

A blue wavy banner spanning the width of the page, containing the NETOP logo and the product name.

NETOP™

RemoteControl

Secure Remote Management and Support

NGuestX API specifications

December 2009

Table of Contents

Introduction.....	3
Design	3
Abbreviations	3
Class Documentation	4
NGuestXLib::_INGuestXCtrlEvents	4
INGuestXCtrl.....	14
INGuestXEventParam	40
INGuestXFont	41
INGuestXRcArea	43
INGuestXShortcut.....	45

Introduction

This document describes the Netop Guest ActiveX API.

Design

The Netop Guest ActiveX control partially implements the functionality of the *Netop* Remote Control Guest application and is ready for use from a Web page or other ActiveX container.

A container application can load any number of instances of the control. Each instance has its own connection settings and scripted event handlers, each instance can connect to one host and open one session of each type (currently only Remote Control and Chat are supported).

The control has built in default dialogs, which can be overwritten from a script, for all user input.

Abbreviations

Netop Guest ActiveX - **NGA**
Remote Control - **RC**

Class Documentation

NGuestXLib::_INGuestXCtrlEvents Interface Reference

Event handler interface for INGuestX class.

Public Member Functions

- HRESULT [OnOpenPre](#) ()
- HRESULT [OnOpenPost](#) ([in] VARIANT_BOOL Ok)
- HRESULT [OnClosePre](#) ()
- HRESULT [OnClosePost](#) ([in] VARIANT_BOOL Ok)
- HRESULT [OnConnectPre](#) ()
- HRESULT [OnConnectPost](#) ([in] VARIANT_BOOL Ok)
- HRESULT [OnDisconnectPre](#) ()
- HRESULT [OnDisconnectPost](#) ()
- HRESULT [OnBeginSessionPre](#) ([in] LONG Type)
- HRESULT [OnBeginSessionPost](#) ([in] LONG Type,[in] VARIANT_BOOL Ok)
- HRESULT [OnEndSessionPre](#) ([in] LONG Type)
- HRESULT [OnEndSessionPost](#) ([in] LONG Type,[in] VARIANT_BOOL Ok)
- HRESULT [OnSessionStarted](#) ([in] LONG Type)
- HRESULT [OnEndedByHost](#) ()
- HRESULT [OnConnectionLost](#) ()
- HRESULT [OnLoginPassword](#) ([in] LONG Reason,[in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnLoginNetop](#) ([in] LONG Reason,[in] VARIANT_BOOL bNss,[in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnLoginWindows](#) ([in] LONG Reason,[in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnLoginLdap](#) ([in] LONG Reason,[in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnLoginRsa](#) ([in] LONG Reason,[in] LONG Shadow,[in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnLoginFailed](#) ([in] LONG Reason)
- HRESULT [OnEnterRsaPincode](#) ([in] LONG Reason,[in] LONG Mode,[in] BSTR SuggestedPin,[in] LONG MinLen,[in] LONG MaxLen,[in] VARIANT_BOOL AllowNonNumeric,[in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnHostScreenSizeInfo](#) ([in] LONG Width,[in] LONG Height)
- HRESULT [OnHostMultiGuestInfo](#) ([in] LONG Event,[in] LONG NumGuests,[in] LONG Error)
- HRESULT [OnEraseBackground](#) ([in] LONG hWnd,[in] LONG hDC,[in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnChatMessageIn](#) ([in] BSTR Msg,[in] [INGuestXFont](#) *Font,[in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnChatMessageOut](#) ([in] BSTR Msg,[in] [INGuestXFont](#) *Font,[in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnErrorMsg](#) ([in] LONG MsgNo,[in] BSTR Message,[in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnInfoMsg](#) ([in] LONG MsgNo,[in] BSTR Message,[in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnConnectWindow](#) ([in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnStatusWindow](#) ([in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnPropertyWindow](#) ([in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnChatUIStart](#) ([in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnChatUIEnd](#) ()

- HRESULT [OnInfoWindow](#) ([in] LONG Reason,[in] [INGuestXEventParam](#) *EventParam)
- HRESULT [OnLicenseRequired](#) ([in] LONG MsgNo,[in] BSTR Message,[in] [INGuestXEventParam](#) *EventParam)

Detailed Description

Event handler interface for INGuestX class.

Member Function Documentation

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnOpenPre ()

Fired when the NGA instance is about to open by [Open\(\)](#) method.

The event is not fired when the instance is already opened when [INGuestXCtrl::Open\(\)](#) is called.

The event is always followed by [OnClosePost\(\)](#) event.

When the event is fired the [INGuestXCtrl::IsOpen](#) property is always false.

HRESULT _INGuestXCtrlEvents::OnOpenPost ([in] VARIANT_BOOL *Ok*)

Fired when the NGA instance has been opened by [INGuestXCtrl::Open\(\)](#) method.

When the event is fired [INGuestXCtrl::IsOpen](#) property is true if the instance was opened successfully. It is safe to call the [INGuestXCtrl::Close\(\)](#) in response to OnOpenPost(true).

Parameters:

- Ok* - status of Open request
- true - the NGA was opened successfully
 - false - failed to open NGA instance.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnClosePre ()

Fired when the NGA instance is about to close by [INGuestXCtrl::Close\(\)](#) method.

The event is not fired when the instance is already closed when [INGuestXCtrl::Close\(\)](#) is called.

There will be [OnClosePost\(\)](#) event fired for each [OnClosePre\(\)](#) event.

When the event is fired [INGuestXCtrl::IsOpen](#) is always true.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnClosePost ([in] VARIANT_BOOL *Ok*)

Fired when the NGA instance has been closed by [INGuestXCtrl::Close\(\)](#) method.

When the event is fired [INGuestXCtrl::IsOpen](#) property is false if the instance was closed successfully. It is safe to call the [INGuestXCtrl::Open\(\)](#) in response to OnClosePost(true).

Parameters:

- Ok* - status of Close() request
- true - the NGA was closed successfully
 - false - failed to close NGA.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnConnectPre ()

Fired if a new connection should be established in response to [INGuestXCtrl::BeginSession\(\)](#) function.

The event is fired before any long lasting network operations started.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnConnectPost ([in] VARIANT_BOOL Ok)

Fired after [OnConnectPre\(\)](#) when a connection was established successfully or NGA failed to establish a new connection.

Parameters:

Ok - operation status

- true - the connection established successfully
- false - failed to establish a connection

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnDisconnectPre ()

Fired after [OnEndSessionPre\(\)](#) in response to EndSession() if the connection should be terminated because there is no more active session.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnDisconnectPost ()

Fired after [OnDisconnectPre\(\)](#) method when NGA has been disconnected from Host.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnBeginSessionPre ([in] LONG Type)

The very first event that can be fired in response to [INGuestXCtrl::BeginSession\(\)](#) before any long lasting network operations started.

Parameters:

Type - session type

- [INGuestXCtrl::SessionType_RemoteControl](#) - Remote control
- [INGuestXCtrl::SessionType_Chat](#) - Text chat

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnBeginSessionPost ([in] LONG Type, [in] VARIANT_BOOL Ok)

Fired after [OnBeginSessionPre\(\)](#) when a session was established successfully or NGA failed to establish a new session.

When a new connection is created after [INGuestXCtrl::BeginSession\(\)](#), after having [OnBeginSessionPre\(\)](#) event, the authentication event should be expected (e.g. OnLoginPassword, OnLoginNetop, etc).

It is safe to open a new session of another type in response to this event only if a [INGuestXCtrl::BeginSession\(\)](#) was called with active connection. Otherwise it will be safe to start a session only after authentication, i.e. in response to [OnSessionStarted\(\)](#) event.

Parameters:

Type - session type

- [INGuestXCtrl::SessionType RemoteControl](#) - Remote control
- [INGuestXCtrl::SessionType Chat](#) - Text chat

Ok - status of BeginSession request

- true - the session established successfully
- false - failed to establish a session

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnEndSessionPre ([in] LONG *Type*)

Fired when NGA was requested to close a session by [INGuestXCtrl::EndSession\(\)](#) method before any long lasting network operations started.

Parameters:

Type - session type

- [INGuestXCtrl::SessionType RemoteControl](#) - Remote control
- [INGuestXCtrl::SessionType Chat](#) - Text chat

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnEndSessionPost ([in] LONG *Type*, [in] VARIANT_BOOL *Ok*)

Fired after [OnEndSessionPre\(\)](#) when the session was closed by [INGuestXCtrl::EndSession\(\)](#).

It is safe to open a new session of the same type in response to this event.

Parameters:

Type - session type

- [INGuestXCtrl::SessionType RemoteControl](#) - Remote control
- [INGuestXCtrl::SessionType Chat](#) - Text chat

Ok - status of EndSession request

- true - the session closed successfully
- false - failed to close a session

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnSessionStarted ([in] LONG *Type*)

Fired after the session went into the running state after having a new connection established.

Parameters:

Type - session type

- [INGuestXCtrl::SessionType RemoteControl](#) - Remote control
- [INGuestXCtrl::SessionType Chat](#) - Text chat

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnEndedByHost ()

Fired when a session and connection was ended by host.

[OnErrorMsg\(\)](#) event will be fired with message #?? after this event.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnConnectionLost ()

Fired after connection was lost.

[OnErrorMsg\(\)](#) event will be fired with message #?? after this event.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnLoginPassword ([in] LONG *Reason*, [in] [INGuestXEventParam](#) * *EventParam*)

Fired when Netop password should be sent to Host.

Default action: Built-in dialog will be shown to prompt password. Event handler can suppress the built-in login dialog and send password with [INGuestXCtrl::SendLoginPassword\(\)](#) method.

To Cancel login [INGuestXCtrl::EndSession\(\)](#) function can be called.

Parameters:

Reason - The why this prompt is needed

EventParam - event parameter object. Handler can change its property Handled to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnLoginNetop ([in] LONG *Reason*, [in] VARIANT_BOOL *bNss*, [in] [INGuestXEventParam](#) * *EventParam*)

Fired when Netop guest ID and password should be sent to Host.

Default action: Built-in dialog will be shown to prompt for ID and password. Event handler can suppress the built-in login dialog and send password with [INGuestXCtrl::SendLoginNetop\(\)](#) method.

To Cancel login [INGuestXCtrl::EndSession\(\)](#) function can be called.

Parameters:

Reason - The why this prompt is needed

bNss - If the Host is configured for Netop Security Server authentication. When a Host is configured for Nss authentication Guest can change by sending new password with [INGuestXCtrl::SendLoginNetop\(\)](#) method.

EventParam - event parameter object. Handler can change its property Handled to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnLoginWindows ([in] LONG *Reason*, [in] [INGuestXEventParam](#) * *EventParam*)

Fired when Windows login, domain and password should be sent to Host.

Default action: Built-in dialog will be shown to prompt for login, domain and password. Event handler can suppress the built-in login dialog and send password with [INGuestXCtrl::SendLoginWindows\(\)](#) method.

To Cancel login [INGuestXCtrl::EndSession\(\)](#) function can be called.

Parameters:

Reason - The why this prompt is needed

EventParam - event parameter object. Handler can change its property Handled to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnLoginLdap ([in] LONG Reason, [in] [INGuestXEventParam](#) * EventParam)

Fired when LDAP server name, login and password should be sent to Host.

Default action: Built-in dialog will be shown to prompt for server name, login and password. Event handler can suppress the built-in login dialog and send password with [INGuestXCtrl::SendLoginLdap\(\)](#) method.

To Cancel login [INGuestXCtrl::EndSession\(\)](#) function can be called.

Parameters:

Reason - The why this prompt is needed

EventParam - event parameter object. Handler can change its property Handled to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnLoginRsa ([in] LONG Reason, [in] LONG Shadow, [in] [INGuestXEventParam](#) * EventParam)

Fired when login name, RSA SecurID passcode and password should be sent to Host.

Default action: Built-in dialog will be shown to prompt for login, RSA passcode and password. Event handler can suppress the built-in login dialog and send password with [INGuestXCtrl::SendLoginRsa\(\)](#) method.

To Cancel login [INGuestXCtrl::EndSession\(\)](#) function can be called.

Parameters:

Reason - The why this prompt is needed

Shadow - 1 if a Netop password is required in addition to the RSA SecurID PASSCODE.

EventParam - event parameter object. Handler can change its property Handled to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnLoginFailed ([in] LONG Reason)

Fired when the login has been failed.

?? Is it safe to open a new session.

Parameters:

Reason - the reason of why login has failed

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnEnterRsaPincode ([in] LONG Reason, [in] LONG Mode, [in] BSTR SuggestedPin, [in] LONG MinLen, [in] LONG MaxLen, [in] VARIANT_BOOL AllowNonNumeric, [in] [INGuestXEventParam](#) * EventParam)

Fired when the server side requires RSA SecurID pin code.

Default action: Built-in dialog will be shown to prompt RSA SecurID pin code. Event handler can suppress the built-in login dialog and send password with [INGuestXCtrl::SendLoginRsaPin\(\)](#) method.

Parameters:

Reason - The why this prompt is needed

Mode - 0 = fixed, 1,2 = selectable (2 has no suggestion).

SuggestedPin - The suggested pin code if any. May be NULL.

MinLen - The minimum length for a valid pin code.

MaxLen - The maximum length for a valid pin code.

AllowNonNumeric - True if characters other than 0-9 are allowed

EventParam - event parameter object. Handler can change its property *Handled* to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnHostScreenSizeInfo ([in] LONG *Width*, [in] LONG *Height*)

Fired when the size of the remote screen is changed.

Parameters:

Width - new width of the remote screen

Height - new height of the remote screen

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnHostMultiGuestInfo ([in] LONG *Event*, [in] LONG *NumGuests*, [in] LONG *Error*)

Fired when multi guest parameters have been updated on Host.

Parameters:

Event - bitwise combination of the *MultiguestEvent_t* flags:

- [INGuestXCtrl::MultiguestEvent InputAssigned](#) - This Guest is assigned input control
- [INGuestXCtrl::MultiguestEvent InputRevoked](#) - This Guest is revoked input control
- [INGuestXCtrl::MultiguestEvent InputDenied](#) - This Guest requested input control but it was denied. See error for optional error code.
- [INGuestXCtrl::MultiguestEvent ConnectionsChanged](#) - Number of session changed
- [INGuestXCtrl::MultiguestEvent MultiSessionsSuspended](#) - More sessions suspended
- [INGuestXCtrl::MultiguestEvent MultiSessionsAllowed](#) - More sessions allowed
- [INGuestXCtrl::MultiguestEvent MultiSessionsDenied](#) - Change of sessions denied

NumGuest - new number of connected Guest (only on [INGuestXCtrl::MultiguestEvent ConnectionsChanged](#) event)

Error - additional information (only for [INGuestXCtrl::MultiguestEvent InputDenied](#) and [INGuestXCtrl::MultiguestEvent MultiSessionsDenied](#) events).

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnEraseBackground ([in] LONG *hWnd*, [in] LONG *hDC*, [in] [INGuestXEventParam](#) * *EventParam*)

Fired when NGA control background should be erased.

Default action: If there is an RC session the background is erased by black color. When there is no RC session the default NGA bitmap is shown.

Parameters:

hWnd - NGA window handler

hDC - device context for erase background windows message

EventParam - event parameter object. Handler can change its property *Handled* to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnChatMessageIn ([in] BSTR *Msg*, [in] [INGuestXFont](#) * *Font*, [in] [INGuestXEventParam](#) * *EventParam*)

Fired on incoming chat message.

Default action: show the message in the chat dialog if the one is opened.

Parameters:

Msg - received chat message

EventParam - event parameter object. Handler can change its property *Handled* to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnChatMessageOut ([in] BSTR *Msg*, [in] [INGuestXFont](#) * *Font*, [in] [INGuestXEventParam](#) * *EventParam*)

Fired on outgoing chat message.

The event fired after [INGuestXCtrl::SendChatMessage\(\)](#) was called. The message can be different from one passed to [INGuestXCtrl::SendChatMessage\(\)](#) method because "<PC Name>" string is inserted.

Default action: show the message in the chat dialog if the one is opened.

Parameters:

Msg - received chat message

EventParam - event parameter object. Handler can change its property *Handled* to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnErrorMsg ([in] LONG *MsgNo*, [in] BSTR *Message*, [in] [INGuestXEventParam](#) * *EventParam*)

Fired before any NGA error message is shown.

Default action: Built-in error message box is shown. Event handler can suppress the message box.

Parameters:

MsgNo - id of message format string

Message - message to be shown

EventParam - event parameter object. Handler can change its property *Handled* to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnInfoMsg ([in] LONG *MsgNo*, [in] BSTR *Message*, [in] [INGuestXEventParam](#) * *EventParam*)

Fired before any NGA information message is shown.

Parameters:

MsgNo - id of message format string

Message - message to be shown

EventParam - event parameter object. Handler can change its property *Handled* to true to suppress the default action.

Default action: Status changed in the built-in window if the window is open.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnConnectWindow ([in] [INGuestXEventParam](#) * *EventParam*)

Fired when user left clicks on ActiveX area when there is no active connection.

Default action: Built-in dialog will be shown to setup a new connection. Event handler can suppress the built-in dialog.

Parameters:

EventParam - event parameter object. Handler can change its property *Handled* to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnStatusWindow ([in] [INGuestXEventParam](#) * *EventParam*)

Fired when user pressed the keyboard shortcut for connection status window when there is an active connection.

Default action: Built-in Connection Status dialog will be shown. Event handler can suppress the built-in dialog.

Parameters:

EventParam - event parameter object. Handler can change its property *Handled* to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnPropertyWindow ([in] [INGuestXEventParam](#) * *EventParam*)

Fired when user right clicks on ActiveX area when there is no active connection.

Default action: Built-in Connection Properties dialog will be shown. Event handler can suppress the built-in dialog.

Parameters:

EventParam - event parameter object. Handler can change its property *Handled* to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnChatUIStart ([in] [INGuestXEventParam](#) * *EventParam*)

Fired when a Chat UI should be shown.

Default action: show the chat window.

Parameters:

EventParam - event parameter object. Handler can change its property *Handled* to true to suppress the default action.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnChatUIEnd ()

Fired when a Chat UI should be hidden.

NGA hides the chat window if the one is opened.

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnInfoWindow ([in] LONG *Reason*, [in] [INGuestXEventParam](#) * *EventParam*)

Fired when an info window should be shown.

Default action: open the modal info window to show info messages.

Parameters:

Reason one of the [INGuestXCtrl::InfoWindowReason_t](#) constants

HRESULT NGuestXLib::_INGuestXCtrlEvents::OnLicenseRequired ([in] LONG *MsgNo*, [in] BSTR *Message*, [in] [INGuestXEventParam](#) * *EventParam*)

Fired when current license does not allow to start the connection initiated either via GUI or BeginSession() method.

Default ActiveX handler shows the License GUI window. Default GUI can be suppressed by setting the property Handled of the param object to TRUE in application event handler.

This event will be fired continuously until either the license will be accepted by the ActiveX or the connection will be canceled by setting the property Canceled of the param object to TRUE.

Parameters:

MsgNo - id of message format string

Message - Either contains the description that a license is required for non WebConnect connections or the Sentinel error why current license key cannot be used.

EventParam - event parameter object. Handler can change its property Handled to true to suppress the default action. Handler can change its property Canceled to true to terminate connection process. Property Canceled is checked only if Handled was set to true.

Default action: open the modal license dialog window to enter the license.

INGuestXCtrl Interface Reference

Netop Guest ActiveX Interface.

Constants

- enum [CommProfile_t](#) { [CommProfile_TCP](#) = 0x00000000, [CommProfile_HTTP](#) = 0x00000001, [CommProfile_UDP](#) = 0x00000002, [CommProfile_WebConnect](#) = 0x00000003 }
- enum [SessionType_t](#) { [SessionType_RemoteControl](#) = 0x00000001, [SessionType_Chat](#) = 0x00000004 }
- enum [MouseMode_t](#) { [MouseMode_Local](#) = 0x00000000, [MouseMode_Remote](#) = 0x00000001, [MouseMode_None](#) = 0x00000002 }
- enum [KeyboardMode_t](#) { [KeyboardMode_Local](#) = 0x00000000, [KeyboardMode_Remote](#) = 0x00000001, [KeyboardMode_None](#) = 0x00000002 }
- enum [StretchMode_t](#) { [Stretch_FitWindowToHost](#) = 0x00000000, [Stretch_FitHostToWindow](#) = 0x00000001, [Stretch_FitNone](#) = 0x00000002 }
- enum [GraphicsMode_t](#) { [GraphicsMode_Hook](#) = 0x00000000, [GraphicsMode_AccBitmap](#) = 0x00000001, [GraphicsMode_NormalBitmap](#) = 0x00000002 }
- enum [MaxColors_t](#) { [MaxColors_Actual](#) = 0x00000000, [MaxColors_256](#) = 0x00000001, [MaxColors_16](#) = 0x00000002, [MaxColors_2](#) = 0x00000003 }
- enum [CompressionLevel_t](#) { [CompressionLevel_Auto](#) = 0x00000000, [CompressionLevel_None](#) = 0x00000001, [CompressionLevel_Low](#) = 0x00000002, [CompressionLevel_High](#) = 0x00000003 }
- enum [EncryptionLevel_t](#) { [EncryptionLevel_Compatible](#) = 0x00000000, [EncryptionLevel_None](#) = 0x00000001, [EncryptionLevel_DataIntegrity](#) = 0x00000002, [EncryptionLevel_Keyboard](#) = 0x00000003, [EncryptionLevel_DataIntegrityAndKeyboard](#) = 0x00000004, [EncryptionLevel_High](#) = 0x00000005, [EncryptionLevel_VeryHigh](#) = 0x00000006 }
- enum [DesktopOptimization_t](#) { [DesktopOptimization_DisableEverything](#) = 0x00000001, [DesktopOptimization_DisableWallpaper](#) = 0x00000002, [DesktopOptimization_DisableScreenSaver](#) = 0x00000004, [DesktopOptimization_DisableAnimation](#) = 0x00000008, [DesktopOptimization_DisableFullWindowDrag](#) = 0x00000010, [DesktopOptimization_DisableMenuAnimation](#) = 0x00000020, [DesktopOptimization_DisableComboboxAnimation](#) = 0x00000040, [DesktopOptimization_DisableSmoothScrolling](#) = 0x00000080, [DesktopOptimization_DisableGradientCaption](#) = 0x00000100, [DesktopOptimization_DisableActiveDesktop](#) = 0x00000200, [DesktopOptimization_DisableMenuFade](#) = 0x00000400, [DesktopOptimization_DisableSelectionFade](#) = 0x00000800, [DesktopOptimization_DisableTooltipFade](#) = 0x00001000, [DesktopOptimization_DisableMenuDropShadowEffect](#) = 0x00002000, [DesktopOptimization_DisableFontSmoothing](#) = 0x00004000, [DesktopOptimization_DisableVistaAero](#) = 0x00008000, [DesktopOptimization_DisableOverlappedContent](#) = 0x00010000, [DesktopOptimization_DisableVistaAnimations](#) = 0x00020000 }
- enum [Language_t](#) { [Language_English](#) = 1033, [Language_French](#) = 1036, [Language_German](#) = 1031, [Language_Spanish](#) = 1034 }
- enum [LicenseType_t](#) { [LicenseType_None](#) = 0x00000000, [LicenseType_Network](#) = 0x00000001, [LicenseType_Standalone](#) = 0x00000002, [LicenseType_File](#) = 0x00000003 }
- enum { [NGA_UNKNOWN](#) = -1 }
- enum [MultiguestEvent_t](#) { [MultiguestEvent_InputAssigned](#) = 0x00000001, [MultiguestEvent_InputRevoked](#) = 0x00000002, [MultiguestEvent_InputDenied](#) = 0x00000004, [MultiguestEvent_ConnectionsChanged](#) = 0x00000008, [MultiguestEvent_MultiSessionsSuspended](#) =

- 0x00000010, [MultiguestEvent_MultiSessionsAllowed](#) = 0x00000020,
[MultiguestEvent_MultiSessionsDenied](#) = 0x00000040 }
- enum [SessionStatus_t](#) { [SessionStatus_Idle](#) = 0, [SessionStatus_Connecting](#) = 1, [SessionStatus_Opening](#) = 2, [SessionStatus_Authenticating](#) = 3, [SessionStatus_Starting](#) = 4, [SessionStatus_Running](#) = 5, [SessionStatus_Closing](#) = 6 }
 - enum [InfoWindowReason_t](#) { [InfoWindowReason_Connecting](#) = 1, [InfoWindowReason_Connected](#) = 2, [InfoWindowReason_CancelLogin](#) = 3, [InfoWindowReason_Disconnecting](#) = 4, [InfoWindowReason_Closing](#) = 5 }
 - enum [ErrorCode_t](#) { [NGA_OK](#) = 0x00000000, [NGA_ERROR](#) = 0x00000001, [NGA_ERR_BASE](#) = 0x00010000, [NGA_ERR_INVALID_PARAMETER](#), [NGA_ERR_INVALID_STATUS](#), [NGA_ERR_NOT_OPENED](#), [NGA_ERR_PERMISSION_DENIED](#), [NGA_ERR_NO_SESSION](#) }

Current connection properties

- LONG [CurrentCommProfile](#) [get]
- BSTR [CurrentHostAddress](#) [get]
- LONG [CurrentPortNumber](#) [get]
- BSTR [CurrentGatewayAddress](#) [get]
- BSTR [CurrentHttpProxyAddress](#) [get]
- LONG [CurrentGraphicsMode](#) [get]
- LONG [CurrentGraphicsMaxColors](#) [get, set]
- LONG [CurrentCompressionLevel](#) [get, set]
- LONG [CurrentEncryptionPreferred](#) [get]
- VARIANT_BOOL [IsOpen](#) [get]
- VARIANT_BOOL [IsConnected](#) [get]
- LONG [SessionStatus](#) [get]
- LONG [HostScreenWidth](#) [get]
- LONG [HostScreenHeight](#) [get]
- VARIANT_BOOL [IsMultiguestAdminOnHost](#) [get]
- LONG [NumGuestsOnHost](#) [get]
- [INGuestXFont](#) [ChatFont](#) [get]

Properties for next connection

New values for these properties are not used until next connection.

- LONG [CommProfile](#) [get, set]
- BSTR [HostAddress](#) [get, set]
- LONG [PortNumber](#) [get, set]
- BSTR [GatewayAddress](#) [get, set]
- BSTR [HttpProxyAddress](#) [get, set]
- LONG [CompressionLevel](#) [get, set]
- LONG [EncryptionPreferred](#) [get, set]
- LONG [GraphicsMode](#) [get, set]
- LONG [GraphicsMaxColors](#) [get, set]
- VARIANT_BOOL [LockHostKeyboardOnConnect](#) [get, set]
- VARIANT_BOOL [BlankHostScreenOnConnect](#) [get, set]
- VARIANT_BOOL [GuardHostOnConnect](#) [get, set]

Remote desktop display properties

- LONG [DesktopOptimizeMask](#) [get, set]
- LONG [StretchToFitWindow](#) [get, set]
- VARIANT_BOOL [AutoScroll](#) [get, set]
- LONG [ScrollPositionX](#) [get, set]
- LONG [ScrollPositionY](#) [get, set]
- [INGuestXRcArea RcArea](#) [get]

Keyboard and mouse properties

- LONG [MouseMode](#) [get, set]
- VARIANT_BOOL [ShowRemoteMouseMovements](#) [get, set]
- LONG [KeyboardMode](#) [get, set]
- VARIANT_BOOL [UnicodeKeyboardMode](#) [get, set]
- VARIANT_BOOL [RemoteCursor](#) [get, set]
- VARIANT_BOOL [AutoTakeControl](#) [get, set]

Localization properties

- LONG [Language](#) [get, set]

License properties

- LONG [LicenseType](#) [get, set]
- BSTR [LicenseKey](#) [get, set]
- BSTR [LicenseServer](#) [get, set]
- BSTR [LicenseFile](#) [get, set]
- VARIANT_BOOL [LicenseAutoSave](#) [get, set]

WebConnect properties

- BSTR [WebConnectAddress](#) [get, set]
- BSTR [WebConnectCredentialsAccount](#) [get, set]
- BSTR [WebConnectCredentialsPassword](#) [get, set]
- BSTR [WebConnectCredentialsDomain](#) [get, set]
- BSTR [WebConnectProvidedTicket](#) [get, set]
- BSTR [WebConnectNameQualifier](#) [get, set]

Public Member Functions

- HRESULT [Open](#) ([out, retval] LONG *result)
- HRESULT [Close](#) ([out, retval] LONG *result)
- HRESULT [BeginSession](#) ([in] LONG SessionType,[out, retval] LONG *result)
- HRESULT [EndSession](#) ([in] LONG SessionType,[out, retval] LONG *result)
- HRESULT [SendLoginPassword](#) ([in] BSTR Pwd,[out, retval] LONG *result)
- HRESULT [SendLoginNetop](#) ([in] BSTR GuestId,[in] BSTR Pwd,[in] BSTR NewPassword,[out, retval] LONG *result)
- HRESULT [SendLoginWindows](#) ([in] BSTR UserId,[in] BSTR Domain,[in] BSTR Pwd,[out, retval] LONG *result)

- HRESULT [SendLoginLdap](#) ([in] BSTR Server,[in] BSTR User,[in] BSTR Pwd,[out, retval] LONG *result)
- HRESULT [SendLoginRsa](#) ([in] BSTR UserId,[in] BSTR Pco,[in] BSTR Pwd,[in] BSTR NewPassword,[out, retval] LONG *result)
- HRESULT [SendLoginRsaPin](#) ([in] BSTR Pin,[out, retval] LONG *result)
- HRESULT [CancelLogin](#) ([out, retval] LONG *result)
- HRESULT [SendRefreshScreen](#) ([out, retval] LONG *result)
- HRESULT [SendCtrlAltDel](#) ([out, retval] LONG *result)
- HRESULT [SendCtrlEsc](#) ([out, retval] LONG *result)
- HRESULT [SendAltTab](#) ([in] VARIANT_BOOL bSendAltUp,[out, retval] LONG *result)
- HRESULT [SendAltShiftTab](#) ([in] VARIANT_BOOL bSendAltUp,[out, retval] LONG *result)
- HRESULT [SendAltUp](#) ([out, retval] LONG *result)
- HRESULT [SendGoSolo](#) ([out, retval] LONG *result)
- HRESULT [RequestKeyboardAndMouseControl](#) ([out, retval] LONG *result)
- HRESULT [SendGuardHost](#) ([in] VARIANT_BOOL Guard,[out, retval] LONG *result)
- HRESULT [SendChatMessage](#) ([in] BSTR Message,[out, retval] LONG *result)
- HRESULT [SetCustomString](#) ([in] LONG StringId,[in] BSTR Str,[out, retval] LONG *result)
- HRESULT [GetKeyboardShortcut](#) ([in] LONG ShortcutType,[out, retval] [INGuestXShortcut](#) **result)

Detailed Description

Netop Guest ActiveX Interface.

Member Enumeration Documentation

enum [INGuestXCtrl::CommProfile](#) t

Communication profile.

Enumerator:

CommProfile_TCP 0 - "LAN (TCP)" profile.

CommProfile_HTTP 1 - "HTTP" profile.

CommProfile_UDP 2 - "UDP" profile.

CommProfile_WebConnect 3 - "WebConnect" profile.

enum [INGuestXCtrl::SessionType](#) t

Session types.

Enumerator:

SessionType_RemoteControl 1 - Remote Control Session.

SessionType_Chat 4 - Chat Session.

enum [INGuestXCtrl::MouseMode_t](#)

Mouse mode.

Enumerator:

MouseMode_Local 0 - Only send click and drag to Host.

MouseMode_Remote 1 - Send all mouse events to Host.

MouseMode_None 2 - Do not send mouse events to Host.

enum [INGuestXCtrl::KeyboardMode_t](#)

Keyboard mode.

Enumerator:

KeyboardMode_Local 0 - Don't send special keystrokes.

KeyboardMode_Remote 1 - Send all keystrokes to host.

KeyboardMode_None 2 - No keyboard control.

enum [INGuestXCtrl::StretchMode_t](#)

Remote desktop stretch mode.

Enumerator:

Stretch_FitWindowToHost 0 - Do not stretch, show in actual size.

Stretch_FitHostToWindow 1 - Stretch Host window to fit control.

Stretch_FitNone 2 - Do not stretch.

enum [INGuestXCtrl::GraphicsMode_t](#)

Graphics mode.

Enumerator:

GraphicsMode_Hook 0 - Command (hook) mode.

GraphicsMode_AccBitmap 1 - Accelerated bitmap.

GraphicsMode_NormalBitmap 2 - Normal bitmap.

enum [INGuestXCtrl::MaxColors_t](#)

The limits of colors bitmap graphic modes.

Enumerator:

MaxColors_Actual 0 - Actual Colors.

MaxColors_256 1 - 256 colors.

MaxColors_16 2 - 16 colors.

MaxColors_2 3 - 2 colors.

enum [INGuestXCtrl::CompressionLevel t](#)

The connection compression level.

Enumerator:

CompressionLevel_Auto 0 - Compression level selected automatically.

CompressionLevel_None 1 - None.

CompressionLevel_Low 2 - Low.

CompressionLevel_High 3 - High.

enum [INGuestXCtrl::EncryptionLevel t](#)

The connection encryption level.

Enumerator:

EncryptionLevel_Compatible 0 - Compatible.

EncryptionLevel_None 1 - None.

EncryptionLevel_DataIntegrity 2 - DataIntegrity.

EncryptionLevel_Keyboard 3 - Keyboard.

EncryptionLevel_DataIntegrityAndKeyboard 4 - DataIntegrityAndKeyboard.

EncryptionLevel_High 5 - High.

EncryptionLevel_VeryHigh 6 - Very High.

enum [INGuestXCtrl::DesktopOptimization t](#)

The remote desktop optimization flags.

Enumerator:

DesktopOptimization_DisableEverything 0x00000001 - Disable everything.

DesktopOptimization_DisableWallpaper 0x00000002 - Disable wallpaper.

DesktopOptimization_DisableScreenSaver 0x00000004 - Disable screen saver.

DesktopOptimization_DisableAnimation 0x00000008 - Disable animation.

DesktopOptimization_DisableFullWindowDrag 0x00000010 - Disable full window drag.

DesktopOptimization_DisableMenuAnimation 0x00000020 - Disable menu animation / not supported by current API.

DesktopOptimization_DisableComboboxAnimation 0x00000040 - Disable combobox animation / not supported by current API.

DesktopOptimization_DisableSmoothScrolling 0x00000080 - Disable smooth scrolling / not supported by current API.

DesktopOptimization_DisableGradientCaption 0x00000100 - Disable gradient caption / not supported by current API.

DesktopOptimization_DisableActiveDesktop 0x00000200 - Disable active desktop.

DesktopOptimization_DisableMenuFade 0x00000400 - Disable menu fade / not supported by current API.

DesktopOptimization_DisableSelectionFade 0x00000800 - Disable selection fade / not supported by current API.

DesktopOptimization_DisableTooltipFade 0x00001000 - Disable tooltip fade / not supported by current API.

DesktopOptimization_DisableMenuDropShadowEffect 0x00002000 - Disable drop shadow effect on menus / not supported by current API.

DesktopOptimization_DisableFontSmoothing 0x00004000 - Disable font smoothing feature / not supported by current API.

DesktopOptimization_DisableVistaAero 0x00008000 - Disable Windows Vista Aero / not supported by current API.

DesktopOptimization_DisableOverlappedContent 0x00010000 - Disable overlapped content / not supported by current API.

DesktopOptimization_DisableVistaAnimations 0x00020000 - Disable all animations on Vista / not supported by current API.

enum [INGuestXCtrl::Language](#) t

User interface languages.

Enumerator:

Language_English 1033 - English.

Language_French 1036 - French.

Language_German 1031 - German.

Language_Spanish 1034 - Spanish.

enum [INGuestXCtrl::LicenseType](#) t

License type.

Enumerator:

LicenseType_None 0 - No license.

LicenseType_Network 1 - Network license.

LicenseType_Standalone 2 - Standalone license.

LicenseType_File 3 - Standalone license from file.

anonymous enum

Enumerator:

NGA_UNKNOWN -1 - Unknown property value

enum [INGuestXCtrl::MultiguestEvent](#) **t**

Multi guest event flags.

Enumerator:

MultiguestEvent_InputAssigned 0x00000001 - This Guest is assigned input control

MultiguestEvent_InputRevoked 0x00000002 - This Guest is revoked input control

MultiguestEvent_InputDenied 0x00000004 - This Guest requested input control but it was denied.

MultiguestEvent_ConnectionsChanged 0x00000008 - Number of session changed

MultiguestEvent_MultiSessionsSuspended 0x00000010 - More sessions suspended

MultiguestEvent_MultiSessionsAllowed 0x00000020 - More sessions allowed

MultiguestEvent_MultiSessionsDenied 0x00000040 - Change of sessions denied

enum [INGuestXCtrl::SessionStatus](#) **t**

Session status.

Enumerator:

SessionStatus_Idle 0 - Idle

SessionStatus_Connecting 1 - Connection started

SessionStatus_Opening 2 - Connected, opening a session

SessionStatus_Authenticating 3 - Session can be opened, authenticating

SessionStatus_Starting 4 - Authenticated, initializing RC/Chat

SessionStatus_Running 5 - Session initialized

SessionStatus_Closing 6 - Closing

enum [INGuestXCtrl::InfoWindowReason](#) **t**

Info window reason.

Enumerator:

InfoWindowReason_Connecting 1 - Starting a new connection

InfoWindowReason_Connected 2 - When info window was closed for gateway authentication and now it should be reopened to display the progress of connecting to a host behind the gateway.

InfoWindowReason_CancelLogin 3 - Cancel login button is pressed

InfoWindowReason_Disconnecting 4 - Disconnecting from host

InfoWindowReason_Closing 5 - Closing ActiveX instance

enum [INGuestXCtrl::ErrorCode](#) t

Error codes.

Enumerator:

NGA_OK 0 - Ok

NGA_ERROR 1 - General error (NSDK Dw::Error code)

NGA_ERR_BASE 0x1000 - base for NGA errors

NGA_ERR_INVALID_PARAMETER 0x1001 - Invalid parameter

NGA_ERR_INVALID_STATUS 0x1002 - A session or instance cannot be opened or closed because of the current status

NGA_ERR_NOT_OPENED 0x1003 - The NGA instance is not opened

NGA_ERR_PERMISSION_DENIED 0x1004 - User do not have the right to complete the operation

NGA_ERR_NO_SESSION 0x1005 - There is no session of appropriate type to complete the operation

Member Function Documentation

HRESULT [INGuestXCtrl::Open](#) ([out, retval] LONG * *result*)

Opens NGA control instance.

Opens NGA control instance and change IsOpen property if opened successfully. Instance should be open to create sessions.

This method is synchronous.

The method fires the following events:

- [NGuestXLib:: INGuestXCtrlEvents::OnOpenPre\(\)](#)
- [NGuestXLib:: INGuestXCtrlEvents::OnOpenPost\(\)](#)

[NGuestXLib:: INGuestXCtrlEvents::OnOpenPre](#) event is always followed by [NGuestXLib:: INGuestXCtrlEvents::OnOpenPost\(\)](#) event.

The events are not fired if the instance is already opened.

Returns:

:

- 0 - opened successfully ([NGA_OK](#))
- 1 - failed to open instance ([NGA_ERROR](#))
- [NGA_ERR_INVALID_STATUS](#) - the instance is already opened

HRESULT [INGuestXCtrl::Close](#) ([out, retval] LONG * *result*)

Close NGA control instance.

Ends all active sessions, disconnects from the host, close the NGA control instance and change IsOpen property if closed successfully.

This method is synchronous.

The method fires the following events:

- [NGuestXLib:: INGuestXCtrlEvents::OnClosePre\(\)](#)
- [NGuestXLib:: INGuestXCtrlEvents::OnClosePost\(\)](#)

[NGuestXLib:: INGuestXCtrlEvents::OnClosePre](#) event is always followed by [NGuestXLib:: INGuestXCtrlEvents::OnClosePost\(\)](#) event.

The events are not fired if the instance is already closed.

After having the [Close\(\)](#) method called, the [Open\(\)](#) method can be called once again.

This method is called automatically when ActiveX window is being destroyed.

Returns:

- 0 - closed successfully ([NGA_OK](#))
- 1 - failed to close the instance. The instance is not closed and cannot be opened.
- [NGA_ERR_INVALID_STATUS](#) - the instance is already closed

HRESULT INGuestXCtrl::BeginSession ([in] LONG SessionType, [out, retval] LONG * result)

Initiates a new session.

This function can be used to start a chat session or to resume an RC session when the connection is active or to start a new connection with chat or RC session.

When there is no active connection, the new connection will be established with a host specified by HostAddress, PortNumber, GatewayAddress, HttpProxyAddress, CommProfile properties.

If there is already an active connection, this function either opens a new chat session or resumes an RC session.

The function is asynchronous. The following events can be fired during and after calling this method:

1. [NGuestXLib:: INGuestXCtrlEvents::OnBeginSessionPre\(\)](#)
 2. [NGuestXLib:: INGuestXCtrlEvents::OnConnectPre\(\)](#)
 3. [NGuestXLib:: INGuestXCtrlEvents::OnConnectPost\(\)](#)
 4. [NGuestXLib:: INGuestXCtrlEvents::OnBeginSessionPost\(\)](#)
 5. [NGuestXLib:: INGuestXCtrlEvents::OnSessionStarted\(\)](#)
- [NGuestXLib:: INGuestXCtrlEvents::OnConnectPre\(\)](#), [NGuestXLib:: INGuestXCtrlEvents::OnConnectPost\(\)](#) and [NGuestXLib:: INGuestXCtrlEvents::OnSessionStarted\(\)](#) are fired only when a new connection is established.

When there was an active connection, only [NGuestXLib:: INGuestXCtrlEvents::OnBeginSessionPre\(\)](#) and [NGuestXLib:: INGuestXCtrlEvents::OnBeginSessionPost\(\)](#) events are fired.

[NGuestXLib:: INGuestXCtrlEvents::OnConnectPre\(\)](#) is always followed by [NGuestXLib:: INGuestXCtrlEvents::OnConnectPost\(\)](#).

There is always [NGuestXLib:: INGuestXCtrlEvents::OnBeginSessionPost\(\)](#) for each [NGuestXLib::INGuestXCtrlEvents::OnBeginSessionPre\(\)](#).

None of these event may be produced if the [BeginSession\(\)](#) returns an error.

It is safe to call this function only in some particular states:

- When there is no connection (session status: idle). For example, it is safe to call [BeginSession\(\)](#) in response to the last connection [NGuestXLib:: INGuestXCtrlEvents::OnEndSessionPost\(\)](#) event. A new connection will be created.
- When there is a running Rc or chat session (session status: running) to open session of another type. For example, it is safe to call [BeginSession\(chat\)](#) in response to [NGuestXLib::_INGuestXCtrlEvents::OnSessionStarted\(rc\)](#) or [NGuestXLib::_INGuestXCtrlEvents::EndSessionPost\(chat, true\)](#). A new Rc or Chat session will be opened using the current connection.

In other cases such as when a connection is closing, or when a connection is starting, or when Host requested authentication this function will return an error.

For example in the following code:

```
nga->Open();
nga->BeginSession(Rc);
nga->BeginSession(Chat);
```

the [BeginSession\(Chat\)](#) in most cases will return error because connection and Rc session is not established yet.

Parameters:

SessionType - a session to open:

- [SessionType RemoteControl](#) - Remote control (SU_RemoteControl)
- [SessionType Chat](#) - Text chat (SU_Chat)

Returns:

- 0 - success ([NGA_OK](#)).
- 1 - failed to start session ([NGA_ERROR](#)).
- [NGA_ERR_INVALID_PARAMETER](#) - either some of the connection properties or the parameter is invalid.
- [NGA_ERR_INVALID_STATUS](#) -
 - session of this type is already opened
 - no session can be started at this moment
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.
- [NGA_ERR_PERMISSION_DENIED](#) - a second session cannot be opened because user has no permissions on Host to open a session of the given type.

HRESULT INGuestXCtrl::EndSession ([in] LONG SessionType, [out, retval] LONG * result)

Ends an active session of the given type.

If there is no more active session, this function disconnects the NGA instance from Host.

The function is asynchronous. The following events can be produced after calling this method:

1. [NGuestXLib:: INGuestXCtrlEvents::OnEndSessionPre\(\)](#)
2. [NGuestXLib:: INGuestXCtrlEvents::OnDisconnectPre\(\)](#)
3. [NGuestXLib:: INGuestXCtrlEvents::OnDisconnectPost\(\)](#)
4. [NGuestXLib:: INGuestXCtrlEvents::OnEndSessionPost\(\)](#)

[NGuestXLib:: INGuestXCtrlEvents::OnDisconnectPre\(\)](#) and [NGuestXLib:: INGuestXCtrlEvents::OnDisconnectPost\(\)](#) are fired only when the instance is disconnected from Host.

When there is still an active connection, only [NGuestXLib:: INGuestXCtrlEvents::OnEndSessionPre\(\)](#) and [NGuestXLib:: INGuestXCtrlEvents::OnEndSessionPost\(\)](#) events are fired.

[NGuestXLib:: INGuestXCtrlEvents::OnDisconnectPre\(\)](#) is always followed by [NGuestXLib:: INGuestXCtrlEvents::OnDisconnectPost\(\)](#).

There is always [NGuestXLib:: INGuestXCtrlEvents::OnEndSessionPost\(\)](#) for each [NGuestXLib:: INGuestXCtrlEvents::OnEndSessionPre\(\)](#).

None of these event can be produced if the function returns an error.

It is safe to call this function only in some particular states:

- There is an active connection and running session of the given type (session status: running). For example in response to the [NGuestXLib:: INGuestXCtrlEvents::OnSessionStarted\(\)](#) event.
- There is already an active connection but the session is not authenticated yet (session status: authenticating). For example in response to [NGuestXLib:: INGuestXCtrlEvents::OnBeginSessionPost](#) event.

In other states (e.g. connecting, closing) the function will return an error.

For example in the following code:

```
// nga is not connected yet
if (nga->BeginSession(Rc) == 0) // start new connection
    nga->EndSession(Rc)
```

the EndSession(Rc) will return an error because the Rc session wasn't opened yet.

Parameters:

SessionType - a session to open

- [SessionType RemoteControl](#) - Remote control (SU_RemoteControl)
- [SessionType Chat](#) - Text chat (SU_Chat)

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to close session ([NGA_ERROR](#))
- [NGA_ERR_INVALID_PARAMETER](#) - the parameter is invalid
- [NGA_ERR_INVALID_STATUS](#) -
 - there is no opened session of specified type
 - session can not be finished at this moment
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.

HRESULT INGuestXCtrl::SendLoginPassword ([in] BSTR *Pwd*, [out, retval] LONG * *result*)

Sends the password credentials to Host.

This function shall only be called on [NGuestXLib:: INGuestXCtrlEvents::OnLoginPassword\(\)](#) event.

The function is asynchronous.

Parameters:

Pwd - The password must not be NULL and not longer than 16 characters.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_INVALID_PARAMETER](#) - the parameter is invalid
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.
- [NGA_ERR_INVALID_STATUS](#) - the authentication cannot be sent at this moment

HRESULT INGuestXCtrl::SendLoginNetop ([in] BSTR *GuestId*, [in] BSTR *Pwd*, [in] BSTR *NewPassword*, [out, retval] LONG * *result*)

Sends the Netop credentials to Host.

This function shall only be called on [NGuestXLib::_INGuestXCtrlEvents::OnLoginNetop\(\)](#) event.

The function is asynchronous.

Parameters:

GuestId - The user ID must not be NULL and not longer than 32 characters.

Pwd - The password must not be NULL and not longer than 16 characters.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_INVALID_PARAMETER](#) - a parameter is invalid
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.
- [NGA_ERR_INVALID_STATUS](#) - the authentication cannot be sent at this moment

HRESULT INGuestXCtrl::SendLoginWindows ([in] BSTR *UserId*, [in] BSTR *Domain*, [in] BSTR *Pwd*, [out, retval] LONG * *result*)

Sends Windows system credentials to Host.

This function shall only be called on [NGuestXLib::_INGuestXCtrlEvents::OnLoginNetop\(\)](#) event.

The function is asynchronous.

Parameters:

UserId - The user ID must not be NULL and not longer than 512 characters.

Domain - The domain must not be NULL and not longer than 512 characters.

Pwd - The password must not be NULL and not longer than 512 characters.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_INVALID_PARAMETER](#) - a parameter is invalid
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.
- [NGA_ERR_INVALID_STATUS](#) - the authentication cannot be sent at this moment

HRESULT INGuestXCtrl::SendLoginLdap ([in] BSTR *Server*, [in] BSTR *User*, [in] BSTR *Pwd*, [out, retval] LONG * *result*)

Sends LDAP credentials to Host.

This shall only be called on [NGuestXLib:: INGuestXCtrlEvents::OnLoginLdap\(\)](#) event.

The function is asynchronous.

Parameters:

Server - The server ID must not be NULL and not longer than 512 characters.

User - The user ID must not be NULL and not longer than 512 characters.

Pwd - The password must not be NULL and not longer than 512 characters.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_INVALID_PARAMETER](#) - a parameter is invalid
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.
- [NGA_ERR_INVALID_STATUS](#) - the authentication cannot be sent at this moment

HRESULT INGuestXCtrl::SendLoginRsa ([in] BSTR *UserId*, [in] BSTR *Pco*, [in] BSTR *Pwd*, [in] BSTR *NewPassword*, [out, retval] LONG * *result*)

Sends RSA credentials to Host.

This function shall only be called on [NGuestXLib:: INGuestXCtrlEvents::OnLoginRsa\(\)](#) event.

The function is asynchronous.

Parameters:

UserId - The user ID must not be NULL and not longer than 32 characters.

Pco - The RSA SecurID passcode must not be NULL and not longer than 16 characters.

Pwd - The optional password. May be be NULL. Must not be longer then 16 chars.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_INVALID_PARAMETER](#) - a parameter is invalid
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.
- [NGA_ERR_INVALID_STATUS](#) - the authentication cannot be sent at this moment

HRESULT INGuestXCtrl::SendLoginRsaPin ([in] BSTR *Pin*, [out, retval] LONG * *result*)

Sends RSA SecurID pin code to Host.

Sends an RSA SecurID pin code. This shall only be called on [NGuestXLib:: INGuestXCtrlEvents::OnEnterRsaPincode\(\)](#) event.

The function is asynchronous.

Parameters:

Pin - The pin code.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_INVALID_PARAMETER](#) - a parameter is invalid
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.
- [NGA_ERR_INVALID_STATUS](#) - the authentication cannot be sent at this moment

HRESULT INGuestXCtrl::CancelLogin ([out, retval] LONG * result)

Cancels the authentication on a Gateway or Host.

Can be used to cancel the authentication on a gateway or host. When canceling the authentication on Host, the [NGuestXLib:: INGuestXCtrlEvents::OnLoginFailed\(\)](#) is fired.

This function should be called on in response to OnLogin events:

- [NGuestXLib:: INGuestXCtrlEvents::OnLoginPassword\(\)](#)
- [NGuestXLib:: INGuestXCtrlEvents::OnLoginNetop\(\)](#)
- [NGuestXLib:: INGuestXCtrlEvents::OnLoginWindows\(\)](#)
- [NGuestXLib:: INGuestXCtrlEvents::OnLoginLdap\(\)](#)
- [NGuestXLib:: INGuestXCtrlEvents::OnLoginRsa\(\)](#)
- [NGuestXLib:: INGuestXCtrlEvents::OnEnterRsaPincode\(\)](#)

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.
- [NGA_ERR_INVALID_STATUS](#) - the authentication cannot be canceled at this moment

HRESULT INGuestXCtrl::SendRefreshScreen ([out, retval] LONG * result)

Forces the Host to resend its screen.

This function forcefully refreshes RC screen.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_INVALID_STATUS](#) - there is no running Rc session.

HRESULT INGuestXCtrl::SendCtrlAltDel ([out, retval] LONG * result)

Sends Ctrl-Alt-Del keystroke to Host.

This function sends both key down and up scancodes.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_NO_SESSION](#) - there is no running Rc session.
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.

HRESULT INGuestXCtrl::SendCtrlEsc ([out, retval] LONG * *result*)

Sends Ctrl-Esc keystroke to Host.

This function sends both down and up scancodes.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_NO_SESSION](#) - there is no running Rc session.
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.

HRESULT INGuestXCtrl::SendAltTab ([in] VARIANT_BOOL *bSendAltUp*, [out, retval] LONG * *result*)

Sends Alt+Tab keystroke to Host.

Function sends scancodes for Alt-Tab key down and key up. To prevent the function from sending Alt key up scancode the *bSendAltUp* parameter can be set to FALSE.

Parameters:

bSendAltUp - when false the function do not send Alt up scancode.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_NO_SESSION](#) - there is no running Rc session.
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.

HRESULT INGuestXCtrl::SendAltShiftTab ([in] VARIANT_BOOL *bSendAltUp*, [out, retval] LONG * *result*)

Sends Alt-Shift-Tab keystroke to Host.

Function sends scancodes for Alt-Shift-Tab key down and key up. To prevent the function from sending Alt key up scancode the *bSendAltUp* parameter can be set to FALSE.

Parameters:

UpDown - If TRUE, send down+up scancodes, otherwise only down.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_NO_SESSION](#) - there is no running Rc session.
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.

HRESULT INGuestXCtrl::SendAltUp ([out, retval] LONG * *result*)

Sends Alt key up scan code to Host.

This function can be use to send a key up scan code for Alt button to Host when the Alt up scan code was not sent by [SendAltTab\(\)](#) or [SendAltShiftTab\(\)](#).

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_NO_SESSION](#) - there is no running Rc session.
- [NGA_ERR_NOT_OPENED](#) - the NGA instance is not opened with [Open\(\)](#) method.

HRESULT INGuestXCtrl::SendGoSolo ([out, retval] LONG * *result*)

Sends the Go Solo command to Host.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command.
- [NGA_ERR_NOT_OPENED](#) - NGA instance is not opened
- [NGA_ERR_NO_SESSION](#) - there is no session of the appropriate type.
- [NGA_ERR_PERMISSION_DENIED](#) - returned when the Guest is not an Administrator (Power User) and hence don't have the right for this command.

HRESULT INGuestXCtrl::RequestKeyboardAndMouseControl ([out, retval] LONG * *result*)

Sends Request Keyboard And Mouse Control command to Host.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command.
- [NGA_ERR_NOT_OPENED](#) - NGA instance is not opened
- [NGA_ERR_NO_SESSION](#) - there is no session of the appropriate type.
- [NGA_ERR_PERMISSION_DENIED](#) - returned when the Guest is not an Administrator (Power User) and hence don't have the right for this command.

HRESULT INGuestXCtrl::SendGuardHost ([in] VARIANT_BOOL *Guard*, [out, retval] LONG * *result*)

Sends Guard command to Host.

Parameters:

Guard true - to prevent further guest connections false - to enable further guest connections

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send the command ([NGA_ERROR](#))
- [NGA_ERR_NOT_OPENED](#) - NGA instance is not opened
- [NGA_ERR_NO_SESSION](#) - there is no session of the appropriate type.

- [NGA_ERR_PERMISSION_DENIED](#) - returned when the Guest is not an Administrator (Power User) and hence don't have the right for this command.

HRESULT INGuestXCtrl::SendChatMessage ([in] BSTR *Message*, [out, retval] LONG * *result*)

Sends chat message to Host.

Parameters:

Message - a chat message to send to host.

Font - a font of chat message

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command.
- [NGA_ERR_NOT_OPENED](#) - NGA instance is not opened
- [NGA_ERR_NO_SESSION](#) - there is no chat session.

HRESULT INGuestXCtrl::SetCustomString ([in] LONG *StringId*, [in] BSTR *Str*, [out, retval] LONG * *result*)

Overrides the given GUI string.

Parameters:

StringId - The Id of GUI string to override.

Str - New GUI string. Passing a NULL string will remove the overridden string.

Returns:

- 0 - success ([NGA_OK](#))
- 1 - failed to send a command ([NGA_ERROR](#))
- [NGA_ERR_INVALID_PARAMETER](#) - unknown string id

HRESULT INGuestXCtrl::GetKeyboardShortcut ([in] LONG *ShortcutType*, [out, retval] [INGuestXShortcut](#) ** *result*)

Keyboard shortcut interface.

This method can be used to set/get ActiveX keyboard shortcuts, for example "Send Alt-Ctrl-Del to Host", "Send Ctrl-Esc to Host", etc.

Parameters:

ShortcutType - the ID of the shortcut to return (one of the [INGuestXShortcut::ShortcutType_t](#) constants)

Returns:

button assignments for the given shortcut. See [INGuestXShortcut](#) interface description for more details.

Property Documentation

LONG INGuestXCtrl::CurrentCommProfile [get]

The communication profile of the current connection, read only.

When there is no active connection the [NGA_UNKNOWN](#) is always returned.

BSTR INGuestXCtrl::CurrentHostAddress [get]

The host address of the current connection, read only.

When there is no connection the empty string is returned.

LONG INGuestXCtrl::CurrentPortNumber [get]

The port number of the current connection, read only.

The value 0 means that default port for current communication profile should be used.

When there is no active connection [NGA_UNKNOWN](#) is returned.

BSTR INGuestXCtrl::CurrentGatewayAddress [get]

The address of the gateway for the current connection, read only.

When there is no active connection the empty string is returned.

BSTR INGuestXCtrl::CurrentHttpProxyAddress [get]

The address of HTTP proxy for the current connection, read only.

When there is no connection the empty string is returned.

LONG INGuestXCtrl::CurrentGraphicsMode [get]

The graphic mode of the current connection, read only.

One of the [GraphicsMode_t](#) constant can be assigned to this property.

When there is no active connection the [NGA_UNKNOWN](#) is returned.

LONG INGuestXCtrl::CurrentGraphicsMaxColors [get, set]

The limit of bitmap mode colors for the current connection, read/write.

One of the [MaxColors_t](#) constant can be assigned to this property.

Used only for bitmap modes (e.g. when GraphicsMode property is either [GraphicsMode_AccBitmap](#) or [GraphicsMode_NormalBitmap](#)).

When there is no active connection the [NGA_UNKNOWN](#) is returned.

LONG INGuestXCtrl::CurrentCompressionLevel [get, set]

The compression level of the current connection, read/write.

One of the [CompressionLevel_t](#) constants can be assigned to this property.

When there is no active connection the [NGA_UNKNOWN](#) is returned.

LONG INGuestXCtrl::CurrentEncryptionPreferred [get]

The encryption level of the current connection, read/write.

One of the [EncryptionLevel_t](#) constants can be assigned to this property.

When there is no active connection the [NGA_UNKNOWN](#) is returned.

VARIANT_BOOL INGuestXCtrl::IsOpen [get]

Whether the instance of NGA was opened successfully by [Open\(\)](#) function, read only.

VARIANT_BOOL INGuestXCtrl::IsConnected [get]

Whether the instance of NGA is connected to remote Host, read only.

Property is true when there is an active session (chat or RC).

LONG INGuestXCtrl::SessionStatus [get]

Current status of RC session, read only.

Status: Can be one of the [SessionStatus_t](#) constants. When the NGA instance is not open, status is [SessionStatus_Idle](#).

LONG INGuestXCtrl::HostScreenWidth [get]

The current width of the remote desktop, read only.

When there is no active RC session the [NGA_UNKNOWN](#) is returned.

LONG INGuestXCtrl::HostScreenHeight [get]

The current height of the remote desktop, read only.

When there is no active RC session the [NGA_UNKNOWN](#) is returned.

VARIANT_BOOL INGuestXCtrl::IsMultiguestAdminOnHost [get]

Whether the current RC session has multi guest admin role on Host, read only.

When there is no active session the false is returned.

LONG INGuestXCtrl::NumGuestsOnHost [get]

The number of guests connected to Host, read only.

When there is no active session the [NGA_UNKNOWN](#) is returned.

[INGuestXFont](#) INGuestXCtrl::ChatFont [get]

Get chat font interface, read only.

LONG INGuestXCtrl::CommProfile [get, set]

Communication profile for a next connection.

One of the [CommProfile_t](#) constants can be assigned to this property.

Default value: [CommProfile_TCP](#)

[BeginSession\(\)](#) method use this profile when establishing a new connection.

BSTR INGuestXCtrl::HostAddress [get, set]

The address of remote host for a next connection.

[BeginSession\(\)](#) method uses this address when establishing a new connection. Default value: empty string

LONG INGuestXCtrl::PortNumber [get, set]

Port number for a next connection.

The value 0 means that default port for current communication profile should be used.

[BeginSession\(\)](#) method use this property when establishing a new connection. Default value: 0

BSTR INGuestXCtrl::GatewayAddress [get, set]

Address of gateway for a next connection.

When empty string is specified the gateway is not used. This property is ignored when UDP communication profile is selected.

[BeginSession\(\)](#) method use this address when establishing a new connection. Default value: empty string

BSTR INGuestXCtrl::HttpProxyAddress [get, set]

Address of HTTP proxy for a next connection.

The proxy address is ignored when UDP or TCP communication profile is selected.

[BeginSession\(\)](#) method use this address when establishing a new connection. Default value: empty string

LONG INGuestXCtrl::CompressionLevel [get, set]

The compression level for a next connection.

One of the [CompressionLevel_t](#) constants can be assigned to this property.

[BeginSession\(\)](#) method uses this mode when establishing a new connection.

Default value: [CompressionLevel_Auto](#) - select compression level automatically

LONG INGuestXCtrl::EncryptionPreferred [get, set]

The encryption level for a next connection.

One of the [EncryptionLevel_t](#) constants can be assigned to this property.

[BeginSession\(\)](#) method uses this mode when establishing a new connection.

Default value: [EncryptionLevel_Compatible](#) - Netop 6.5 compatible encryption

LONG INGuestXCtrl::GraphicsMode [get, set]

The graphic mode for a next connection.

One of the [GraphicsMode_t](#) constant can be assigned to this property.

[BeginSession\(\)](#) method use this mode when establishing a new connection.

LONG INGuestXCtrl::GraphicsMaxColors [get, set]

The limit of bitmap mode colors for a next connection.

One of the [MaxColors_t](#) constant can be assigned to this property.

Used only for bitmap modes (e.g. when GraphicsMode property is either [GraphicsMode_AccBitmap](#) or [GraphicsMode_NormalBitmap](#)).

[BeginSession\(\)](#) method uses this mode when establishing a new connection.

Default value: [MaxColors_Actual](#) - use actual colors

VARIANT_BOOL INGuestXCtrl::LockHostKeyboardOnConnect [get, set]

Keyboard locking mode for a next connection.

[BeginSession\(\)](#) method uses this mode when establishing a new connection.

Default value: FALSE (do not lock)

VARIANT_BOOL INGuestXCtrl::BlankHostScreenOnConnect [get, set]

Host screen blanking mode for a next connection.

[BeginSession\(\)](#) method uses this mode when establishing a new connection.

Default value: FALSE (do not blank)

VARIANT_BOOL INGuestXCtrl::GuardHostOnConnect [get, set]

Host guard settings for a next connection.

[BeginSession\(\)](#) method uses this mode when establishing a new connection.

Default value: FALSE (do not guard)

LONG INGuestXCtrl::DesktopOptimizeMask [get, set]

Desktop optimization mask for the current and next connection.

A bitwise OR of the [DesktopOptimization_t](#) constants can be assigned to this property.

Changing this property will affect current RC session immediately. Same setting will be used for next RC session.

Default value: [DesktopOptimization_DisableEverything](#) - disable everything

LONG INGuestXCtrl::StretchToFitWindow [get, set]

Remote desktop stretch mode.

Property indicate how the remote desktop image is displayed inside NGA control. Changing this property with active RC session will redraw the NGA control.

One of the [StretchMode_t](#) constants can be assigned to this property.

Default value: [Stretch_FitWindowToHost](#)

VARIANT_BOOL INGuestXCtrl::AutoScroll [get, set]

Auto scroll mode.

The scroll is done when the Mouse enters a hot zone close to the border (1/10 of the width or height in each side (left/right/top/bottom) of the RC window.

Default value: TRUE (enabled).

LONG INGuestXCtrl::ScrollPositionX [get, set]

The horizontal position of the remote desktop image inside NGA control.

Default value: 0

LONG INGuestXCtrl::ScrollPositionY [get, set]

The vertical position of the remote desktop image inside NGA control.

Default value: 0

INGuestXRcArea INGuestXCtrl::RcArea [get]

Rc area interface (read only property).

This property can be used to set/get the remote control area to be shown in the control. Changing this property does not affect the remote control area of the current connection, settings will be used for next connections.

See [INGuestXRcArea](#) interface description for more details.

LONG INGuestXCtrl::MouseMode [get, set]

Mouse mode of current and next RC session.

One of the [MouseMode_t](#) constants can be assigned to this property.

Default value: [MouseMode_Remote](#)

VARIANT_BOOL INGuestXCtrl::ShowRemoteMouseMovements [get, set]

Gets or sets the value of remote mouse movements property.

When this property is true the remote desktop mouse movements are shown when the control is focused.

Default value: false (do not show remote mouse movements)

LONG INGuestXCtrl::KeyboardMode [get, set]

The keyboard mode of the current and next RC session.

One of the [KeyboardMode_t](#) constants can be assigned to this property.

Default value: [KeyboardMode_Local](#)

VARIANT_BOOL INGuestXCtrl::UnicodeKeyboardMode [get, set]

Indicates whether keyboard events will be sent as unicode characters or as scan codes.

Returns:

false - NGA sends scancodes true - NGA sends unicode characters

Default value: : false (Send scan codes)

VARIANT_BOOL INGuestXCtrl::RemoteCursor [get, set]

The remote cursor display mode.

When 'true' NGA mouse cursor have the shape of the Host mouse cursor when displayed in the NGA control. The shape of the cursor is not stretched when the remote desktop mode is stretched to fit the screen.

Default value: true

VARIANT_BOOL INGuestXCtrl::AutoTakeControl [get, set]

Gets or sets the auto take control property.

When several guests are connected to the same host only one of these guests controls Host's keyboard and mouse. When this option is enabled, Guest requests the control over Host keyboard and mouse automatically on keyboard or mouse hit.

Default value: true

LONG INGuestXCtrl::Language [get, set]

Gets or sets the language used for build-in dialogs.

One of the [Language t](#) constants can be assigned to this property.

When attempting to assign unsupported language value, the current UI language is not changed.

Default value: selected in accordance with system locale. If the current system locale is not supported, the [Language English](#) language is used.

LONG INGuestXCtrl::LicenseType [get, set]

Sentinel license type (network, standalone, etc).

One of the [LicenseType t](#) constants can be assigned to this property.

When attempting to assign unsupported license type value, the current License type is not changed.

Default value: LicenseType_None

BSTR INGuestXCtrl::LicenseKey [get, set]

The Sentinel license key.

ActiveX uses this property when the license type is standalone. The string property contains the license key (not the file name).

Default value: empty string

BSTR INGuestXCtrl::LicenseServer [get, set]

The IP/hostname of the Sentinel license server.

ActiveX uses this property when the License Type is network.

Default value: empty string

BSTR INGuestXCtrl::LicenseFile [get, set]

The Sentinel license file.

ActiveX uses this property when the license type is standalone file. The string property contains the full path to the license file.

Default value: empty string

VARIANT_BOOL INGuestXCtrl::LicenseAutoSave [get, set]

The Sentinel license properties autosave flag.

ActiveX checks this property when some license property is changed and if flag is TRUE property value is saved into registry. The setting of this flag affects only properties changed after flag was set.

Default value: TRUE

BSTR INGuestXCtrl::WebConnectAddress [get, set]

Gets or sets the WebConnect address for a next connection.

This property is ignored when communication profile other than WebConnect is selected.

[BeginSession\(\)](#) method use this address when establishing a new connection. Default value: empty string

BSTR INGuestXCtrl::WebConnectCredentialsAccount [get, set]

Gets or sets the WebConnect credentials account for a next connection.

This property is ignored when communication profile other than WebConnect is selected.

[BeginSession\(\)](#) method use this address when establishing a new connection. Default value: empty string

BSTR INGuestXCtrl::WebConnectCredentialsPassword [get, set]

Gets or sets the WebConnect credentials password for a next connection.

This property is ignored when communication profile other than WebConnect is selected.

[BeginSession\(\)](#) method use this address when establishing a new connection. Default value: empty string

BSTR INGuestXCtrl::WebConnectCredentialsDomain [get, set]

Gets or sets the WebConnect credentials domain for a next connection.

This property is ignored when communication profile other than WebConnect is selected.

[BeginSession\(\)](#) method use this address when establishing a new connection. Default value: empty string

BSTR INGuestXCtrl::WebConnectProvidedTicket [get, set]

Gets or sets the WebConnect provided ticket for next connection.

This property is ignored when communication profile other than WebConnect is selected.

[BeginSession\(\)](#) method use this address when establishing a new connection.
Default value: empty string

BSTR INGuestXCtrl::WebConnectNameQualifier [get, set]

Gets or sets the WebConnect name qualifier for next connection.

This property is ignored when communication profile other than WebConnect is selected.

[BeginSession\(\)](#) method use this address when establishing a new connection. Default value: "HST"

INGuestXEventParam Interface Reference

Auxiliary parameter interface for INGuestX events.

Properties

- VARIANT_BOOL [Handled](#) [get, set]
- VARIANT_BOOL [Canceled](#) [get, set]

Detailed Description

Auxiliary parameter interface for INGuestX events.

This is an internal interface and cannot be created via CoCreateInstance() function. Scripting languages can change Handled property provided in this interface to indicate that an event was handled successfully and default NGuestX action should be suppressed.

Example of using this property from javascript language:

```
function nguestx_OnLoginPassword(reason, eventparam)
{
    nguestx.SendLoginPassword("1");
    eventparam.Handled = 1; // Set event handled
}
```

Property Documentation

VARIANT_BOOL INGuestXEventParam::Handled [get, set]

Indicates that an event has been handled by an application.

Event handler can set this property to true to suppress default NGuestX action.

VARIANT_BOOL INGuestXEventParam::Canceled [get, set]

Indicates that an action should be canceled.

Event handler can set this property to true to indicate that processing should not continue.

INGuestXFont Interface Reference

Chat font interface.

Public Types

- enum [FontEffect_t](#) { [FontEffect_Italic](#) = 1, [FontEffect_StrikeOut](#) = 2 }

Properties

- LONG [Height](#) [get, set]
- LONG [Weight](#) [get, set]
- LONG [CharSet](#) [get, set]
- LONG [Effects](#) [get, set]
- VARIANT_BOOL [Underline](#) [get, set]
- LONG [FgColor](#) [get, set]
- LONG [BgColor](#) [get, set]
- BSTR [Name](#) [get, set]

Detailed Description

Chat font interface.

Member Enumeration Documentation

enum [INGuestXFont::FontEffect_t](#)

Font effect.

Enumerator:

FontEffect_Italic 1 - Italic

FontEffect_StrikeOut 2 - Strike out

Property Documentation

LONG [INGuestXFont::Height](#) [get, set]

Font height.

LONG [INGuestXFont::Weight](#) [get, set]

Font weight.

LONG [INGuestXFont::CharSet](#) [get, set]

Font character set.

LONG INGuestXFont::Effects [get, set]

Font effects.

The bitwise OR of the [FontEffect_t](#) constants.

VARIANT_BOOL INGuestXFont::Underline [get, set]

Font underline.

LONG INGuestXFont::FgColor [get, set]

Font foreground color.

LONG INGuestXFont::BgColor [get, set]

Font background color.

BSTR INGuestXFont::Name [get, set]

Font name property.

INGuestXRcArea Interface Reference

RC area interface.

Public Types

- enum [RcAreaMode_t](#) { [RcAreaMode_None](#) = 0, [RcAreaMode_Rect](#) = 1, [RcAreaMode_Monitor](#) = 2 }

Properties

- LONG [Mode](#) [get, set]
- LONG [Monitor](#) [get, set]
- LONG [Top](#) [get, set]
- LONG [Left](#) [get, set]
- LONG [Width](#) [get, set]
- LONG [Height](#) [get, set]

Detailed Description

RC area interface.

Member Enumeration Documentation

enum [INGuestXRcArea::RcAreaMode_t](#)

RC area modes.

Enumerator:

RcAreaMode_None 0 - None

RcAreaMode_Rect 1 - Rectangle

RcAreaMode_Monitor 2 - Monitor

Property Documentation

LONG [INGuestXRcArea::Mode](#) [get, set]

RC area mode:

- 0 ([RcAreaMode_None](#)) - Show entire remote control screen.
- 1 ([RcAreaMode_Rect](#)) - Show only rectangular area specified by Top, Left, Width, Height properties.
- 2 ([RcAreaMode_Monitor](#)) - Show only Host monitor specified by Monitor property.

LONG [INGuestXRcArea::Monitor](#) [get, set]

RC area monitor.

LONG INGuestXRcArea::Top [get, set]

RC area rectangle top.

LONG INGuestXRcArea::Left [get, set]

RC area rectangle left side coordinate.

LONG INGuestXRcArea::Width [get, set]

RC area rectangle width.

LONG INGuestXRcArea::Height [get, set]

RC area rectangle height.

INGuestXShortcut Interface Reference

Keyboard shortcut interface.

Public Types

- enum [ShortcutType_t](#) { [ShortcutType_AltCtrlDel](#) = 0, [ShortcutType_CtrlEsc](#) = 1, [ShortcutType_AltTab](#) = 2, [ShortcutType_Status](#) = 3 }

Properties

- VARIANT_BOOL [Alt](#) [get, set]
- VARIANT_BOOL [Ctrl](#) [get, set]
- LONG [VkCode](#) [get, set]

Detailed Description

Keyboard shortcut interface.

Defines the button assignments for a shortcut returned by [INGuestXCtrl::GetKeyboardShortcut\(\)](#) method.

Member Enumeration Documentation

enum [INGuestXShortcut::ShortcutType_t](#)

Keyboard shortcut types.

Enumerator:

ShortcutType_AltCtrlDel 0 - Send ALT + CTRL + DEL to Host

ShortcutType_CtrlEsc 1 - Send CTRL + ESC to Host

ShortcutType_AltTab 2 - Send ALT + TAB to Host

ShortcutType_Status 3 - Connection Status Dialog

Property Documentation

VARIANT_BOOL [INGuestXShortcut::Alt](#) [get, set]

ALT usage in shortcut:

- TRUE - ALT is used.
- FALSE - ALT is not used.

VARIANT_BOOL [INGuestXShortcut::Ctrl](#) [get, set]

CTRL usage in shortcut:

- TRUE - CTRL is used.

- FALSE - CTRL is not used.

LONG INGuestXShortcut::VkCode [get, set]

VK code that should be used in shortcut.

Allowed VK codes are:

- A-Z, 0-9, F1-F12, VK_INSERT, VK_HOME, VK_END, VK_PRIOR (Page Up), VK_NEXT (Page Down), VK_UP, VK_DOWN, VK_LEFT, VK_RIGHT.
- To disable the shortcut set this property to -1.