
ByVal delay as **cToolTimeSleep**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

delay – The desired hibernation delay time.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.66 GetPortToolAdjustTemp

Description: Returns the current adjust temperature for the tool model at the indicated port.

Prototype: Public **function** GetPortToolAdjustTemp(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**) as
Ctemperature

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

Returns: The current adjust temperature for the tool model at the indicated port.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

2.1.4.67 SetPortToolAdjustTemp

Description: Sets the adjust temperature for the tool model at the indicated port.

Prototype: Public **sub** SetPortToolAdjustTemp(ByVal stationID as **Ulong**,
ByVal port as **cPort**,
ByVal tool as **cGenericStationTools**,
ByVal newTemp as **Ctemperature**)

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

port – The identifier of the port where the desired tool is.

tool – The desired tool model.

newTemp – The desired adjust temperature.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

PORT_NOT_IN_RANGE – if the port identifier is not in range in the current station model number of ports.

TEMPERATURE_OUT_OF_RANGE – if the temperature passed is outside acceptable ranges.

2.1.4.68 SetContinuousMode

Description: Activates or deactivates the continuous mode. When the continuous mode is active the station sends port data automatically at the indicated speed (actually it's a period). The data sent is internally stored in a FIFO queue that can be accessed using GetContinuousModeNextData(). The maximum length of data that the queue stores is indicated in the constant:

CONTINUOUS_MODE_QUEUE_MAX_LENGTH

The port data is composed basically of the current port tool tip temperature and power consumption, it also contains a sequence number used to know the order of the transmissions and possible data transmission lost.

The desired ports to activate can be from 1 to 4 (depending on the station model) and must be indicated to the function.

Prototype: Public **sub** SetContinuousMode(ByVal stationID as **Ulong**,
ByVal speed as **cSpeedContinuousMode**,
Optional ByVal portA as **cPort** = cPort.NO_PORT,
Optional ByVal portB as **cPort** = cPort.NO_PORT,
Optional ByVal portC as **cPort** = cPort.NO_PORT,
Optional ByVal portD as **cPort** = cPort.NO_PORT**)**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

speed – The desired speed (period) value for the continuous mode.

portA – The first desired port where to activate the continuous mode.

portB – The second desired port where to activate the continuous mode.

portC – The third desired port where to activate the continuous mode.

portD – The fourth desired port where to activate the continuous mode.

Returns: Nothing.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

CONTINUOUS_MODE_ON_WITHOUT_PORTS – if no ports selected when continuous mode is active.

2.1.4.69 GetContinuousMode

Description: Returns the current continuous mode status. Indicates the current speed (period) and the active ports in a special structure.

Prototype: Public **function** GetContinuousMode(ByVal stationID as **Ulong)** as **tContinuousModeStatus**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The current continuous mode status of the station.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.70 GetContinuousModeDataCount

Description: Returns the current continuous mode pending data in the internal queue. The data in the queue is extracted by calling GetContinuousModeNextData().

Prototype: Public **function** GetContinuousModeDataCount(ByVal stationID as **Ulong)** as **Integer**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The current continuous mode internal queue pending data to extract.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.71 GetContinuousModeNextData

Description: Returns the next data in the continuous mode queue and removes it from the queue.

Prototype: Public **function** GetContinuousModeNextData(ByVal stationID as **Ulong)** as **tContinuousModeData**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: The next data in the continuous mode queue.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.1.4.72 SetTransaction

Description: *The library will raise a TransactionFinished(transactionID) event when receive the response from the station. You can use this function to know that all previous functions have been processed by the station.*

Prototype: Public **function** SetTransaction(ByVal stationID as **Ulong)** as **UInteger**

Parameters: **stationID** – The identifier of the desired station, this identifier is obtained by getting the "NewStationConnected" event.

Returns: A transaction number that will be received within the *TransactionFinished event*.

Errors: **STATION_ID_NOT_FOUND** – if the station identifier is not found in the internal list of connected stations.

2.2 Ctemperature Class

This is a tool class used to represent temperatures. Some of the API methods use this class in order to pass and receive temperature parameters.

This class stores the temperature in UTI units and is designed to easily set and get the temperature in Fahrenheit and in Celsius degrees.

Following are the methods and properties of this class:

1. **UTI**:

   Description: Stores and retrieves the temperature value in UTI.

   Prototype: Public **Property** UTI as **Integer**

2. **ToRoundCelsius**

   Description: Returns the temperature in Celsius degrees in steps of five.

   Prototype: Public **Function** ToRoundCelsius() as **Integer**

3. **ToCelsius**

   Description: Returns the temperature in Celsius degrees.

   Prototype: Public **Function** ToCelsius() as **Integer**

4. **ToRoundFahrenheit**

   Description: Returns the temperature in Fahrenheit degrees in steps of ten.

   Prototype: Public **Function** ToRoundFahrenheit() as **Integer**

5. **ToFahrenheit**

   Description: Returns the temperature in Fahrenheit degrees.

   Prototype: Public **Function** ToFahrenheit() as **Integer**

6. **InCelsius**

   Description: Sets the temperature in Celsius degrees.

   Prototype: Public **Sub** InCelsius(ByVal value as **Integer**)

   Parameters: **value** – The desired temperature in Celsius

7. **InFahrenheit**

   Description: Sets the temperature in Fahrenheit degrees.

   Prototype: Public **Sub** InFahrenheit(ByVal value as **Integer**)

   Parameters: **value** – The desired temperature in Fahrenheit

8. **ToCelsiusToAdjust**

   Description: Returns the temperature in Celsius degrees of an adjust temperature.

   Prototype: Public **Function** ToCelsiusToAjust() as **Integer**

9. **ToFahrenheitToAdjust**

   Description: Returns the temperature in Fahrenheit degrees of an adjust temperature.

   Prototype: Public **Function** ToFahrenheitToAdjust() as **Integer**

10. InCelsiusToAdjust

Description: Sets the temperature in Celsius degrees of an adjust temperature.

Prototype: Public **Sub** InCelsiusToAdjust(ByVal value as **Integer**)

Parameters: **value** – The desired temperature in Celsius

11. InFahrenheitToAdjust

Description: Sets the temperature in Fahrenheit degrees of an adjust temperature.

Prototype: Public **Sub** InFahrenheitToAdjust(ByVal value as **Integer**)

Parameters: **value** – The desired temperature in Fahrenheit

12. **isValid**

Description: Indicates if the temperature value is in range. A temperature that is not valid should not be accepted as the temperature of the corresponding parameter.

Prototype: Public **Function** isValid() as **Boolean**

2.3 Cerror Class

This is the class for the user errors. User errors refer to bad parameter values passed to the API methods. When these methods detect an error throw the "UserError" event and pass the error as an object of this class.

Following are the methods of this class:

1. **cErrorCodes**:

   Description: List of constants that are the possible user error codes.

2. **GetMsg**

   Description: Returns the user error message.

   Prototype: Public **Function** GetMsg() as **String**

3. **GetCode**

   Description: Returns the user error code.

   Prototype: Public **Function** GetCode() as **cErrorCodes**

4. **show**

   Description: Shows a MsgBox with error code and the error message.

   Prototype: Public **Sub** show**()**

2.4 History

2.4.1 Version 1.66.76.0

- Added the following methods to Ctemperature class:
  - ToCelsiusToAdjust
  - ToFahrenheitToAdjust
  - InCelsiusToAdjust
  - InFahrenheitToAdjust
- Removed GetStationPowerLimit and SetStationPowerLimit functions.

2.4.2 Version 1.66.75.1

- Added SetTransaction function and TransactionFinished event

2.4.3 Version 1.66.74.1

- Added GetPortToolStandStatus function
- Changed DefaultStationParametters to DefaultStationParameters
- Added optional paramater when creating the API object: Stations Initial Control Mode

MORE INFORMATION
www.jbctools.com