WIN-PAK PRO

Database API Manual

NORTHERN COMPUTERS, INC.

ADEMCO GROUP
INTEGRATED SYSTEMS

TD0093 rev0600

Table of Contents

Introduction

The success of the WIN-PAK access control product and the need to provide access to its database across multiple platforms has resulted in this database Application Program Interface (API). With this API, we have created a set of COM objects that provide the ability to read and write information to the WIN-PAK database. With this first release, the capabilities are limited to the most often used data. Navigation within the object model mimics the navigation common to Microsoft Word, Microsoft Excel, and other tools. You will have access to the following objects:

- **NCIWinPak.AccessLevel:** Cards may have one or more access levels assigned to them. This object contains the Access Level ID and the name assigned to it in WIN-PAK.

- **NCIWinPak.Account:** Grants the user access to cardholders and account specific events/alarms. From this level, you also obtain the note field names and corresponding index names within a card holder record.

- **NCIWinPak.Application:** Grants the user access to all accounts and events/alarms on all accounts and devices.

- **NCIWinPak.Card:** Grants the user access to card specific events and alarms. You can also use this object to assign a card to cardholder in an account.

- **NCIWinPak.CardHolder:** Grants the user access to cards and cardholder specific events and alarms. You can also use this interface to add a cardholder to an account.

- **NCIWinPak.CardHolderCallBackSetup:** Provides the NCIWinPak.Account object with the cardholder information that should be returned and who should get it.

- **NCIWinPak.EventCallBackSetup:** Provides the NCIWinPak.Account, NCIWinPak.Application, NCIWinPak.Card, NCIWinPak.CardHolder, and NCIWinPak.HWIndependentDevice objects with event information to be returned and who should get it.

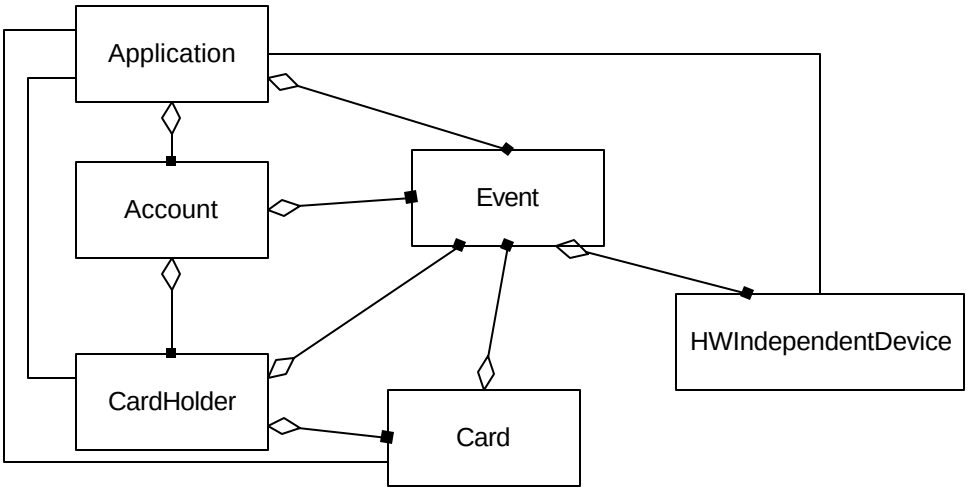
- **NCIWinPak.EventObj:** Contains the details relating to an event/alarm. From this object, you can find the card, cardholder, and device associated with the event. It is possible that no card or cardholder will be associated with an event (example: Forced Entry).

- **NCIWinPak.HWIndependentDevice:** This object contains information about the item that generated an event. For card reads, this object will be used to identify the reader used to read the card.

- **NCIWinPak.Operator:** Allows the COM objects and users of the COM objects to get all the benefits of the WIN-PAK operator level permissions outside of the WIN-PAK user interface. This includes standard WIN-PAK login capabilities.

Object Model

The following diagram depicts the relationships between the objects as well as paths available to navigate between the various objects.



All of the above boxes represent COM objects available from the system. (The diagram uses UML to express the relationships between objects.) A few of the objects not in this diagram, such as AccessLevel, can be obtained through the Application object. The diagram shows how to navigate between the major components of the dbAPI. For example, the one Application object allows navigation to:

- Account (zero or more)
- Event (zero or more)
- A specific CardHolder
- A specific Card
- A specific HWIndependentDevice

Object Specifications

AccessLevel

Properties:

Property Name	VB Type	VC++ Type	Description
ID	Long	long	Uniquely identifies the entity to the application.
Description	String	*bstr*t	The access level's description.

Account

Properties:

Property Name	VB Type	VC++ Type	Description
ID	Long	long	Uniquely identifies the entity to the application.
AccountName	String	BSTR	The account's name.
MiscFields	String()	SAFEARRAY*	You can add up to 10 miscellaneous pieces of data to an account. This is that data.
CardHolderFilterFieldNames	String()	SAFEARRAY*	Returns the template field names that the user can use to filter cardholders.

Methods:

Init

Method Name: Init
VB Signature: (operatorObj As Object)
VC++ Signature: (Idispatch** operatorObj)
Description: If you create the Account object, it is your responsibility to tell it which operator is manipulating the Account. Failure to do so will cause it to throw COM exceptions whenever you try to view or set data within the object. If a NCIWinPak object returns an Account to you, you can assume that the Init has already been called for you.
Parameters: operatorObj: An initialized NCIWinPak.Operator object.

GetCardHolders

Method Name: GetCardHolders
VB Signature: (ByRef callbackSetup as Object)
VC++ Signature: (Idispatch** callbackSetup)
Description: This asynchronous function returns all cardholders related to the account based on the information specified in the callbackSetup argument. callbackSetup must be an instance of a CardHolderCallBackSetup object.
Parameters: callbackSetup: Specifies any search criteria for the cardholder retrieval.

GetCardHolders_sync

Method Name: GetCardHolders_sync
VB Signature: (numItems as Long, cardHolders as Variant)
VC++ Signature: (long* numItems, VARIANT* cardHolders)
Description: Returns the cardholders only when asked. On the first call it returns records starting from the first record up to numItems records. Subsequent calls continue from the next position beyond the last return (i.e. if it returns 1-10 on the first call, call number 2 returns 11-20). This will not return more records than are available in the database. If you ask for 300 records and the database only has two left to return, you will only get two records.
Parameters: callbackSetup: Specifies any search criteria for the cardholder retrieval.

GetEvents

Method Name: GetEvents
VB Signature: (ByRef callbackSetup as Object)
VC++ Signature: (Idispatch** callbackSetup)
Description: This asynchronous function returns all events related to the account based on the information specified in the callbackSetup argument. callbackSetup must be an instance of an EventCallBackSetup object.
Parameters: callbackSetup: Specifies any search criteria for the event retrieval.

GetNumCardHolders

Method Name: GetNumCardHolders
VB Signature: as Long
VC++ Signature: long
Description: Returns the number of cardholders associated with this account. If you want to know the number based on some search settings (ex. FirstName < "M") then you need to call SetAdditionalSearchCriteria_sync first.

SetAdditionalSearchCriteria_sync

Method Name: SetAdditionalSearchCriteria_sync
VB Signature: (inputSearchFields() as String, fieldData() as String, comparisons() as ComparisonType)
VC++ Signature: (SAFEARRAY** inputSearchFields, SAFEARRAY** fieldData, SAFEARRAY** comparisons)
Description: This function sets the cardholder lookup information for a synchronous cardholder callback. That is, the user will have to repeatedly call GetCardHolders_sync until all the card holders are retrieved or until the caller no longer wants more cardholders. The size of all three arrays must be the same, because there is a 1:1:1 correspondence at each index.
Parameters:

- **InputSearchFields:** Array of fields to search on. The field names should come from the CardHolderFilterFieldNames property.
- **FieldData:** Data to perform the search against.
- **Comparisons:** Using the ComparisonType data-type, this tells what type of comparison you want to perform between the inputSearchFields(i) and fieldData(i). See ComparisonType in the enumerated types section for details.

Application

Properties:

Property Name	VB Type	VC++ Type	Description
Operator	Object	_variant_t	Used to identify the currently logged in operator. Common usage dictates that you must login, create an Operator, and then set the Operator to the Application object before executing any methods.

Methods:

GetAccessLevels
Method Name: GetAccessLevels
VB Signature: (accessLevels as Variant)
VC++ Signature: (VARIANT* accessLevels)
Description: Returns all access levels available from the WIN-PAK installation.
Parameters: accessLevels: On return this contains an array of AccessLevel objects containing the access level description and ID.

GetAccounts
Method Name: GetAccounts
VB Signature: (ByRef accounts as Variant)
VC++ Signature: (VARIANT* accounts)
Description: Returns all the accounts registered with this WIN-PAK installation.
Parameters: accounts: Contains all the accounts within this installation.

GetAccountByID
Method Name: GetAccountByID
VB Signature: (ByVal ID as Long, ByRef anAccount as Variant)
VC++ Signature: (long ID, VARIANT* anAccount)
Description: Returns the Account by its ID.
Parameters:

- **ID:** Uniquely identifies the Account to retrieve.
- **anAccount:** On return, this will be set to an Account object.

GetAllDevices
Method Name: GetAllDevices
VB Signature: (devices as Variant)
VC++ Signature: (VARIANT* devices)
Description: Returns all the devices setup on the system.
Parameters: devices: An array of HWIndependentDevice objects.

GetAccessLevels
Method Name: GetAccessLevels
VB Signature: (accessLevels as Variant)
VC++ Signature: (VARIANT* accessLevels)
Description: Returns all the access levels available to cardholders on the system.
Parameters: accessLevels: An array of AccessLevel objects.

GetCardByID
Method Name: GetCardByID
VB Signature: (ByVal ID as Long, ByRef aCard as Variant)
VC++ Signature: (long ID, VARIANT* aCard)
Description: Returns the Card by its ID.
Parameters:

- **ID:** Uniquely identifies the Card to retrieve.
- **aCard:** On return, this will be set to a Card object.

GetCardHolderByID
Method Name: GetCardHolderByID
VB Signature: (ByVal ID as Long, ByRef aCardHolder as Variant)
VC++ Signature: (long ID, VARIANT* aCardHolder)
Description: Returns the CardHolder by its ID.
Parameters:

- **ID:** Uniquely identifies the CardHolder to retrieve.
- **aCardHolder:** On return, this will be set to a CardHolder object.

GetEvents
Method Name: GetEvents
VB Signature: (ByRef callbackSetup as Object)
VC++ Signature: (Idispatch** callbackSetup)
Description: The asynchronous function returns all events related to the account based on the information specified in the callbackSetup argument. callbackSetup must be an instance of an EventCallBackSetup object.
Parameters: callbackSetup: Specifies any search criteria for the event retrieval.

GetHardwareByDeviceID
Method Name: GetHardwareByDeviceID
VB Signature: (ByVal deviceID as Long, ByRef hardwareDevice as Variant)
VC++ Signature: (long deviceID, VARIANT* hardwareDevice)
Description: Returns the HWIndependentDevice by its device ID.
Parameters:

- **deviceID:** One of two ways to uniquely identify a piece of hardware.
- **hardwareDevice:** On return, this will be set to a HWIndependentDevice object.

GetHardwareByID
Method Name: GetHardwareByID
VB Signature: (ByVal ID as Long, ByRef hardwareDevice as Variant)
VC++ Signature: (long ID, VARIANT* hardwareDevice)
Description: Returns the HWIndependentDevice by its ID.
Parameters:

- **ID:** One of two ways to uniquely identify a piece of hardware.
- **hardwareDevice:** On return, this will be set to a HWIndependentDevice object.

Login
Method Name: Login
VB Signature: (userID as Long, Optional userName as String = "", Optional password As String = "", Optional domainName As String = "")
VC++ Signature: (long* userID, BSTR* userName, BSTR* password, BSTR* domainName)
Description: Performs a normal, WIN-PAK login through the database server. This method allows the caller to use WIN-PAK's NT authentication if NT authentication has been enabled through the System Settings dialog in WIN-PAK. When using NT authentication, you can allow the current user to login without getting their userName and password—the system knows how to extract the user's SID (security ID) and find out if they are in the WIN-PAK group. For information on using NT integrated security, see the WIN-PAK user manual.
Parameters:

- **userID:** If the user successfully log in this contains a value greater than 0. The number identifies the currently logged in user. Pass this value to the Operator.Init method.
- **userName:** Identifies the name WIN-PAK knows the user by.
- **password:** The plaintext password the user typed in.
- **domainName:** If the system uses NT security, identifies the domain (or machine if using an NT workgroup) the user belongs to. This is not needed if the system only has one user by the name given in userName. Ex. You only need domainName if you have users server1\bob and server2\bob in the WIN-PAK group.

Card

Properties:

Property Name	VB Type	VC++ Type	Description
ID	Long	Long	Uniquely identifies the entity to the application.
CardNumber	String	_bstr_t	The string that identifies the card.
ActivationDate	Date	DATE	The date that the card was activated.
ExpirationDate	Date	DATE	The date that the card will expire.
AccountID	Long	Long	Identifies which account the card belongs to.
CardHolderID	Long	Long	Identifies which cardholder the card belongs to.
AccessLevel	Long()	_variant_t	Identifies which access levels the card is associated with.

Methods:

GetEvents
Method Name: GetEvents
VB Signature: (ByRef callbackSetup as Object)
VC++ Signature: (IDispatch** callbackSetup)
Description: The asynchronous function returns all events related to the account based on the information specified in the callbackSetup argument. callbackSetup must be an instance of an EventCallBackSetup object.
Parameters: callbackSetup: Specifies any search criteria for the event retrieval.

Init
Method Name: Init
VB Signature: (operatorObj As Object)
VC++ Signature: (IDispatch** operatorObj)
Description: If you create the Card object, it is your responsibility to tell it which operator is manipulating the Card. Failure to do so will cause it to throw COM exceptions whenever you try to view or set data within the object. If a NCIWinPak object returns a Card to you, you can assume that the Init has already been called for you.
Parameters: operatorObj: An initialized NCIWinPak.Operator object.

Save

Method Name: Save
VB Signature: No parameters
VC++ Signature: No parameters
Description: Saves a card and its changes. If this is a new card, you must minimally set the AccountID and CardHolderID.

CardHolder

Properties:

Property Name	VB Type	VC++ Type	Description
ID	Long	long	Uniquely identifies the entity to the application.
FirstName	String	BSTR	Cardholder's first name.
LastName	String	BSTR	Cardholder's last name.
NoteFields	String()	BSTR[]	A WIN-PAK user can define up to 40 user defined fields. In the future, this limit will be removed and we will allow more (or fewer) fields as the user decides. These arrays will be passed around as SAFEARRAYs. Always ask for the UBound of the value before iterating over the entries in the array.
AccountID	Long	long	Identifies which account the cardholder belongs to.

Methods:

GetCards

Method Name: GetCards
VB Signature: (ByRef cards as Variant)
VC++ Signature: (VARIANT* cards)
Description: Returns all cards related to the cardholder.
Parameters: cards: This returns all cards associated with the cardholder.

GetEvents

Method Name: GetEvents
VB Signature: (ByRef callbackSetup as Object)
VC++ Signature: (IDispatch** callbackSetup)
Description: The asynchronous function returns all events related to the account based on the information specified in the callbackSetup argument. callbackSetup must be an instance of an EventCallBackSetup object.
Parameters: callbackSetup: Specifies any search criteria for the event retrieval.

Init

Method Name: Init
VB Signature: (operatorObj As Object)
VC++ Signature: (IDispatch** operatorObj)
Description: If you create the CardHolder object, it is your responsibility to tell it which operator is manipulating the CardHolder. Failure to do so will cause it to throw COM exceptions whenever you try to view or set data within the object. If a NCIWinPak object returns a CardHolder to you, you can assume that the Init has already been called for you.
Parameters: operatorObj: An initialized NCIWinPak.Operator object.

Save
Method Name: Save
VB Signature: No parameters
VC++ Signature: No parameters
Description: Saves a cardholder and its changes. If this is a new cardholder, you must set the AccountID.

SetExpirationDateForAllCards
Method Name: SetExpirationDateForAllCards
VB Signature: (ByVal expDate As Date)
VC++ Signature: (DATE expDate)
Description: Provides a method to set the expiration date for all cards owned by a cardholder.
Parameters: expDate: The date to expire all of the cardholder's cards.

GetPictures
Method Name: GetPictures
VB Signature: (pictures As Variant)
VC++ Signature: (VARIANT* pictures)
Description: Returns all of the pictures associated with a given cardholder. If a site has many pictures for each cardholder this method could eat up a lot of memory quickly. Each picture is returned as an array of bytes. The sample VB program writes the bytes to a temp file and then loads the file into a Picture or Image control.
Parameters: pictures: An array of arrays of bytes. Each byte array contains a JPEG file as it was loaded into memory (i.e. raw bytes, not a decoded image).

GetSignatures
Method Name: GetSignatures
VB Signature: (signatures As Variant)
VC++ Signature: (VARIANT* signatures)
Description: Returns all of the signatures associated with a given cardholder. If a site has many signatures for each cardholder this method could eat up a lot of memory quickly. Each signature is returned as an array of bytes. The sample VB program writes the bytes to a temp file and then loads the file into a Picture or Image control.
Parameters: signatures: An array of arrays of bytes. Each byte array contains a WMF (Windows metafile) file as it was loaded into memory (i.e. raw bytes, not a decoded image).

AddPhoto
Method Name: AddPhoto
VB Signature: (thePhoto As Variant)
VC++ Signature: (VARIANT* thePhoto)
Description: Provides a way to import photos into WIN-PAK. The bytes contained in thePhoto must be the raw bytes of a JPEG file (as it would be stored on disk). If thePhoto is not a JPEG, the function throws a COM exception.
Parameters: thePhoto: The file loaded as a byte array (VT_ARRAY | VT_UI1 (C safearray or Byte() in VB).

AddSignature
Method Name: AddSignature
VB Signature: (theSignature As Variant)
VC++ Signature: (VARIANT* theSignature)
Description: Provides a way to import signatures into WIN-PAK. The bytes contained in theSignature must be the raw bytes of a WMF file (as it would be stored on disk).
Parameters: theSignature: The file loaded as a byte array (VT_ARRAY | VT_UI1 (C safearray or Byte() in VB).

GetNoteField:

Method Name: GetNoteField
VB Signature: (fieldName As String) As String
VC++ Signature: _bstr_t (BSTR* fieldName)
Description: Retrieves a NoteField by the name used to identify the field within the WIN-PAK notefield template tab(s). You can retrieve the NoteField names from the appropriate Account instance.
Parameters: fieldName: Name of the field as known to the Account object.

SetNoteField:

Method Name: SetNoteField
VB Signature: (fieldName As String, value As String)
VC++ Signature: (BSTR* theSignature, BSTR* value)
Description: Sets a NoteField by the name used to identify the field within the WIN-PAK notefield template tab(s). You can retrieve the NoteField names from the appropriate Account instance.
Parameters:

- **fieldName:** Name of the field as known to the Account object.
- **value:** The new value for the named fieldName.

CardHolderCallBackSetup

Properties:

Property Name	VB Type	VC++ Type	Description
ObjectsRequested	Long	long	Caller fills this in to indicate how many objects they want to retrieve at a time. Set this to –1 to get all available objects. (–1 is the default value.)
CallBackObject	Variant	VARIANT	When records are ready, this object gets them. This object must implement IDispatch so that the proper function names can be located and called. This must follow the setup defined by the NCICallBackObject. If you are using synchronous callbacks, this may be an empty variant.
theOperator	Object	IDispatchPtr	Assigns or retrieves the currently logged in operator to the object.
AccountID	Long	long	Identifies the account to retrieve cardholders from.

Methods:

SetAdditionalSearchCriteria
Method Name: SetAdditionalSearchCriteria
VB Signature: (inputSearchFields() As String, fieldData() As String, comparisons() as ComparisonType)
VC++ Signature: (SAFEARRAY** inputSearchFields, SAFEARRAY** fieldData, SAFEARRAY** comparisons)
Description: Allows you to specify additional criteria when looking up cardholders. This allows searching on FirstName, LastName, and any of the other fields returned by Account.CardHolderFilterFieldNames.
Parameters:

- **inputSearchFields:** Array of fields to search on. The field names should come from the CardHolderFilterFieldNames property.
- **fieldData:** Data to perform the search against.

- **comparisons:** Using the ComparisonType data-type, this tells what type of comparison you want to perform between the inputSearchFields(i) and fieldData(i). See ComparisonType in the enumerated types section for details.

StopCallback
Method Name: StopCallback
VB Signature: No parameters
VC++ Signature: No parameters
Description: If you want to terminate the processing of the cardholder callback for any reason, call this function. You may want to do this if you no longer want cardholders to be returned.

EventObj

Properties:

Property Name	VB Type	VC++ Type	Description
CardHolderFirstName	String	_bstr_t	The related cardholder's first name. You can also obtain this information from the relatedCardHolder.
CardHolderLastName	String	_bstr_t	The related cardholder's last name. You can also obtain this information from the relatedCardHolder.
GenTime	Date	DATE	Time that the history record was created.
HWDeviceName	String	_bstr_t	The name of the hardware device involved in the event.
ID	Long	long	Uniquely identifies the entity to the application.
InternalDescription	String	_bstr_t	
ReceiveTime	Date	DATE	Time that the history record was received.
relatedCard	Variant	_variant_t	Returns the related Card object if one exists.
relatedCardHolder	Variant	_variant_t	Returns the related CardHolder object if one exists.
relatedHWIndependent Device	Variant	_variant_t	Returns the related HWIndependentDevice object if one exists.
SequenceID	Long	long	Identifies the sequence ID of the event.

Methods:

GetEventCodeText
Method Name: GetEventCodeText
VB Signature: (ByVal language as String, ByRef codeText as String)
VC++ Signature: (BSTR language, BSTR* codeText)
Description: The event code as text (English only)—This should eventually become a lookup where user requests code in LanguageX and the object gives back the string in that language.
Parameters:

- **language:** At this point in time, the object only supports an English translation of the event code. This language code corresponds to the standard language codes one can create using the MAKELANGID macro in C.
- **codeText:** On return, this will contain the text in the language of your choice.

GetTypeText
Method Name: GetEventCodeText
VB Signature: (ByVal language as String, ByRef codeText as String)
VC++ Signature: (BSTR language, BSTR* codeText)
Description: The event code as text (English only)—This should eventually become a lookup where user requests code in LanguageX and the object gives back the string in that language.
Parameters:

- **language:** At this point in time, the object only supports an English translation of the event code. This language code corresponds to the standard language codes one can create using the MAKELANGID macro in C.
- **codeText:** On return, this will contain the text in the language of your choice.

Init
Method Name: Init
VB Signature: (operatorObj As Object)
VC++ Signature: (IDispatch** operatorObj)
Description: If you create the EventObj object, it is your responsibility to tell it which operator is manipulating the EventObj. Failure to do so will cause it to throw COM exceptions whenever you try to view or set data within the object. If a NCIWinPak object returns an EventObj to you, you can assume that the Init has already been called for you.
Parameters: operatorObj: An initialized NCIWinPak.Operator object.

EventCallBackSetup

Properties:

Property Name	VB Type	VC++ Type	Description
alarmPointID	Long	long	Set this value to the ID of a specific alarm point to get events related to that point.
CallBackObject	Variant	IDispatch*	When records are ready, this object gets them. This object must implement IDispatch so that the proper function names can be located and called. This must follow the setup defined by the NCICallBackObject. If you are using synchronous callbacks, this may be an empty variant.
cardNumber	String	_bstr_t	Set this value to search for events related to a specific card.
EndDate	Date	DATE	Specifies the date and time of the next record to retrieve. The object will not look at this value once the object is passed to a GetEvents method. (Unless you call Reset()).
EventTypes	EventType()	SAFEARRAY	Used to set the event types to return. If this member is not set, all event types are returned.
firstName	String	_bstr_t	Set this value to search for events related to cardholders with a specific first name.
lastName	String	_bstr_t	Set this value to search for events related to cardholders with a specific last name.
ObjectsRequested	Long	long	Caller fills this in to indicate how many objects they want to retrieve at a time. Set this to –1 to get all available objects. (–1 is the default value.)
readerID	Long	long	Set this value to search for events related to a specific card reader.
SortOnSequenceID	Boolean	BOOL	Defaults to false. Set this value to true if you want the events returned in SequenceID order.
StartDate	Date	DATE	Specifies the date and time of the first record to retrieve. The object will not look at this value once the object is passed to a GetEvents method. (Unless you call Reset()).
theAlarmCodes	AlarmType ()	SAFEARRAY	Used to set the alarm types to return. If this member is not set, all alarm types are returned.
theOperator	Object	VARIANT	Assigns or retrieves the currently logged in operator to the object.

Methods:

StopCallback

Method Name: StopCallback
VB Signature: No parameters
VC++ Signature: No parameters
Description: If you want to terminate the processing of the event callback for any reason, call this function. You may do this if you no longer want events to be returned.

Reset

Method Name: Reset
VB Signature: No parameters
VC++ Signature: No parameters
Description: Re-queries the data and moves the initial record, set pointer to the start.

HWIndependentDevice

Properties:

Property Name	VB Type	VC++ Type	Description
Name	String	_bstr_t	Gives the device's name.
DeviceType	Long	long	Value giving the device's type.
Description	String	_bstr_t	Setup in WIN-PAK to describe the device in more detail than just the name.
ID	Long	long	Uniquely identifies the entity to the application
DeviceID	Long	long	Uniquely identifies the device to the application. (Yes, a HWIndependentDevice has two unique IDs.)

Methods:

GetEvents

Method Name: GetEvents
VB Signature: (ByRef callbackSetup as Variant)
VC++ Signature: (IDispatch** callbackSetup)
Description: The asynchronous function returns all events related to the account based on the information specified in the callbackSetup argument. callbackSetup must be an instance of an EventCallBackSetup object.
Parameters: callbackSetup: Specifies any search criteria for the event retrieval.

Init

Method Name: Init
VB Signature: (operatorObj As Object)
VC++ Signature: (IDispatch** operatorObj)
Description: If you create the HWIndependentDevice object, it is your responsibility to tell it which operator is manipulating the HWIndependentDevice. Failure to do so will cause it to throw COM exceptions whenever you try to view or set data within the object. If a NCIWinPak object returns an HWIndependentDevice to you, you can assume that the Init has already been called for you.
Parameters: operatorObj: An initialized NCIWinPak.Operator object.

Operator

Properties:

Property Name	VB Type	VC++ Type	Description
operatorName	String	_bstr_t	Gives the operator's name.
operatorDomain	String	_bstr_t	Gives the operator's domain name.
accounts	Variant	_variant_t	Array of longs that are the IDs of accounts the operator has access to.

Methods:

Init
Method Name: Init
VB Signature: (operatorID as Long)
VC++ Signature: (long* operatorID)
Description: Given an operator ID, loads up all the permissions and other data related to the operator.
Parameters: operatorID: The operator's ID as returned from a call to NCIWinPak.Login.

User Provided Callback Object

This object will implement the properties and methods described in a COM object that also implements IDispatch.

Properties:

Property Name	VB Type	VC++ Type	Description
TotalCount	Long	long	Allows the NCIWinPak object to inform the client how many objects it will return.

Methods:

SetAllObjectsRetrieved
Method Name: SetAllObjectsRetrieved
VB Signature: No arguments
VC++ Signature: No arguments
Description: NCIWinPak will call into this function to indicate that it has returned all the requested objects. This allows the DBAPI to tell you how many objects it thinks it will return but allow for more or fewer objects to be returned. For example, an operator may unbuffer, causing a number of events to show up in history that were not present when the object count was retrieved.

SetData
Method Name: SetData
VB Signature: (values As Variant)
VC++ Signature: (VARIANT* values)
Description: NCIWinPak will call into this function to deliver the array of EventObj or CardHolder.
Parameters: operatorID: The operator's ID as returned from a call to NCIWinPak.Login.

Enumerated Types

AlarmType

These values are used when filtering alarm-specific events.

EntranceDoorAjar - Door is stuck open.

EntranceDoorNormal - The entrance is functioning normally.

EntranceForcedOpen - Something forced the door open.

EntranceTrouble - Something is wrong at an entrance.

InputActive - A alarm was triggered by an input point.

InputDoorAjar - Door is stuck open.

InputNormal - A check of an input point indicated everything is working.

InputTrouble1 - Input trouble, type 1. Usually, you will combine this with InputTrouble2 when querying history.

InputTrouble2 - Input trouble, type 2. Usually, you will combine this with InputTrouble1 when querying history.

CardCodes

These identify the various event types one can see related to a card event.

APBViolation - Anti-passback violation.

BeforeActivation - Card was found but has not been activated yet.

CardNotFound - Card was not found in the card database.

DoorUnlocked - Door was unlocked.

EntranceAPBUsed - Anti-passback used.

EntranceDuressUsed - Door was forced open.

EntranceFormatReversed - User put their card through backwards.

EntranceLocked - The entrance was locked.

EntranceNeverAllowed - The door is an exit only.

ExpiredCard - Someone presented an expired card to the reader.

InvalidPIN - Someone tried to open the door using an invalid PIN number.

OccupancyLimit - The room has reached its occupancy limit. The door did not open.

UseLimit - The card has reached its usage limit and did not open the door.

ValidCard - Return all valid card reads.

ComparisonType

CtEquals - Performs the comparison: **`field = value`**

CtGreaterThan - Performs the comparison: **`field > value`**

CtGreaterThanEqual - Performs the comparison: **`field >= value`**

CtLessThan - Performs the comparison: **`field < value`**

CtLessThanEqual - Performs the comparison: **`field <= value`**

EventType

HisAlarm - Includes general alarms in the event callback.

HisAlarmAck - Includes alarm acknowledgements in the event callback.

HisAlarmClr - Includes alarm clears in the event callback.

HisAlarmMessage - Includes alarm messages in the event callback.

HisCard - Includes miscellaneous card events in the event callback.

HisCardAck - Includes card acknowledges in the event callback.

HisCardClr - Includes card clears in the event callback.

HisCardMessage - Includes card messages in the event callback.

HisDatabase - Includes database events in the event callback.

HisOperator - Includes operator events in the event callback.

HisSystem - Includes system events in the event callback.

HisSystemAlarm - Includes general system alarms in the event callback.

HisSystemAlarmAck - Includes system alarm acknowledgments in the event callback.

HisSystemAlarmClr - Includes system alarm clears in the event callback.

HisSystemAlarmMessage - Includes system alarm messages in the event callback.

5007 South Howell Avenue
Mi.lwaukee, WI 53207

www.nciaccess.com

Phone: (800) 323-4576
Fax: (414) 769-5989