

# BINARY SPACE

RELIABLE SPACE SYSTEMS

## SatView™ Data I/O Device Driver Development

All information is subject to change without notice and does not represent a commitment on the part of **BINARY SPACE**.  
Release 1.05 (May 2015)

### Table of Contents

1. Introduction
2. Interface Architecture
3. Abstract Interface Classes
4. Software Prerequisites
5. Samples

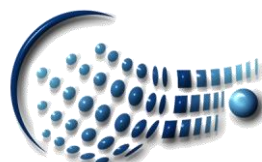
### Appendix

### Table of Figures

Figure	Description
Figure 2.1.	SatView™ Interfaces (exposed to third-party developers)

### Document Change Log

Issue	Revision	Date	Affected	Reason for change
1	1	March 2011	All	New document
1	2	November 2012	3.1.2.	Added ' <b>IsLocal ()</b> CONST' function
1	3	May 2013	3.1.2.	Added ' <i>ReadOnly</i> ' argument to the ' <b>Configure (...)</b> ' function
1	4	March 2014	3.1.2.	Added ' <i>nAllowedTypes</i> ' argument to the ' <b>Configure (...)</b> ' function
1	5	May 2015	3.1.2.	Modified the prototypes for the ' <b>Initialize (...)</b> ', ' <b>Send (...)</b> ' and ' <b>Receive (...)</b> ' procedures



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)

# BINARY SPACE

## RELIABLE SPACE SYSTEMS

### 1. Introduction

SatView™ exposes important interfaces to third-party developers which allow an effective customization of the product.

The following areas can be subject of such an extension:

- **Data I/O**

Writing data I/O device drivers can make SatView™ compatible to any available TM/TC front-end equipment.

- **Database**

Multiple database standards can be supported by developing database drivers which make a migration of SatView™ to new missions possible in a flexible way.

- **Automation**

Offers an interface to all automation services of SatView™; ideal for the integration into complex and highly automated ground segments.

This document focuses on the development of data I/O device drivers.

### 2. Interface Architecture

By exposing important interfaces to third-party developers customers can adapt SatView™ to their current environment in a flexible way:



Figure 2.1. – SatView™ Interfaces



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)

The work includes the development of an interface driver in form of a *Dynamic Link Library* (DLL) for the Data I/O and the database; the automation is performed through XML-formatted requests via TCP/IP.

### 3. Abstract Interface Classes

The interface specification is implemented via C++ abstract base classes exposing the interface through virtual member functions. This approach has the advantage that SatView™ does not have to care about any device specifics because they are encapsulated by the DLL.

#### 3.1. The CIODevice Abstract Base Class

Any data I/O device driver to be written for SatView™ must be derived from the base class called **CIODevice**.

This chapter hereinafter explains the various member functions, types and flags involved with this class.

##### 3.1.1. Non-virtual Functions

The functions listed here do not need to be implemented in any derived class. They provide the basic functionality of the data I/O device driver.

**BOOL Open (LPCTSTR pszName, LPCTSTR pszDeviceBrand)**

Opens the data I/O device driver.

#### Parameters:

*pszName*

Specifies the location of the data I/O device driver DLL.

*pszDeviceBrand*

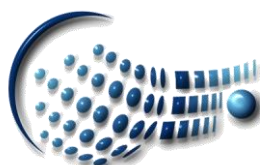
Identifies the brand of the device to be opened.

#### ☑ Note:

Some drivers support multiple types of devices within the same DLL and this argument specifies which one to open.

#### Return Value:

Indicates if the data I/O device driver DLL could be loaded and the specified device was opened successfully.



# BINARY SPACE

RELIABLE SPACE SYSTEMS

## BOOL **Open ()**

Re-opens the data I/O device driver.

### Note:

The function can only be used if the data I/O device driver was successfully opened before.

### **Return Value:**

Indicates if the data I/O device driver was re-opened successfully.

## BOOL **IsOpen ()** CONST

Checks if the data I/O device driver is already open.

### **Return Value:**

Returns TRUE if the data I/O device driver is currently open; FALSE if not.

## VOID **Close ()**

Closes the data I/O device driver.

### **Return Value:**

None

## VOID **SetDeviceBrand (LPCTSTR *pszDeviceBrand*)**

Sets the brand name for the device currently open.

### Note:

A data I/O device driver can support multiple devices of a certain brand differentiated by a unique name.

### **Parameters:**

*pszDeviceBrand*

Contains the brand name to be set for the device.

### **Return Value:**

None

## CString **GetDeviceBrand ()** CONST

Gets the brand name for the current device.

### **Return Value:**

Returns the brand name of the device as a string.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)

# BINARY SPACE

RELIABLE SPACE SYSTEMS

**VOID SetDeviceName (LPCTSTR *pszDeviceName*)**

Sets the name for the device currently open.

 **Note:**

A data I/O device driver can support multiple devices of a certain brand.

**Parameters:**

*pszDeviceName*

Contains the name to be set for the currently open device.

**Return Value:**

None

**CString GetDeviceName () CONST**

Gets the name for the current device.

**Return Value:**

Returns the device name as a string.

**VOID SetDeviceModule (LPCTSTR *pszDeviceModule*)**

Sets the file path of the data I/O device driver DLL.

**Parameters:**

*pszDeviceModule*

Contains the path name of the DLL.

**Return Value:**

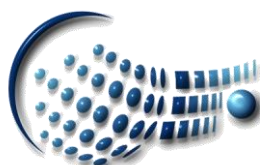
None

**CString GetDeviceModule () CONST**

Gets the file path of the data I/O device driver DLL.

**Return Value:**

Returns the path name of the DLL as a string.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)

# BINARY SPACE

RELIABLE SPACE SYSTEMS

```
VOID SetDeviceProperties(CONST CByteArray &nDeviceData)
```

Associates device specific data with the device currently open.

**Parameters:**

*nDeviceData*

Contains the device specific serialized data.

**Return Value:**

None

```
INT GetDeviceProperties(CByteArray &nDeviceData) CONST
```

Gets the device specific data of the current device.

**Parameters:**

*nDeviceData*

Contains the device specific serialized data.

**Return Value:**

Returns the size (in bytes) of the returned data.

```
VOID SetDeviceSecurity(CONST CStringArray &szUsers,  
CONST CUIntArray &nCodes,  
CONST CTimeKeyArray &tStartTimes,  
CONST CTimeKeyArray &tStopTimes)
```

Specifies the security policy for the device currently open.

**Parameters:**

*szUsers*

Contains a list of clients for which an access policy is specified.

 **Note:**

Each of these entries must consist of the client display name and IP address separated by an end-of-line character.

*nCodes*

Specifies the access policy for each of the clients:



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)

# BINARY SPACE

RELIABLE SPACE SYSTEMS

Security Policy Table:

Policy	Description
IODEVICE_SECURITY_UNLIMITEDACCESS	A client receives unlimited access to the services of the device.
IODEVICE_SECURITY_RESTRICTEDACCESS	Any access to the device is restricted to the time between <i>tStartTime</i> and <i>tStopTime</i> for the client.
IODEVICE_SECURITY_DENIEDACCESS	The client cannot make use of the device's services.
IODEVICE_SECURITY_AUDITACCESS	Requests an event message to be issued whenever a client logs-in or logs-out. <input checked="" type="checkbox"/> Note: This flag can be combined with one of the above ones.

*tStartTime*

Contains a list of access start times.

Note:

This argument is only applicable when the IODEVICE\_SECURITY\_RESTRICTEDACCESS policy is specified. Use 0 in all other cases.

*tStopTimes*

Contains a list of access stop times.

Note:

This argument is only applicable when the IODEVICE\_SECURITY\_RESTRICTEDACCESS policy is specified. Use 0 in all other cases.

**Return Value:**

None



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: info@binary-space.com, Web: www.binary-space.com

# BINARY SPACE

RELIABLE SPACE SYSTEMS

```
INT GetDeviceSecurity(CStringArray &szUsers,  
    CUIntArray &nCodes,  
    CTimeKeyArray &tStartTimes,  
    CTimeKeyArray &tStopTimes) CONST
```

Retrieves the security policy for the current device.

## Parameters:

*szUsers*

Contains a list of clients for which an access policy is specified.

### ☑ Notes:

- Each of these entries consists of the client display name and IP address separated by an end-of-line character.
- An empty entry combined with the `IODEVICE_SECURITY_UNLIMITEDACCESS` policy indicates that any client not part of the list receives full access to the device.

*nCodes*

Specifies the access policy for each of the clients.

### ☑ Notes:

See above for all options available.

*tStartTimes*

Contains a list of access start times.

### ☑ Note:

This argument is only applicable when the `IODEVICE_SECURITY_RESTRICTEDACCESS` policy is specified. Use 0 in all other cases.

*tStopTimes*

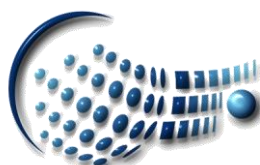
Contains a list of access stop times.

### ☑ Note:

This argument is only applicable when the `IODEVICE_SECURITY_RESTRICTEDACCESS` policy is specified. Use 0 in all other cases.

## Return Value:

Returns the number of entries in the device security list.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)



# BINARY SPACE

RELIABLE SPACE SYSTEMS

```
BOOL MonitorDeviceLogins (UINT nMask=0,  
IODEVICELOGINSPROCEDURE pLoginProcedure=NULL,  
LPVOID pData=NULL)
```

Installs a callback procedure for all client activities on the current device.

## Parameters:

*nMask*

Specifies the type of activity to be monitored:

Activity	Description
IODEVICE_LOGIN_SUCCESS	The supplied callback procedure will be called whenever a client logs-in.
IODEVICE_LOGIN_FAILURE	All denied log-ins will be notified.
IODEVICE_LOGOUT_SUCCESS	A call to the supplied procedure will always take place when a client logs-out.
IODEVICE_LOGINOUT_AUDIT	Indicates that an event message should be issued.
0	Disables the client activity monitoring.

## Note:

All activity flags (except 0) can be combined with each other.

*pLoginProcedure*

Specifies the callback procedure to be called for all specified client activities. The supplied procedure must conform to the following syntax:

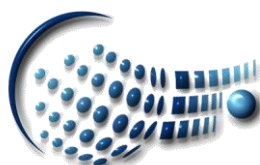
```
VOID (CALLBACK *) (UINT nFlags, LPCTSTR pszClient, LPVOID pData);
```

*pData*

Contains a pointer to the **CIODevice**-derived class calling the *pLoginProcedure* procedure.

## Return Value:

Returns TRUE if the client activity can be monitored; FALSE if not.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: info@binary-space.com, Web: www.binary-space.com

# BINARY SPACE

RELIABLE SPACE SYSTEMS



```
BOOL MonitorDeviceDataStream(UINT nMask=0,  
CONST CTimeSpan &tInterval=0,  
IODEVICEDATASTREAMPROCEDURE pDataStreamProcedure=NULL,  
LPVOID pData=NULL)
```

Installs a callback procedure for all data stream events on the current device.

## Parameters:

*nMask*

Specifies the type of data stream events to be monitored:

Event	Description
IODEVICE_DATASTREAM_DROP	The supplied callback procedure will be called whenever the device detects a drop in the data stream.  <b>Note:</b> A gap in the data stream is considered as such when no data is received for a period longer than specified by <i>tInterval</i> .
IODEVICE_DATASTREAM_RECONNECTED	A notification will be performed when the data stream is resumed after a drop.
IODEVICE_DATASTREAM_DELAYED	Clients of the data I/O device may temporarily be overloaded; a situation which can be signaled if desired.  <b>Note:</b> A delay is defined by the difference between the data stream time and the system time and specified by <i>tInterval</i> .
IODEVICE_DATASTREAM_BACKINTIME	Requests a notification whenever a period of delayed data delivery is terminated.
0	Disables the data stream event monitoring.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)

# BINARY SPACE

## RELIABLE SPACE SYSTEMS

*tInterval*

Specifies a time interval (in seconds).

☑ Note:

The argument must be set to a value > 0 s when the `IODEVICE_DATASTREAM_DROP` or `IODEVICE_DATASTREAM_DELAYED` notification events are requested. Call the `MonitorDeviceDataStream` procedure twice if both callbacks should be installed.

*pDataStreamProcedure*

Specifies the callback procedure to be called for all specified data stream events. The supplied procedure must conform to the following syntax:

```
VOID (CALLBACK *) (UINT nFlags, TIMEKEY tInterval, LPVOID pData)
```

*pData*

Contains a pointer to the `CIODevice`-derived class calling the `pDataStreamProcedure` procedure.

### Return Value:

Returns TRUE if the data stream events can be monitored; FALSE if not.

```
BOOL MonitorDeviceDataBuffers (HANDLE &hInBuffer,  
HANDLE &hOutBuffer)
```

Sets the event handles for the input and output buffers of the device.

☑ Note:

The supplied handles get signaled when data is available in the corresponding buffers. Using handles is strongly recommended because it is more efficient than performing polling.

### Return Value:

Returns TRUE if the handles could be set; FALSE if not.

```
BOOL ShowDeviceStatus (LPCTSTR pszStatus, UINT nStatus)
```

Shows the current status of the device.

### Parameters:

*pszStatus*

Specifies the current status of the device.

☑ Note:

This status indication text is device specific should not be longer than 16 characters.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)

# BINARY SPACE

## RELIABLE SPACE SYSTEMS

*nStatus*

Specifies the current status of the device in a numerical form:

Status	Description
IODEVICE_STATUS_GOOD	The device is in a healthy state.
IODEVICE_STATUS_WARNING	A warning condition occurred.
IODEVICE_STATUS_ERROR	A major failure has been encountered.
IODEVICE_STATUS_CONNECTED	The device is physically connected.
IODEVICE_STATUS_NOTCONNECTED	No connection has been established or is disconnected.
IODEVICE_STATUS_ONLINE	The device is scheduled to be in-service.
IODEVICE_STATUS_OFFLINE	No services should be provided by the device.

☑ Note:

- The numerical status should reflect the one indicated by *pszStatus* as close as possible
- The status IODEVICE\_STATUS\_GOOD is mutual exclusive with IODEVICE\_STATUS\_WARNING and IODEVICE\_STATUS\_ERROR
- IODEVICE\_STATUS\_CONNECTED cannot be used with IODEVICE\_STATUS\_NOTCONNECTED
- IODEVICE\_STATUS\_ONLINE and IODEVICE\_STATUS\_OFFLINE cannot be used together

### Return Value:

Returns TRUE if the device status could be set; FALSE if not.

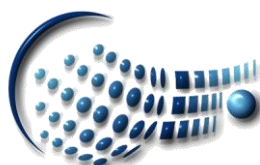
```
BOOL ShowDeviceMessage (LPCTSTR pszMessage,  
UINT nMessageType=IODEVICE_MESSAGE_TYPE_ERROR)
```

Shows a device specific message.

### Parameters:

*pszMessage*

Contains the message.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)

# BINARY SPACE

## RELIABLE SPACE SYSTEMS

*nMessageType*

Specifies the severity of the message:

Message Severity Table:

Severity	Description
IODEVICE_MESSAGETYPE_INFORMATIONAL	The message has an informational character.
IODEVICE_MESSAGETYPE_SUCCESS	A successful event is reported by the message.
IODEVICE_MESSAGETYPE_WARNING	The message contains a warning.
IODEVICE_MESSAGETYPE_ERROR	An error is reported by the message.

**Return Value:**

Returns TRUE if the message could be shown; FALSE if not.

```
CIODevice *GetIODevice() CONST
```

Returns a pointer to the device currently open.

**Return Value:**

Returns the pointer to the **CIODevice**-derived class.

### 3.1.2. Virtual Functions

The virtual functions make up the actual interface between the physical device and SatView™. They have to be implemented.

```
virtual BOOL Start()
```

Starts the operation of the I/O device driver.

**Return Value:**

Returns TRUE if the I/O device driver could be started successfully; FALSE if not.

```
virtual BOOL Suspend()
```

Suspends the operation of the I/O device driver.

**Return Value:**

Returns TRUE if the I/O device driver could be suspended; FALSE if not.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: info@binary-space.com, Web: www.binary-space.com

# BINARY SPACE

## RELIABLE SPACE SYSTEMS

```
virtual BOOL IsSuspended () CONST
```

Checks if the I/O device driver is in a suspended state.

**Return Value:**

Returns TRUE if the I/O device driver is in a suspended state; FALSE if not.

```
virtual BOOL Resume ()
```

Resumes the operation of the I/O device driver when in a suspended state.

**Return Value:**

Returns TRUE if the I/O device driver could be resumed successfully; FALSE if not.

```
virtual BOOL Stop ()
```

Stops the operation of the I/O device driver.

**Return Value:**

Returns TRUE if the I/O device driver could be stopped; FALSE if not.

```
virtual BOOL Configure (CWnd *pParentWnd=NULL, UINT  
nAllowedTypes=IODEVICE_TYPE_SERVER |  
IODEVICE_TYPE_CLIENT, BOOL bReadOnly=FALSE)
```

Configures the I/O device driver.

**Parameters:**

*pParentWnd*

Specifies the parental window of the I/O device driver configuration dialog box.

**Note:**

Specifying NULL as argument value uses the desktop as parent.

*nAllowedTypes*

Specifies the type of the device to be configured.

**Note:**

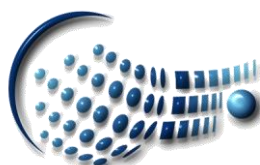
Specify either the IODEVICE\_TYPE\_CLIENT flag or both.

*bReadOnly*

Specifies if the I/O device driver configuration dialog box is to be used in read-only (browse) mode.

**Return Value:**

Returns TRUE if the I/O device driver configuration was completed by pushing the OK button; FALSE if not.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: info@binary-space.com, Web: www.binary-space.com

# BINARY SPACE

## RELIABLE SPACE SYSTEMS

```
virtual BOOL Initialize(UINT nFlags=IODEVICE_TYPE_CLIENT |
    IODEVICE_SERVICE_TM | IODEVICE_DATASERVICE_SOURCEPACKETS,
    CONST CTimeSpan &tDataDrop=0,
    CONST CTimeSpan &tDataDelay=0,
    IODEVICESTATUSPROCEDURE pStatusProcedure=NULL,
    IODEVICEDATASTATUSPROCEDURE pDataStatusProcedure=NULL,
    IODEVICEMESSAGEPROCEDURE pMessageProcedure=NULL)
```

Initializes the I/O device driver and prepares it for operation.

### Parameters:

*nFlags*

Specifies the type of I/O device driver requested including the desired data service:

Flags	Description
IODEVICE_TYPE_SERVER	The I/O device driver should act as a server. ☑ Note: This mode may not be implemented by some drivers.
IODEVICE_TYPE_CLIENT	The I/O device driver should implement client functionality.
IODEVICE_SERVICE_TM	The service is requested to handle telemetry data.
IODEVICE_SERVICE_TC	The service is requested to handle telecommand data.
IODEVICE_DATASERVICE_RAW	The data transferred should be in a raw (serialized) format.
IODEVICE_DATASERVICE_TRANSFERFRAMES	Transfer frames are requested as transmission format. ☑ Note: See the <b>CTMTransferFrame</b> & <b>CTCTransferFrame</b> classes for more information.





In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: info@binary-space.com, Web: www.binary-space.com

# BINARY SPACE

## RELIABLE SPACE SYSTEMS

### *I/O Device Driver Capability Flags (continued)*

IODEVICE_DATASERVICE_SOURCEPACKETS	The data exchange should take place on the basis of source packets.  Note: See the <b>CTMUnit</b> & <b>CTCUnit</b> classes for more information.
IODEVICE_PROTOCOL_PFLP	The <i>Packet Front-End Link Protocol</i> (PFLP) should be used for the data exchange.  Note: If this flag is not specified, any other protocol implemented by the I/O device driver will be used.

 Note:

- IODEVICE\_TYPE\_SERVER and IODEVICE\_TYPE\_CLIENT are mutual exclusive flags
- The flags IODEVICE\_SERVICE\_TM and IODEVICE\_SERVICE\_TC cannot be used together
- Specify either IODEVICE\_DATASERVICE\_RAW, IODEVICE\_DATASERVICE\_TRANSFERFRAMES or IODEVICE\_DATASERVICE\_SOURCEPACKETS as the requested data service
- The IODEVICE\_PROTOCOL\_PFLP flag is required only if the use of the *Packet Front-End Protocol* (PFLP) is mandatory

#### *tDataDrop*

Specifies the minimum time interval with no data transfer activity considered to be a drop.

 Note:

This argument is used when the IODEVICE\_SERVICE\_TM flag is specified; it may be 0 for any other service.

#### *tDataDelay*

Specifies the maximum time delay allowed (in seconds) for out-of-band data (e.g. high-priority data).

 Note:

This argument is used when the IODEVICE\_SERVICE\_TM flag is specified; it may be 0 for any other service.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: info@binary-space.com, Web: www.binary-space.com



## *pStatusProcedure*

Supplies the address of a device status indication procedure to be called when the status changes.

The supplied procedure must conform to the following syntax:

```
VOID (CALLBACK *) (LPCTSTR pszStatus,UINT nStatus)
```

### Note:

Consult the **ShowDeviceStatus** function for the possible argument values.

## *pDataStatusProcedure*

Specifies the callback procedure to be called for all data status events.

The supplied procedure must conform to the following syntax:

```
VOID (CALLBACK *) (INT nDataID,UINT nStatus,BOOL bSuccess)
```

The variable *nDataID* identifies the data to which the provided status applies. It is identical to the return value of the **Send(...)** procedure.

The values for *nStatus* can be (a combination) of:

Status	Description
IODEVICE_DATASTATUS_SENT	The data identified by <i>nDataID</i> was sent successfully to the destination device (if <i>bSuccess</i> =TRUE).
IODEVICE_DATASTATUS_PROCESSED	The data identified by <i>nDataID</i> was processed successfully at the destination device (if <i>bSuccess</i> =TRUE).

## *pMessageProcedure*

Supplies the address of a procedure handling (i.e. showing) the device messages issued by the I/O device driver.

The supplied procedure must conform to the following syntax:

```
VOID (CALLBACK *) (LPCTSTR pszMessage,UINT nMessageType)
```

### Note:

Consult the **ShowDeviceMessage** function for the possible argument values.

### **Return Value:**

Returns TRUE if the I/O device driver could be initialized with the requested flags; FALSE if not.



# BINARY SPACE

## RELIABLE SPACE SYSTEMS

```
virtual BOOL Update (CONST CIODevice *pDevice, BOOL &bRestart)
```

Updates the I/O device driver with the configuration of another one and returns an indication if a restart is required to apply the changes.

### Parameters:

*pDevice*

Contains the pointer to another open I/O device driver those configuration should be copied.

*bRestart*

In: Directs the procedure to restart the I/O device driver automatically (when required) if set to TRUE.

Out: Indicates if a restart is required to make any changes apply (when FALSE was specified as input value) or if a restart has taken place (when TRUE on input)

### Return Value:

Returns TRUE if the I/O device driver configuration was updated successfully; FALSE if not.

```
virtual INT Send (CONST CByteArray &nData)
virtual INT Send (CONST CTMTransferUnit &cTMTransferUnit)
virtual INT Send (CONST CTCTransferUnit &cTCTransferUnit)
virtual INT Send (CONST CTMUnit &cTMUnit)
virtual INT Send (CONST CTCUnit &cTCUnit)
```

Sends the supplied data via the I/O device driver to the physical device in a non-blocking way.

### ☑ Note:

The **Send** procedure used to transmit data must correspond with the flags specified during the **Initialize** call i.e. the flag `IODEVICE_DATASERVICE_RAW` implies the use of the first overloaded procedure, `IODEVICE_DATASERVICE_TRANSFERFRAMES` requires either the second or third one to be used (depending on `IODEVICE_SERVICE_TM` or `IODEVICE_SERVICE_TC`). The same applies for the latter two procedures that can only be used when the `IODEVICE_DATASERVICE_SOURCEPACKETS` flag was specified previously.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)

# BINARY SPACE

## RELIABLE SPACE SYSTEMS

### Parameters:

*nData*

Contains the data to be transmitted in a serialized format.

*cTMTransferUnit*

*cTCTransferUnit*

Supplies the data as a transfer frame.

*cTMUnit*

*cTCUnit*

Supplies the data as a source packet.

### Return Value:

Returns a data identification number  $\geq 0$  if the I/O device driver could add the data to the output buffer; -1 if not. The returned number is identical to the one provided by the *nDataID* argument of the callback procedure *pDataStatusProcedure* specified by

**Initialize(...)**.

```
virtual BOOL Receive(CByteArray &nData)
virtual BOOL Receive(CTMTransferUnit &cTMTransferUnit)
virtual BOOL Receive(CTCTransferUnit &cTCTransferUnit)
virtual BOOL Receive(CTMUnit &cTMUnit)
virtual BOOL Receive(CTCUnit &cTCUnit)
```

Retrieves data from the I/O device driver input buffer in a non-blocking way.

 Note:

The **Receive** procedure used must correspond with the flags specified during the **Initialize** call (see above).

### Parameters:

*nData*

Returns the data in a serialized format.

*cTMTransferUnit*

*cTCTransferUnit*

Retrieves the data as a transfer frame.

*cTMUnit*

*cTCUnit*

Retrieves the data as a source packet.

### Return Value:

Returns TRUE if the data could be retrieved in the requested format; FALSE if not.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)

# BINARY SPACE

## RELIABLE SPACE SYSTEMS

```
virtual BOOL Clear(BOOL bAll=TRUE)
```

Clears the input or output buffers of the I/O device driver.

**Parameters:**

*bAll*

Indicates if all buffers should be cleared or not.

If FALSE is specified, only the output buffers are cleared when the I/O device driver acts as a server or the input buffers when operating as a client.

**Return Value:**

Returns TRUE if the specified buffers could be cleared; FALSE if not.

```
virtual BOOL SetStatus(LPCTSTR pszStatus)
```

Sets the status of the current device.

**Parameters:**

*pszStatus*

Specifies the current status of the device.

 **Note:**

This status indication text is device specific should not be longer than 16 characters.

**Return Value:**

Returns TRUE if the device status could be set; FALSE if not.

```
virtual CString GetStatus() CONST
```

Gets the current status of the device.

**Return Value:**

Returns the current device status as a string.

```
virtual BOOL SetStatusIndication(UINT nStatus)
```

Sets the status of the device in a numerical form.

**Parameters:**

*nStatus*

Specifies the current status of the device in a numerical form.

 **Note:**

- The numerical status should reflect the textual status as close as possible
- See the procedure **ShowDeviceStatus** for all allowed values

**Return Value:**

Returns the TRUE if the device status could be set; FALSE if not.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)

# BINARY SPACE

## RELIABLE SPACE SYSTEMS

```
virtual UINT GetStatusIndication() CONST
```

Gets the status of the device in a numerical form.

### Return Value:

Returns the current device status as a numerical value.

### Note:

See the procedure **ShowDeviceStatus** for all values that may be returned.

```
virtual UINT GetCapabilities() CONST
```

Gets the capabilities of the I/O device driver.

### Note:

This function can be used to choose the way how the I/O device driver should operate. The flags returned are identical to those required by the **Initialize** function.

### Return Value:

Returns the capabilities of the I/O device driver:

Flags	Description
IODEVICE_TYPE_SERVER	The I/O device driver can act as a server.
IODEVICE_TYPE_CLIENT	The I/O device driver can operate as a client.
IODEVICE_SERVICE_TM	Telemetry data can be handled.
IODEVICE_SERVICE_TC	Telecommand data can be handled.
IODEVICE_DATASERVICE_RAW	Any data transfer can take place in a serialized way.
IODEVICE_DATASERVICE_TRANSFERFRAMES	Transfer frames are supported for sending or receiving. <b>Note:</b> See the <b>CTMTransferUnit</b> and <b>CTCTransferUnit</b> classes for more information.
IODEVICE_DATASERVICE_SOURCEPACKETS	The data exchange can take place on the basis of source packets. <b>Note:</b> See the <b>CTMUnit</b> and <b>CTCUnit</b> classes for more information.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: info@binary-space.com, Web: www.binary-space.com

# BINARY SPACE

RELIABLE SPACE SYSTEMS

## *I/O Device Driver Capabilities (continued)*

IODEVICE_PROTOCOL_PFLP	The I/O device driver supports the <i>Packet Front-End Link Protocol (PFLP)</i> .
------------------------	---

```
virtual CString GetDetails() CONST
```

Gets details about the I/O device driver which are displayed as a tool-tip.

### **Return Value:**

Returns additional information about the I/O device driver as a string.

```
virtual INT EnumConnections(CStringArray &szComputers) CONST  
virtual INT EnumConnections(CStringArray &szComputers,  
    CTimeTagArray &tConnectTimes,  
    CTimeTagArray &tLastUseTimes,  
    CDWordArray &cbInBytes,  
    CDWordArray &cbOutBytes) CONST
```

Returns all server/client connections currently handled by the I/O device driver.

### **Note:**

If the I/O device driver is configured to act as a server, this procedure returns all client connections; when operating in client mode it enumerates only the one to the server.

### **Parameters:**

*szComputers*

Contains the name of the connected peer in the format:

**Peer Name**end-of-line**IP Address** (Peer name and IP address separated by a '\n')

*tConnectTimes*

Contains the time when the connection was established.

*tLastUseTimes*

Contains the time when the connection was used for the last time.

*cbInBytes*

Contains the number of KB received on the connection.

*cbOutBytes*

Contains the number of KB sent on the connection.

### **Return Value:**

Returns the number of connections enumerated.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: info@binary-space.com, Web: www.binary-space.com

# BINARY SPACE

## RELIABLE SPACE SYSTEMS

```
virtual BOOL AbortConnection(LPCTSTR pszComputer,CONST CTimeTag  
&tConnectTime)
```

Aborts the connection to the peer with the IP address *pszComputer* by the I/O device driver.

**Parameters:**

*pszComputer*

Contains the IP address of the peer to which the connection should be aborted in a dotted string format (e.g. 195.74.165.216).

*tConnectTime*

Specifies the establishment time of the connection to be aborted.

**Return Value:**

Returns TRUE if the connection was aborted successfully; FALSE if not.

```
virtual BOOL IsLocal () CONST
```

Checks if the I/O device driver is connected to a local device i.e. if the data received is generated locally (e.g. by a simulator).

 **Note:**

This function is intended to be used to check for a simulator data generation device.

**Return Value:**

Returns TRUE if the I/O device driver is connected to a local device; FALSE if not or if the device driver is not open.

```
virtual BOOL Copy (CONST CIODriver *pDevice)
```

Copies the I/O device driver.

**Parameters:**

*pDevice*

Contains a pointer to the I/O device driver to be copied.

**Return Value:**

Returns TRUE if the copy was successful; FALSE if not.

```
virtual BOOL Compare (CONST CIODriver *pDevice) CONST
```

Compares the I/O device driver with another one.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)

# BINARY SPACE

RELIABLE SPACE SYSTEMS

**Parameters:**

*pDevice*

Contains a pointer to the I/O device driver to be compared with.

**Return Value:**

Returns TRUE if both I/O device drivers are identical (including their configuration); FALSE if not.

```
virtual BOOL Map (CByteArray &nInfo) CONST
```

Serializes the I/O device driver.

**Parameters:**

*nInfo*

Contains the I/O device driver data in a serialized format.

**Return Value:**

Returns TRUE if the serialization was successful; FALSE if not.

```
virtual BOOL Unmap (CONST CByteArray &nInfo) CONST
```

De-serializes data previously serialized with the **Map** procedure.

**Parameters:**

*nInfo*

Contains the I/O device driver data in a serialized format.

**Return Value:**

Returns TRUE if the de-serialization was successful; FALSE if not.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)



# BINARY SPACE

RELIABLE SPACE SYSTEMS

## 4. Software Prerequisites

The Microsoft® Visual Studio® 2010 or higher is required as well as the **SatView™ Data I/O Device Driver Development Kit** (DDK) which is available for free for all customers of **BINARY SPACE**.

## 5. Samples

The SatView™ Data I/O DDK contains a complete implementation of an I/O device driver for the ENERTEC TT&C 3801 device.



In der Weid 3, CH-8122 Binz

Tel: +41 44 8877987, Fax: +41 44 8877989, Email: [info@binary-space.com](mailto:info@binary-space.com), Web: [www.binary-space.com](http://www.binary-space.com)