

Programmer's Reference

Attachmate®
INFOCONNECT.
Enterprise Edition

OCX for Windows®

Copyrights and Notices

Attachmate® INFOConnect® Enterprise Edition

© 2011 Attachmate Corporation. All Rights Reserved.

Patents

This Attachmate software is protected by U.S. patents 6252607 and 6803914.

Trademarks

Attachmate, the Attachmate logo, CryptoConnect, FileXpress, and PEPgate are either registered trademarks or trademarks of Attachmate Corporation in the USA. INFOConnect is a registered trademark of Unisys Corporation. FIPS 140-1 Validated is a certification mark of NIST, which does not imply product endorsement by NIST, the U.S. or Canadian Governments. All other trademarks, trade names, or company names referenced in product materials are used for identification only and are the property of their respective owners.

Attachmate Software License Agreement

A copy of the Attachmate software license agreement governing this product can be found in a 'license' file in the root directory of the product.

Licensors

Attachmate Corporation
1500 Dexter Avenue North
Seattle, WA 98109 USA
USA
+1.206.217.7100
<http://www.attachmate.com>

Third-Party Notices

Third Party Terms and notices are provided in a 'thirdpartynotices' file in the root directory of the product.

Contents

	About the Documentation	vii
	Conventions	viii
	Related Documentation	ix
	Readme Files	ix
	Guides	ix
	Online Help	ix
Chapter 1	Introduction	1
	What is the INFOConnect OLE Custom Control?	2
	Data Filters	2
	How the INFOConnect OCX Works	3
	Developing Applications	3
	Running Applications	3
	Overview of the Methods, Events, and Properties	4
Chapter 2	Guidelines for Developing Applications	7
	Prerequisites	8
	Converting 16-bit VBX Applications to 32-bit OCX	9
	Adding the INFOConnect OCX to a Project	12
	Defining Properties	13
	Defining Event Procedures	14
	SessionEstablishedEvent	14
	SessionClosed Event	14

Contents

	TransmitDone Event	14
	ReceiveDone Event	14
	RptMsg Event	15
	Status Event	16
	UserEvent Event	16
	Adding a Timer	17
	Reviewing the Sample Application Source Code	19
	Running the Sample Application	21
Chapter 3	Methods, Events, and Properties	23
	AboutBox Method	26
	Action Property	27
	AutoReceive Property	30
	FireUserEvent Method	32
	GetPathNames Method	33
	hSession Property	36
	hWndVBList Property	37
	IsEstablished Property	38
	Open Property	39
	OpenAccessory Method	41
	PathName Property	44
	Receive Method	45
	ReceiveDone Event	47
	RptMsg Event	48
	SendStatus Method	49
	SessionClosed Event	50
	SessionEstablished Event	51
	SetPathList Method	52
	Status Event	53
	StatusMessage Property	54
	Transmit Method	55
	TransmitDone Event	56
	UserEvent Event	57
	XmitData Property	58
Appendix A	Transferring Data Between a PC and a Host	59
	What Is a Data Filter?	60
	Using the Sample Data Filter	61

Appendix B	RptMsg Messages	63
	INFOConnect OCX Errors	64
	INFOConnect Termination Errors	66
	INFOConnect Severe Errors	69
Appendix C	Status Messages	81
	Status Messages Sent from an Accessory to a Library	82
	Blocking Status Messages	82
	Connection Status Messages	82
	Function Key Status Messages	83
	Focus Shift Status Messages	83
	Status Messages Sent from a Library to an Accessory	85
	Connection Status Messages	85
	Control Status Messages	86
	Line State Status Messages	86
	Status Messages Sent From the INFOConnect Manager to an Accessory	87
	Communication Manager Status Messages	87
	UTS-Specific Status Messages	88
Appendix D	Troubleshooting	91
	General Troubleshooting Procedures	92
	Glossary	93
	Index	97

About the Documentation

This guide describes the INFOConnect™ OLE custom control (OCX). It includes programming guidelines and describes the methods, events, and properties that are supported by this control.

This guide is intended for application developers who are already familiar with a programming language (such as Visual Basic®, Microsoft® C, or Microsoft C++) and who understand INFOConnect concepts such as paths and libraries.

Although the INFOConnect OCX can be used with any OLE-aware application compiler, the examples in this book are limited to Visual Basic.

This preface includes the following headings:

Conventions	viii
Related Documentation	ix

Conventions

The following format and style conventions are used in this guide:

- Text that you type as well as messages and prompts that appear on the screen appear in *this type style*.
- In addition to emphasizing text and highlighting terms used for the first time, *italics* indicate variables. For example, if you were asked to type *drive:\directory\filename.ext*, you would enter the actual drive, directory, and file name in place of the italicized words.

- An underscore (`_`) indicates when a particular line of code is continued to the next line, as in the following example:

```
Private Sub Icsctl32_ReceiveDone (RcvData As _  
String, Length As Long)
```

- All Visual Basic procedures are for version 4.0. If you are using a different version, your procedures might vary.
- The word *PC* refers to any personal computer running Windows® 95 or Windows NT®.
- The word *host* refers to any mainframe, mini-computer, or information hub with which the PC communicates.

Related Documentation

Additional information exists in the form of a Readme file, guides, and online Help.

Readme Files

README.OCC contains important notices, known limitations, and the latest information about the INFOConnect OCX that could not be included in this guide.

If you double-click this file in My Computer or File Manager, your PC might not recognize the file extension. Select Notepad as the application to use to read the file.

Guides

The INFOConnect OCX comes with the following additional documentation:

- The *Getting Started* guide explains how to install your INFOConnect products and get them up and running.
- The *INFOConnect Connectivity Services Installation, Configuration, and Operations Guide* provides information about INFOConnect paths, path templates, libraries, and accessories.

Online Help

The INFOConnect OCX comes with an online Help file (ICSCTL32.HLP). This file includes most of the information included in this guide.

To access the online Help, use My Computer or File Manager to go to the OCX32 subfolder of your INFOConnect folder and double-click ICSCTL32.HLP.

Introduction

1

In This Chapter

This chapter provides an overview of the INFOConnect OLE custom control. It contains the following headings:

What is the INFOConnect OLE Custom Control?	2
How the INFOConnect OCX Works	3
Overview of the Methods, Events, and Properties	4

What is the INFOConnect OLE Custom Control?

The INFOConnect OLE custom control (OCX) is a set of methods, events, and properties that you can use to write Windows applications that use INFOConnect paths.

The INFOConnect OCX is a dynamic link library (ICSCTL32.OCX) that is explicitly linked to your application during development and run-time. The ICSCTL32.OCX interacts with the OC30.DLL to enable your application to interact with the INFOConnect Manager.

You can use the INFOConnect OCX when writing applications using Visual Basic, C, C++, or any OLE-aware application compiler that allows the use of a custom control.

Data Filters

Although you can use the INFOConnect OCX to create an application that can receive data from a host application, the data must be processed before it is displayed. For example, a host application might include escape sequences when sending data.

The INFOConnect OCX does not filter out these sequences. To filter out these sequences, your application must use a data filter that can remove escape sequences and any other unwanted information from the data.

The INFOConnect OCX comes with a sample data filter for 1100/2200 Series hosts. You can use this filter, or you can use the source code for this filter as a model for creating your own. For more information about data filters, refer to [Appendix A, “Transferring Data Between a PC and a Host.”](#)

How the INFOConnect OCX Works

The INFOConnect OCX interacts with your application during both development and run-time.

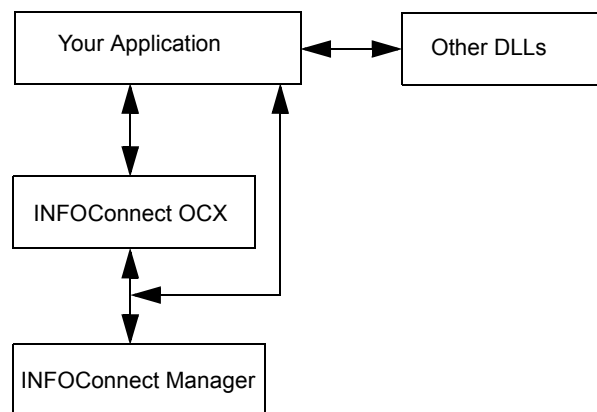
Developing Applications

During application development, the INFOConnect OCX provides access to the methods, events, and properties that you can use to control you application's interaction with the INFOConnect Manager.

For example, when developing applications using Visual Basic, you can add the INFOConnect OCX to your toolbox and then place an INFOConnect OCX object on your form. By clicking this object, you can specify a number of properties (such as AutoReceive, Name, and PathName). Once you have added the INFOConnect OCX object to your form, you can determine which events to use and how to process those events.

Running Applications

When you run your application, the INFOConnect OCX processes requests sent by the application or actions initiated by a user, as shown in the following figure:



In addition, certain properties (such as `hSession` and `hWndVBList`) are set during run-time.

Overview of the Methods, Events, and Properties

The INFOConnect OCX consists of the following methods, events, and properties. These items are briefly described in the following tables. For detailed information about these items and how to use them, refer to [Chapter 2, “Methods, Events, and Properties.”](#)

Methods

Method	Description
AboutBox	Displays the version number of the INFOConnect OCX
FireUserEvent	Triggers the UserEvent event
GetPathNames	Calculates the number of INFOConnect paths and creates a list of them
OpenAccessory	Invokes an INFOConnect accessory using a dynamically created local INFOConnect path
Receive	Sends a receive data request
SendStatus	Sends a status message value to the library or the attached application
SetPathList	Displays all INFOConnect paths in a list or combo box
Transmit	Transmits the specified data

Events

Event	Description
ReceiveDone	Notifies the application that data has been received
RptMsg	Notifies the application of a communication error
SessionClosed	Notifies the application that a path has been closed
SessionEstablished	Notifies the application that a path has been opened
Status	Occurs whenever the INFOConnect Manager sends a status message to the INFOConnect OCX
TransmitDone	Indicates a transmit command has been completed
UserEvent	Occurs in response to a FireUserEvent method call

Properties

Property	Description
Action	Lists INFOConnect paths, sends a status message, or transmits data over an INFOConnect path, depending on how it's defined
AutoReceive	Determines whether a receive data request is sent automatically or only in response to a Receive method call
hSession	Holds the session handle This property's value is set at run-time.
hWndVBList	Holds the handle to a list box or combo box that will display the names of INFOConnect paths This property's value is set at run-time.
IsEstablished	Indicates whether a SessionEstablished or SessionClosed event has occurred
Open	Opens or closes the path named in the PathName property and establishes a session with the host
PathName	Specifies the INFOConnect path to be opened or closed
StatusMessage	Specifies the integer value of the status message to send to the library
XmitData	Specifies the data to be transmitted when the Action property is set to 3

Guidelines for Developing Applications

2

In This Chapter

This chapter provides general guidelines for developing applications using the INFOConnect OCX, as well as instructions for running and reviewing the sample application included with this package.

This chapter includes the following headings:

Prerequisites	8
Converting 16-bit VBX Applications to 32-bit OCX	9
Adding the INFOConnect OCX to a Project	12
Defining Properties	13
Defining Event Procedures	14
Adding a Timer	17
Reviewing the Sample Application Source Code	19
Running the Sample Application	21

Prerequisites

Before using the INFOConnect OCX, you should have a working knowledge of the following:

- The methods, events, and properties used by the standard Windows controls
- Visual Basic, Microsoft C, Microsoft C++, or any other OLE-aware application compiler

Note: To create 32-bit applications, you must use a 32-bit development tool, such as Visual Basic 4.0.

- INFOConnect concepts, including libraries, path templates, and paths

Converting 16-bit VBX Applications to 32-bit OCX

To convert an application that was created using the 16-bit INFOConnect custom control for Visual Basic (ICS_CTRL.VBX) into a 32-bit OCX application, follow these steps:

- 1 Run Visual Basic and open the .MAK file for your existing 16-bit application.

During this process, a number of message boxes appear. Click the appropriate response as listed below:

Prompt	Response
Can't load (or register) custom control: <i>path</i> /ICSCCTRL.VBX - Continue loading project?	Yes
This file was saved in a previous version of Visual Basic. When saved, it will be saved in the Visual Basic 4.0 format.	OK for All
Control VBX.INFOConnect not found	OK
Error loading <i>form</i> .FRM. A control could not be loaded due to load error. Continue?	Yes

You will also see a dialog box that says that if your application used any data access objects, you should add the DA 2.5 object library. Click Add or Don't Add accordingly.

- 2 If your existing application used the ICGLOBAL.BAS file included with the INFOConnect custom control for Visual Basic, click Add File from the File menu and double-click the ICGLOBAL.BAS file in the OCX32 subfolder of the INFOConnect folder.

If you modified your original ICGLOBAL.BAS file, you must copy the changes from your original file into the one in this folder. However, do not copy any function declarations (that is, any lines that begin with `Declare Function`) or any lines that contain the text `ics_ctrl`.

- 3 From the Tools menu, click Custom Control.

- 4 On the Custom Controls dialog box, click **Browse**, go to the Windows SYSTEM folder, and double-click ICSCTL32.OCX.

Icsctl32 OLE Custom Control Module appears in the Available Controls list box.

- 5 Double-click Icsctl32 OLE Custom Control Module.

When you click **OK**, the INFOConnect OCX tool  is added to the toolbox.

- 6 On each form, click the red rectangle in the upper left corner of the form (this represents the VBX object) and note its name (for example, ICSSession1).

If you had more than one VBX object on the form, they are hidden underneath each other. Use the object list on each form to make sure that you have noted them all.

- 7 Delete the VBX objects from the forms.

The code they contain is saved in the (general) object of each form.

- 8 Double-click the INFOConnect OCX tool on the toolbox to add it to each form.

- 9 On the Properties window for each INFOConnect OCX object, change the default name (Icsctl32n) to the same name as the VBX object on that form.

The code that was previously in the event procedures of your VBX objects should now appear in the event procedures of the corresponding OCX objects.

- 10 Modify the argument lists of the event procedures to match the requirements of the INFOConnect OCX.

Be sure to make note of any changes you make so that you can later edit references to them in your code. For example, some data types have changed from Integer to Long.

If you changed the names of any event procedure arguments, keep the names used by your project.

- 11** Modify any references in your code to the event procedure arguments that changed.

For example, any references to the `lpRcvData` argument in the `ReceiveDone` event procedure must be changed to use the `RcvData` argument instead.

- 12** Modify any direct references in your code to VBX functions.

For example, if your project made any direct calls to the `GetPathNames` function, you must modify it to use the corresponding `GetPathNames` method.

Adding the INFOConnect OCX to a Project

To use the INFOConnect OCX, you must first add it to your project.

The procedure to use varies, depending on which tool you use to develop your application. This section includes the procedure for Visual Basic.

To add the INFOConnect OCX to a Visual Basic form, follow these steps:

- 1 Run Visual Basic.
- 2 From the Tools menu, click Custom Control.
- 3 On the Custom Controls dialog box, click Browse, go to the Windows SYSTEM folder, and double-click ICSCTL32.OCX.

Icsetl32 OLE Custom Control Module appears in the Available Controls list box.

- 4 Double-click Icsetl32 OLE Custom Control Module.

When you click OK, the INFOConnect OCX tool  is added to the toolbox.

- 5 Double-click the INFOConnect OCX tool.

This places the INFOConnect OCX tool on your form, making all the methods, events, and properties associated with the custom control available to your application.

- 6 From the File menu, click Add File.
- 7 Double-click ICGLOBAL.BAS.

This adds all the function declarations to your project so that you do not have to declare them manually.

Defining Properties

When you add the INFOConnect OCX tool to a project and click this tool, the Properties window lists the properties that you can define for the project.

You can specify the value for the design-time properties in two ways:

- Place the cursor in the cell next to the property whose value you want to specify and type the desired value.
- Double-click Custom and type the desired value for each property whose value you want to specify on the subsequent dialog box.

The Properties window lists both design-time and runtime properties. However, you can specify the value for design-time properties only. The dialog box that appears when you double-click Custom lists only design-time properties.

Defining Event Procedures

Visual Basic does not automatically add event handling routines; you must decide which events you want to use and then type the necessary code to create a response for them. For example, when your application receives a SessionClosed event, you can display a message box indicating that this has occurred.

For example, using Visual Basic, you would double-click the INFOConnect OCX object on the form, click the desired event from the Proc list box, and type the appropriate code for handling this event in the window. Once you have added code for an event, it appears in bold type in the Proc list box.

The following sections provide sample code that you might want to type for each event.

SessionEstablished Event

This code sets a session established flag and displays a message box whenever a session is successfully established.

```
Sub Icsctl321_SessionEstablished ()
    bSessionEstablished = True
    MsgBox "SessionEstablished Event"
End Sub
```

SessionClosed Event

This code sets a session closed flag and displays a message box whenever a session is closed.

```
Sub Icsctl321_SessionClosed ()
    bSessionEstablished = False
    MsgBox "SessionClosed Event"
End Sub
```

TransmitDone Event

This code displays a message on the status line whenever data has been successfully transmitted over an INFOConnect path.

```
Sub Icsctl321_TransmitDone ()
    Status.Caption = "Data has been sent."
End Sub
```

ReceiveDone Event

This code rebuilds the display area to add newly received data at the end of any previously displayed data. It also positions the pointer at the end of the display, which causes scrolling to take

place when received data goes beyond the end of the current display.

```
Sub Icsctl321_ReceiveDone (RcvData As String, _
Length As Long)
    'If only an acknowledgment was sent, the
    'following code displays a message box indicating
    'this.
    If RcvData = "" Then
        MsgBox "Acknowledgment sent"
    End If

    'Chr$(13) represent a carriage return. Chr$(10)
    'represents a line feed. RcvData is the property
    'that holds the data that was received:
    display.Text = display.Text + Chr$(13) + _
    Chr$(10) + RcvData

    'To position the cursor and scroll if needed:
    display.SelStart = Len(display.Text) + 1
End Sub
```

RptMsg Event

This code displays a message box whenever an error occurs in the INFOConnect OCX.

```
Sub Icsctl321_RptMsg (RptMsgData As String, _
RptMsgNum As Long)
    'The following If statement handles a routine
    'error that occurs when one of two local paths is
    'closed
    If RptMsgNum = IC_ERROR_TERMINATE_NOMSG Then
        Unload Sample
    Else
        'To display a message box with the number and
        'text of a report message:
        MsgBox "RptMsg Number: " + Str$(RptMsgNum) + _
        Chr$(10) + Chr$(13) + "RptMsg Data: " + _
        RptMsgData
    End If
End Sub
```

Status Event

This code displays a message box whenever the user or another application tries to close the INFOConnect Manager.

```
Private Sub Icsctl321_Status (StatusMsg As _
String, StatusNum As Long)
    UpdateEventLog "Status (" + Hex(StatusNum) + ") _
: " + StatusMsg
    'The following status indicates that another
'source is trying to close INFOConnect Manager.
    If StatusNum = &H60002 Then
        MsgBox "The INFOConnect Manager is _
        attempting to close."
    End If
End Sub
```

UserEvent Event


This code sends events to the application in an asynchronous manner.

```
Private Sub Icsctl321_UserEvent (wParam As Long, _
lParam As Long, szBuf As String)
    UpdateEventLog "UserEvent: wParam is " + _
    Str$(wParam) + CRLF + "lParam is " + _
    Str$(lParam) + CRLF + "The string is " + szBuf
    MsgBox "The wParam is " + Str$(wParam) + CRLF + _
    "The lParam is " + Str$(lParam) + CRLF + "The _
    string is " + szBuf
End Sub
```

Adding a Timer


After your application initiates an action, it usually must wait for a response before additional processing can occur. For example, when you invoke the Open property, your application must wait until a SessionEstablished event occurs; otherwise, your application might not function correctly.

To allow for any required processing to occur, you should use a timer. The following procedure shows how to add a timer to your application, create a combo box named Paths that displays a list of INFOConnect paths, and invoke the timer when the user selects a path from the combo box:

- 1 From the toolbox, double-click the Timer tool to place it on your form.
- 2 Click the Timer tool  on the form.
- 3 In the Properties window, change the Name from Timer1 to EstablishTimer.
- 4 Disable the timer by typing the following statement in the EstablishTimer subroutine:

```
Sub EstablishTimer_Timer ()  
EstablishTimer.Enabled = False  
End Sub
```

The Timer event occurs whenever the set time limit expires. Other code in the application enables the timer.

- 5 From the toolbox, double-click the Combo box tool  to place it on your form.
- 6 Click the combo box.
- 7 In the Properties window, change the Name to Paths.
- 8 Double-click the combo box to display the code window.
- 9 From the Proc list box, click Click.

This subroutine, named Sub Paths_Click (), is executed whenever the user clicks a path name in the Paths combo box.

- 10 Type the following code to open a path and establish a session:

```
Sub Paths_Click ()
    'Set the PathName property to the selected path
    If paths.ListIndex >= 0 Then
        Icsctl321.PathName = paths.List_
            (paths.ListIndex)

        'Make it possible to select another path:
        Icsctl321.Open = False
        'Close any existing session
        'Perform a loop until the SessionClosed event
        'occurs:
        Do While bSessionEstablished = True
            'Relinquish control so other PC operations
            'can occur, such as the SessionClosed event:
            DoEvents
        Loop
        'Open the path stored in PathName
        Icsctl321.Open = True
        'Set a 10-sec. timeout to avoid an infinite
        'loop:
        EstablishTimer.Interval = 10000

        'Enable the timer:
        EstablishTimer.Enabled = True

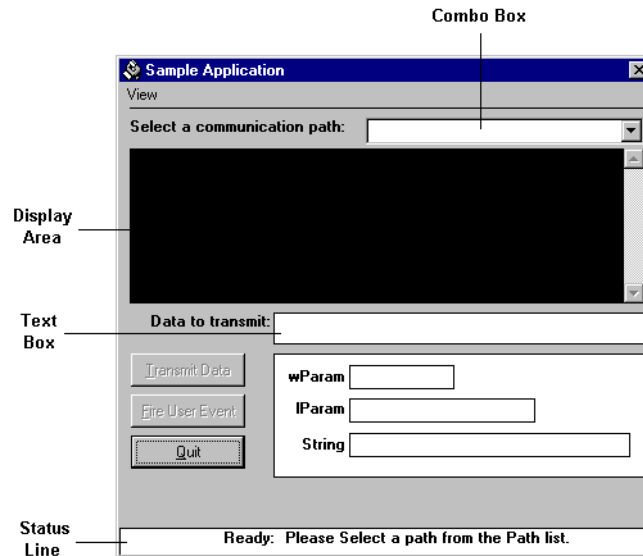
        'Wait until a session is established or the
        'time interval expires:
        Do While bSessionEstablished = False And _
            EstablishTimer.Enabled
            DoEvents
        Loop
        If bSessionEstablished = False Then
            MsgBox "Path failed to open within _
                10-second timeout."
        Else
            EstablishTimer.Enabled = False
            CmdSend.Enabled = True 'Enable the _
                command button
            Status.Caption = "Ready" 'Set the status _
                area
        End If
    End If
End Sub
```

Reviewing the Sample Application Source Code

The INFOConnect OCX comes with Visual Basic source code for the sample application SAMPLE.EXE to provide you with coding examples. (For information about running the sample application, refer to [“Running the Sample Application”](#) on page 21.)

To review the source code, run Visual Basic and load SAMPLE.VBP.

As shown in the following figure, the sample application consists of a display area, a combo box, a text box, several buttons, and other objects. Using this application, you can type a character string in the Data To Transmit text box and send that string to both the originating application’s display area and to a destination reached via an INFOConnect path.



Chapter 2 Guidelines for Developing Applications

The following table describes the components of the sample application:

Component	Description
Display Area	Displays the character strings sent to or received from the host
Combo Box	Lists the names and allows the selection of pre-defined INFOConnect paths
Text Box	Displays character strings
Transmit Data	Transmits the character string displayed in the Data To Transmit text box
Fire User Event	Triggers a UserEvent event and sends the values in wParam, lParam, and String to the event This button illustrates the coding technique for using the FireUserEvent method.
Quit	Closes the path and the application
wParam, lParam, String	Display arguments that will be passed to the UserEvent when you click the Fire User Event button
Status Line	Displays event-related messages and other prompts

Running the Sample Application

The INFOConnect OCX comes with a sample application to show you how various features of the custom control work.

The sample application is similar to a terminal emulator and uses existing INFOConnect paths to connect to a host. Therefore, you must create at least one INFOConnect path before running the sample application. For instructions on creating INFOConnect paths, refer to the online Help for the INFOConnect Database Editor (if you purchased one of Attachmate's emulators) or the INFOConnect Manager.

To run the sample application, follow these steps:

- 1** Using My Computer (or File Manager, if you're using Windows NT), go to the OCX32 subfolder of the INFOConnect folder, and then double-click SAMPLE.EXE.
- 2** From the Select A Communication Path list box, click the INFOConnect path to use to connect to a host.
- 3** Type any desired data (such as your logon name) in the Data To Transmit text box.
- 4** Click Transmit Data.

Methods, Events, and Properties

3

In This Chapter

This chapter contains an alphabetic listing of the methods, events, and properties in the INFOConnect OCX.

The syntax and examples provided here are for Visual Basic applications only.

Note: An underscore (`_`) indicates when a particular line of code is continued to the next line, as in the following example:

```
Private Sub Icsctl32_ReceiveDone (RcvData As _  
String, Length As Long)
```

AboutBox Method	25
Action Property	26
AutoReceive Property	29
FireUserEvent Method	31
GetPathNames Method	32
hSession Property	35
hWndVBLIST Property	36
IsEstablished Property	37
Open Property	38

OpenAccessory Method	40
PathName Property	43
Receive Method	44
ReceiveDone Event	46
RptMsg Event	47
SendStatus Method	48
SessionClosed Event	49
SessionEstablished Event	50
SetPathList Method	51
Status Event	52
StatusMessage Property	53
Transmit Method	54
TransmitDone Event	55
UserEvent Event	56
XmitData Property	57

AboutBox Method

The AboutBox method displays the version number of the INFOConnect OCX.

Prerequisites

None

Syntax

object.AboutBox

Comments

This method displays the About Box at run-time. To view the version number of the INFOConnect OCX at design time, click the INFOConnect OCX object on the form and double-click (About) in the Properties window.

Example

Icsctl321.AboutBox

Action Property

This property lists INFOConnect paths, sends a status message, or transmits data over an INFOConnect path, depending on how it's defined.

Prerequisites

The following table lists the prerequisites for each Action setting:

Setting	Prerequisites
1	SessionEstablished event hWndVBList property
2	SessionEstablished event StatusMessage property
3	SessionEstablished event XmitData property

Syntax

object.Action = setting

where *setting* can be one of the following:

Setting	Action
1	List INFOConnect paths
2	Send status message
3	Transmit data

Data Type

Integer

Comments

The Action property is included in the INFOConnect OCX for backward compatibility with existing applications developed using the INFOConnect Custom Control for Visual Basic. However, for new applications, you should use the following methods instead:

Action Property Setting	Equivalent Method
1	SetPathList
2	SendStatus
3	Transmit

When the Action property is set to 1, a valid handle to the combo box must be assigned to the hWndVBLIST property before the Action is invoked. Setting the Action property to 1 displays all INFOConnect paths in a list or combo box. To display only inactive paths, you must use the GetPathNames method, described later in this chapter.

When the Action property is set to 2, the decimal number of the status message must be assigned to the StatusMessage property. Refer to “[StatusMessage Property](#)” on page 53, as well as [Appendix C, “Status Messages”](#) for a list of status messages and their numbers. Each status that you can send to a library has been declared in the ICGLOBAL.BAS file; consequently, you can use either the decimal number or the name of the status message. For example, for the Message Wait status message, you can use either 458752 or IC_FKEY_MSGWAIT.

When the Action property is set to 3, the data to be transmitted must be assigned as a string to the XmitData property. Only data up to the first null character will be transmitted. To transmit messages that contain null characters, you must use the Transmit method.

**Example:
Action = 1**

The code in this example finds INFOConnect paths and places their names in a combo box called Paths.

```
Sub Form_Load
    'Assigns the handle of the combo box to the
    'session handle:
    Icsctl321.hWndVBLIST = Paths.hWnd
    'Places the names of paths in the combo box
    'specified by hWndVBLIST:
    Icsctl321.Action = 1
End Sub
```

Example:
Action = 2

The code in this example sends a PF2 keystroke to the server application. Note that the host application must be written to accept the keystroke. (The meaning of the keystroke is application dependent.)

```
Sub CmdSend_Click ()
    'Type the status message number for the PF2
    'keystroke:
    Icsctl321.StatusMessage = 458754
    'Send the status message contained in
    'StatusMessage:
    Icsctl321.Action = 2
End Sub
```

Example:
Action = 3

The code in this example transmits the text in the XmitData property when a user clicks a Transmit button.

```
Sub CmdSend_Click ()
    Icsctl321.XmitData = "Hello" 'Set XmitData
    Icsctl321.Action = 3 'Transmit the string in
    'XmitData
End Sub
```

See Also

hWndVBList Property, SendStatus Method, SessionEstablished Event, SetPathList Method, StatusMessage Property, Transmit Method, TransmitDone Event, XmitData Property

AutoReceive Property

The AutoReceive property determines whether receive data requests are sent automatically by the INFOConnect OCX or only in response to a Receive method call.

Prerequisites

None

Syntax

`object.AutoReceive = True | False`

Comments

When this property is set to True (which is the default state), receive data requests are sent automatically. (The Open property triggers the first receive request, and the ReceiveDone event triggers the next receive request.) Each received message is queued to your application as it arrives. If your application will receive messages at a controlled rate (such as a client application that receives one or two reply messages to each message transmitted), you can set AutoReceive to True.

When this property is set to False, you must issue a Receive method call to request receive data. This lets your application control the rate at which messages are received. If your application might receive messages faster than it can process them (such as server application that receives requests from many clients), set AutoReceive to False and use the Receive method to indicate when the application is ready to receive more data. (If you change AutoReceive to True after calling the Receive method, you must call the Receive method once more to trigger the next receive request. Then, the ReceiveDone event triggers the next request automatically.)

Typically, you would call the Receive method following SessionEstablished, ReceiveDone, and RptMsg event handling, since transmit or receive errors cause the current Receive to be canceled.

Example

The code in the following example sets `AutoReceive` to `False`, opens a session, waits for the `IsEstablished` event (indicated by `IsEstablished` property changing to `True`), and sends a receive data request via the `Receive` method.

```
Icsctl321.PathName = "TCPA_1"  
Icsctl321.AutoReceive = False  
Icsctl321.Open = True  
While Icsctl321.Open = False  
    DoEvents  
Wend  
Icsctl321.Receive 'Request next data block
```

See Also

Receive Method, ReceiveDone Event

FireUserEvent Method

The FireUserEvent method call causes the UserEvent to occur and passes three types of data to that event.

Prerequisites Values defined for wParam, lParam, and szBuf

Syntax `object.FireUserEvent ByVal wParam As Long, ByVal _
lParam As Long, ByVal szBuf As String`

Comments The meaning of the variables that are passed depends on their use in your Visual Basic application. Three variables are passed to give you access to three major variable types.

Example The code in this example passes three values (wParam, lParam, and szBuf) to the UserEvent event whenever a user clicks a button called Fire Event. In this example, three values are entered into three text boxes named Text2, Text3, and Text4.

```
Sub FireEvent_Click ()  
    Dim wParam As Long  
    Dim lParam As Long  
    Dim szBuf As String  
  
    wParam = Val(Text2.Text)  
    lParam = Val(Text3.Text)  
    szBuf = Text4.Text  
  
    Icsctl321.FireUserEvent wParam, lParam, szBuf  
End Sub
```

The following code illustrates the UserEvent triggered by the method. The code causes a message box to display the values of the three passed arguments.

```
Private Sub Icsctl321_UserEvent (wParam As Long, _  
lParam As Long, szBuf As String)  
  
    MsgBox "The wParam is " + Str$(wParam) + Chr$(13) + _  
    Chr$(10) + "The lParam is" + Str$(lParam) + _  
    Chr$(13) + Chr$(10) + "The string is " + szBuf  
  
End Sub
```

See Also UserEvent Event

GetPathNames Method

The `GetPathNames` method call determines the total number of INFOConnect paths in the INFOConnect database and returns the names of all paths or all inactive paths in a string variable.

Prerequisites AllFlag must be set to either True or False.

Syntax `ReturnValue = object.GetPathNames (TotalPaths, _
PathNames, AllFlag)`

Return Values There are two types of return values:

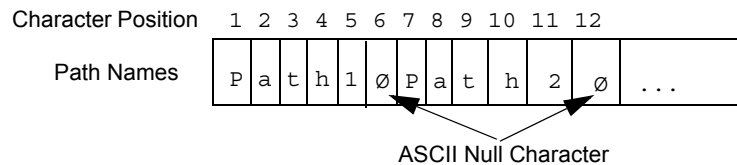
- The data in `TotalPaths` and `PathNames`
- The method's return value

The method's return values are shown in the following table:

Return Value	Description
1	The method completed successfully.
0	No paths were found.
-4, -1	An out-of-memory error occurred.
-2	The INFOConnect Manager was unavailable.
-5, -3	A memory failure occurred.

Comments

TotalPaths is an integer variable that indicates the total number of INFOConnect paths. PathNames is a string variable that is filled with the names of INFOConnect paths. In the string variable, the path names are separated from each other by the ASCII null character (as shown in the following figure).



When AllFlag is set to True, all of the INFOConnect path names are returned in the path names string. When AllFlag is set to False, only the inactive paths (paths that are not in use by another session) are included. Set AllFlag to False when you want to avoid including active paths.

Example

The following example retrieves all the path names, adds them to a Paths combo box, and places a message on the status line:

```
Private Sub Form_Load()

    Dim TotalPaths As Long
    Dim PathNames As String
    Dim bAllPaths As Integer
    Dim ReturnValue As Integer
    Dim AsciiNull As String * 1
    Dim StartPosition As Integer
    Dim ReadPaths As Integer
    Dim NullPosition As Integer
    Dim PathName As String

    CRLF = Chr$(13) & Chr$(10)
    bSessionEstablished = False

    PathNames = "This is a test" 'Testing only

    bAllPaths = False 'Active paths only
    ReturnValue = Icsctl321.GetPathNames (TotalPaths, _
    PathNames, bAllPaths)
    If (ReturnValue < 1) Then
        MsgBox "Cannot access paths"
    ElseIf TotalPaths = 0 Then
        MsgBox "No communication path available"
```

```
Else
  AsciiNull = Chr$(0) 'Set to ASCII null character
  StartPosition = 1 '1st character position of
  'first path name in string
  For ReadPaths = 1 To TotalPaths 'Start reading
  'paths into list box
    NullPosition = InStr(StartPosition, PathNames, _
    AsciiNull) 'Find null-character position
    PathName = Mid$(PathNames, StartPosition, _
    NullPosition - StartPosition) 'Find a path name
    paths.AddItem PathName 'Add found path name
    'to list box
    StartPosition = NullPosition + 1 'Set to one
    'character position beyond last-found null
    'character to start looking for next path name
  Next ReadPaths
End If

Status.Caption = "Ready: Please Select a path from _
the Path list."

End Sub
```

See Also [PathName Property](#)

hSession Property

The hSession property holds the INFOConnect session handle.

Prerequisites SessionEstablished event

Syntax `object.hSession`

Data Type Integer

Comments This property is available as read-only at run-time. It is not available at design time.

In addition, this property is required only if your application makes direct calls to the INFOConnect Accessory API (described in the INFOConnect Developer's Kit).

Example The following example is taken from the FireUserEvent method call used with the Fire Event button in the sample application. The method uses the hSession property to identify the handle for Icsctl321. (For the complete example, refer to “[FireUserEvent Method](#)” on page 31.)

```
Sub GetSessionID ()
Dim SessID As Long
'This returns the handle of the current session.
SessID = Icsctl321.hSession
End Sub
```

See Also SessionEstablished Event

hWndVBList Property

Use the hWndVBList property to hold the handle of a list box or combo box.

Prerequisites A list box or combo box must be on the form.

Syntax `object(hWndVBList = handle`
where *handle* is the handle of the list box or combo box on the form.

Data Type Integer

Comments The hWndVBList, which is available only at run-time, is used by the Action property to identify the list box or combo box that will display INFOConnect path names.

Example The code in this example sets the hWndVBList property to the handle of a combo box named Paths. The SetPathList method or Action property fills the combo box identified by the hWndVBList with INFOConnect path names.

```
Sub Form_Load ()  
    Icsctl321(hWndVBList = Paths(hWnd  
    Icsctl321.Action = 1  
End Sub
```

See Also Action Property, SetPathList Method

IsEstablished Property

The IsEstablished property indicates whether a SessionEstablished or SessionClosed event has occurred.

Prerequisites

None

Syntax

object.IsEstablished

Comments

This property is read-only and is set at run-time by the INFOConnect OCX.

The property is set to True when a SessionEstablished event occurs. It is set to False when a SessionClosed event occurs.

Many of the INFOConnect OCX's methods and properties (such as the Transmit method) require that a session be established before they can be used. You can use the IsEstablished property to test the state of the session. This provides an alternative to using a user-defined state flag (such as bSessionEstablished in the examples elsewhere in this guide).

Example

This example uses the IsEstablished property to make sure that the session is open before calling the Transmit method. Otherwise, an error message appears.

```
Sub CmdTransmit_Click ()
    Dim TxData As String
    Dim TxLen As Integer
    TxData = "Hello"
    TxLen = Len(TxData)

    If (Icsctl321.IsEstablished = True) Then
        Icsctl321.Transmit TxData, DataLength
    Else
        MsgBox ("Session not yet established")
    EndIf
End Sub
```

See Also

SessionClosed Event, SessionEstablished Event

Open Property

The Open property is a Boolean-type property that opens or closes a path.

Prerequisites PathName property

Syntax `object.Open = True | False`

Data Type Integer (Boolean)

Comments If you set Open to True, the path to which the PathName property is set will be opened. If the PathName property is empty, the INFOConnect Manager displays a list of paths for the user to choose from.

Setting the Open property to True initiates the opening of a session, but you cannot use the session until the SessionEstablished event occurs, at which time the IsEstablished property will be set.

If you set the Open property to False, the path will be closed.

Note: To avoid errors, the application should wait for the SessionClosed event to occur before it unloads the Visual Basic form.

Example

The code in the following example shows how to use the `Open` property to open the path specified by the `PathName` property:

```
Sub ... ()  
    .  
    .  
    .  
    Icsctl321.PathName = Localc  
  
    'Initiates the opening of the path specified by the  
    'PathName property:  
    Icsctl321.Open = True  
  
    'Normally, you would include a Timer routine to wait  
    'until the SessionEstablished event occurs or until  
    'IsEstablished is set to True before continuing with  
    'the application  
    .  
    .  
    .  
End Sub
```

See Also

`IsEstablished` Property, `OpenAccessory` Method, `PathName` Property, `SessionClosed` Event, `SessionEstablished` Event

OpenAccessory Method

The OpenAccessory method invokes an INFOConnect accessory using a dynamically created local INFOConnect path and returns a value that indicates whether this operation was successful.

Prerequisites The accessory being invoked must be installed.

Syntax

```
object.OpenAccessory (lpszAccessoryName As String, _  
lpszCmdLineOptions As String, lpszSessionName As _  
String) As Boolean
```

where the first parameter is an Accessory ID or executable file name, the second parameter is a string of command line options, and the third parameter is the name of the dynamically created local INFOConnect path.

Comments The following table lists Accessory IDs for Attachmate's emulators. If you use the OCX with different products, refer to the Accessory ID specified in the INFOConnect Manager for the appropriate value.

Product	Accessory ID
T 27	MT
UTS	UTS60
HiBrow	HIBROW
ALC	ALC
WinFTP	WINFTP

The following table lists valid values for the second parameter. These values are not case-sensitive

Command Line Options

To do this	Use this for the second parameter
Open a specific session	<p><i>session_name</i></p> <p>where <i>session_name</i> is the name of the session to open, including the drive, directory, and file extension (.ADP)</p> <p>If you include this among the command line options, it must be the first item.</p>
Run the application in debug mode (if the application supports this)	-D
Note: Accessory Manager does not support this option.	
Open a default session profile	<p>-K <i>Accessory</i></p> <p>where <i>Accessory</i> is one of the Accessory IDs specified in the table on the preceding page</p> <p>If you use an Accessory ID for the first parameter, or if you specify a session name for the second parameter, you do not have to include this option. However, if you use the executable file name (such as ACCMGR32.EXE) as the first parameter and omit a session name, you can use this command line option to specify the type of session to open.</p>
Position the application window	<p>-L (x,y,xx,yy)</p> <p>where x and y are the coordinates of the top left corner of the window, and xx and yy are the coordinates of the bottom right corner of the window</p> <p>For a 640x480 resolution monitor, the xx range is 0–639, and the yy range is 0–479.</p>

Command Line Options, continued

To do this	Use this for the second parameter										
Specify the application window state	-Wxy where x is one of the following:										
	<table border="1"> <thead> <tr> <th style="text-align: left;">Value</th> <th style="text-align: left;">Meaning</th> </tr> </thead> <tbody> <tr> <td>n</td> <td>normal</td> </tr> <tr> <td>m</td> <td>maximized</td> </tr> <tr> <td>i</td> <td>minimized</td> </tr> <tr> <td>h</td> <td>hidden</td> </tr> </tbody> </table>	Value	Meaning	n	normal	m	maximized	i	minimized	h	hidden
Value	Meaning										
n	normal										
m	maximized										
i	minimized										
h	hidden										
	and y is one of the following:										
	<table border="1"> <thead> <tr> <th style="text-align: left;">Value</th> <th style="text-align: left;">Meaning</th> </tr> </thead> <tbody> <tr> <td>a</td> <td>active</td> </tr> <tr> <td>b</td> <td>background</td> </tr> </tbody> </table>	Value	Meaning	a	active	b	background				
Value	Meaning										
a	active										
b	background										

The third parameter can be any 15-character name. This path uses the Local external interface library (EIL). You can use this path to send and receive data between your OCX application and the invoked application using the standard OCX methods and events. For more information about the Local library, refer to the online Help for the INFOConnect Manager.

Before you call `OpenAccessory` a second time, you must set the `Open` property to `false`.

Example

The following example shows how to use the `OpenAccessory` method to open the default T 27 session profile in a maximized active state using the local path `LOCAL_1`:

```
Dim bResult As Boolean
bResult =
Icsctl321.OpenAccessory("C:\INFOCN32\ACCMGR32\ACCMGR32.E
XE", "-K MT -Wma", "LOCAL_1")
if bResult = True then
    MsgBox "Icsctl321.OpenAccessory = True"
else
    MsgBox "Icsctl321.OpenAccessory = False"
end if
```

See Also

Open Property

PathName Property

The PathName property assigns a specific INFOConnect path to the Icsctl32n control.

Prerequisites

A path must exist in the INFOConnect database.

Syntax

```
object.PathName = pathname
```

where *pathname* is the name of the path to use.

Data Type

String

Comments

This property is available at design and run-time. No action is taken when PathName is set. The Open property opens and closes the path whose name is stored in the PathName property.

Example

The code in the following example assigns to the PathName property the name of an INFOConnect path that the user selects from the Paths combo box. The code then opens the path stored in the PathName property by using the Open property.

A more complete example, which includes timers for session establishment, is included in the Paths_Click subroutine in the sample application.

```
Sub ... ()
    .
    .
    .
    'Sets the PathName property to the name of the path
    'that is opened using the Open property:
    Icsctl321.PathName = Localc

    Icsctl321.Open = True
    .
    .
    .
End Sub
```

See Also

Open Property

Receive Method

The Receive method sends a receive data request which informs the INFOConnect OCX that your application is ready to receive data.

Prerequisites AutoReceive must be set to False.

Syntax `object.Receive`

Comments When the AutoReceive property is set to True (which is the default state), receive data requests are sent automatically. (The Open property triggers the first receive request, and the ReceiveDone event triggers the next receive request.) Each received message is queued to your application as it arrives. If you application will receive messages at a controlled rate (such as a client application that receives one or two reply messages to each message transmitted), you can set AutoReceive to True.

When the AutoReceive is set to False, you must issue a Receive method call to request receive data. This lets your application control the rate at which messages are received. If your application might receive messages faster than it can process them (such as server application that receives requests from many clients), set AutoReceive to False and use the Receive method to indicate when the application is ready to receive more data. (If you change AutoReceive to True after calling the Receive method, you must call the Receive method once more to trigger the next receive request. Then, the ReceiveDone event triggers the next request automatically.)

Typically, you would call the Receive method following SessionEstablished, ReceiveDone, and RptMsg event handling, since transmit or receive errors cause the current Receive to be canceled.

Each time you issue a Receive method call, wait until the ReceiveDone event occurs before issuing another Receive method call.

Example

This example calls the Receive method in response to a SessionEstablished event.

```
Sub Icsctl321_SessionEstablished ()  
    Icsctl321.Receive  
End Sub
```

See Also

AutoReceive Property, ReceiveDone Event, SessionEstablished Event

ReceiveDone Event

The ReceiveDone event occurs whenever the INFOConnect OCX receives data from the host or from the INFOConnect Manager.

Prerequisites SessionEstablished event

Syntax `Icsctl32n_ReceiveDone (RcvData As String, Length _
As Long)`

where RcvData is a string argument that contains the received data and Length is an integer that equals the length of RcvData.

Comments If the AutoReceive property is set to False, your application will not receive any data until it calls the Receive method to indicate that it is ready to receive data.

Example The code in the following example displays a message box that contains the string of data that was sent to the application.

```
Sub Icsctl321_ReceiveDone (RcvData As String, Length _  
As Long)  
    MsgBox "Data received: " + RcvData  
End Sub
```

See Also AutoReceive Property, Receive Method, SessionEstablished Event

RptMsg Event

The RptMsg event occurs when the INFOConnect OCX passes an error to your application. One of the event's arguments, MsgData, contains a description of the error. Refer to [Appendix B, "RptMsg Messages,"](#) for a list of possible error messages.

Prerequisites

None

Syntax

```
Icsctl32n_RptMsg (RptMsgData As String, RptMsgNum _  
As Long)
```

Comments

RptMsgData holds the error message returned by the INFOConnect OCX. RptMsgNum holds the number assigned to the error.

Example

The code in the following example displays a message box whenever an error occurs. The message box displays the error number and text.

```
Sub Icsctl321_RptMsg (RptMsgData As String, RptMsgNum _  
As Long)  
    MsgBox "Error Code:  " + Str$(RptMsgNum) + Chr$(10) _  
        + Chr$(13) + "Error Message:  " + RptMsgData  
End Sub
```

SendStatus Method

The SendStatus method sends a status message to the library.

Prerequisites SessionEstablished event

Syntax `object.SendStatus StatusMsg`

where *StatusMsg* is the message to send.

Comments Refer to “[Status Messages Sent from an Accessory to a Library](#)” on page 80 for a list of status messages and their numbers. Each status that you can send to a library has been declared in the ICGLOBAL.BAS file; consequently, you can use either the decimal number or the name of the status message. For example, to close a connection, you can use either 131073 or IC_CONNECT_CLOSE.

Use this method instead of setting the Action property to 2.

Example This example sends the specified message to the library.

```
Dim StatusMsg As Long  
  
    StatusMsg = 131073  
  
Icsctl321.SendStatus StatusMsg
```

See Also Action Property, SessionEstablished Event

SessionClosed Event

The SessionClosed event occurs after a previously opened INFOConnect path is closed (the Open property is set to False). This event marks the end of a transitional state and signifies that the session is closed.

Prerequisites

None

Syntax

```
Icsctl32n_SessionClosed ()
```

Comments

A session is in a transitional state from the time your application closes the session to the time this event occurs. No OCX operations should be performed during this period.

When the SessionClosed event occurs, the IsEstablished property is set to False. This property can be used by other routines in the application to verify whether a session is closed. Alternatively (as in the following example), the application can include a global variable that is set to False when a SessionClosed event occurs.

Example

The SessionClosed event illustrated in the following example occurs whenever the Open property is set to False. When this event occurs, an integer variable, bSessionEstablished, is set to False, and a message box appears.

```
Sub Icsctl321_SessionClosed ()  
    bSessionEstablished = False  
    MsgBox "The session is closed."  
End Sub
```

See Also

IsEstablished Property, SessionEstablished Event

SessionEstablished Event

The SessionEstablished event occurs whenever an INFOConnect path is successfully opened (accomplished by the setting of the Open property to True). This event marks the end of a transitional state and signifies that the session is open.

Prerequisites

None

Syntax

```
Icsctl32n_SessionEstablished ()
```

Comments

A session is in a transitional state from the time your application opens the session to the time this event occurs. No OCX operations should be performed during this period.

When the SessionEstablished event occurs, the IsEstablished property is set to True. This property can be used by other routines in the application to verify whether a session is open. Alternatively (as in the following example), the application can include a global variable that is set to True when a SessionEstablished event occurs.

Example

In the following example, an integer variable, bSessionEstablished, is set to True and a message box appears whenever an INFOConnect path is successfully opened (that is, whenever a session is established).

```
Sub Icsctl321_SessionEstablished ()
    bSessionEstablished = True
    MsgBox "A session has been established."
End Sub
```

See Also

IsEstablished Property, Open Property, SessionClosed Event

SetPathList Method

The SetPathList method displays all INFOConnect paths in a list or combo box.

Prerequisites A list box or combo box must be on the form, and the SessionEstablished event must have occurred.

Syntax `object.SetPathList hWnd`
where *hWnd* is the handle of a combo box or list box on the form.

Comments This property displays *all* INFOConnect paths in a list or combo box. To display only inactive paths, you must use the GetPathNames method, described earlier in this chapter.

Use this method instead of setting the Action property to 1.

Example The following example fills the Paths list box with a list of all INFOConnect paths.

```
Sub CmdSetPathList_Click ()  
    Icsctl321.SetPathList (Paths.hWnd)  
End Sub
```

See Also Action Property, GetPathNames Method, SessionEstablished Event

Status Event

The Status event occurs when the INFOConnect Manager sends a status message to the INFOConnect OCX. You can trap these messages and process them internally or display them to the user.

Prerequisites

None

Syntax

```
Icsctl32n_Status (StatusData As String, StatusNum _  
As Long)
```

where StatusData contains a string associated with the INFOConnect status message and StatusNum contains the specific INFOConnect status message.

Comments

Refer to [Appendix C, "Status Messages,"](#) for a list of status messages and their meanings.

Example

The following example displays a message box if the INFOConnect Manager is being closed by another source (such as the user or another application).

```
Private Sub Icsctl321_Status (StatusData As String, _  
StatusNum As Long)  
  
    'The following status indicates that another source is  
    'trying to close INFOConnect Manager.  
    If StatusNum = &H60002 Then  
        MsgBox "The INFOConnect Manager is trying to _  
        close"  
    End If  
  
End Sub
```

StatusMessage Property

The StatusMessage property specifies the integer value of the status message to send to the library.

Prerequisites

SessionEstablished event

Syntax

`object.StatusMessage = status_number`

where *status_number* is the integer value of one of the status messages specified in [Appendix C, “Status Messages.”](#)

Data Type

Long

Comments

Each status message is specified by an integer value. The status messages are declared as variables in the ICGLOBAL.BAS file. See [Appendix C, “Status Messages”](#) for a list of these status messages and their integer values.

Example

This example sends a PF2 keystroke to the server application. The host application must be written to accept the keystroke. (The meaning of the keystroke is application dependent.) ` (See page 47)

```
Sub CmdSend_Click ()
    'Send the status message number for the PF2
    'keystroke:
    Icstl321.StatusMessage = 458754

    'Send the status message contained in StatusMessage:
    Icstl321.SendStatus Icstl321.StatusMessage
End Sub
```

See Also

SendStatus Method, SessionEstablished Event

Transmit Method

The Transmit method transmits the specified data.

Prerequisites SessionEstablished event

Syntax `object.Transmit TransmitData, Length`

where *TransmitData* is the data to transmit and *Length* is the length of the data string to be transmitted.

Comments Use this method instead of setting the Action property to 3.

Example The following sends a user ID and password to log on to a system.

```
Dim UserID As String
UserID = "MyUserID"
If (Icsctl321.IsEstablished = True) Then
    Icsctl321.Transmit UserID, Len(UserID)
Else
    MsgBox ("Session not yet established")
EndIf
```

See Also Action Property, SessionEstablished Event, TransmitDone Event

TransmitDone Event

The TransmitDone event occurs whenever a the Transmit method or transmit Action property (Action=3) is successfully completed.

Prerequisites

None

Syntax

```
Icsctl32n_TransmitDone ()
```

Comments

You can use the TransmitDone event to display a message box informing the user that the data they have just sent has been successfully transmitted.

Example

The following example updates a text box with information that is also sent to the host application.

```
Private Sub CmdSend_Click()  
    'If transmit data box empty, exit  
    If text1.Text = "" Then Exit Sub  
    Icsctl321.XmitData = text1.Text  
    text1.Text = ""  
    display.Text = display.Text + CRLF + "=>" + _  
    Icsctl321.XmitData  
    'Reposition the cursor in the text box and scroll  
    'if necessary  
    display.SelStart = Len(display.Text) + 1  
    Icsctl321.Transmit  
End Sub  
  
Sub Icsctl321_TransmitDone ()  
    Status.Caption = "Transmit done."  
End Sub
```

See Also

Action Property, Transmit Method, XmitData Property

UserEvent Event

The UserEvent event occurs whenever the FireUserEvent method is invoked.

Prerequisites None

Syntax `Icsctl32n_UserEvent (wParam As Integer, lParam As Long, _
szBuf As String)`

Comments The way wParam, lParam, and szBuf are used in the application is completely user-defined. The values assigned to the variables are received from the FireUserEvent method.

Example The code in the following example creates a message box and displays the value of each of the variables passed to the UserEvent event from the FireUserEvent method.

```
Sub Icsctl321_UserEvent (wParam As Integer, lParam _  
As Long, szBuf As String)  
    MsgBox "The wParam is " + Str$(wParam) + Chr$(13) _  
        + Chr$(10) + "The lParam is " + Str$(lParam) + _  
        Chr$(13) + Chr$(10) + "The string is " + szBuf  
End Sub
```

See Also FireUserEvent Method

XmitData Property

The XmitData property is used with the Transmit method and the Action property. Place the data that you want to send in the XmitData property before you call the Transmit method or set the Action property to 3.

Prerequisites

SessionEstablished event

Syntax

```
object.XmitData = string
```

where *string* is the string to transmit.

Data Type

String

Comments

This property is available during design and run-time.

Example

The code in the following example takes whatever the user types into the Text1 text box, assigns it to the XmitData property, and then sends it over the open INFOConnect path when the user clicks the Transmit button.

```
Sub CmdSend_Click ()

'XmitData property is set to the string in the Text1
'text box:
  Icsctl321.XmitData = Text1.Text

'The newly typed text is transmitted over the open
'INFOConnect path:
  Icsctl321.Transmit

End Sub
```

See Also

Action Property, SessionEstablished Event, Transmit Method, TransmitDone Event

Transferring Data Between a PC and a Host

A

In This Appendix

This appendix gives general information about transferring data between an application on a PC and an application on a host.

This appendix includes the following headings:

What Is a Data Filter?	60
Using the Sample Data Filter	61

What Is a Data Filter?

Although you can use the INFOConnect OCX to create an application that can receive data from a host application, the data must be processed before it is displayed. For example, a host application might include escape sequences when sending data.

The INFOConnect OCX does not filter out these sequences. To filter out these sequences so that they do not appear on the screen, your application must use a data filter that can remove escape sequences and any other unwanted information from the data.

The INFOConnect OCX comes with a data filter for 1100/2200 Series hosts. You can use this filter, or you can use the source code for this filter as a model for creating your own.

The following data filter files are included with the INFOConnect OCX:

- MAINFORM.FRM
- PRTHNDLR.BAS
- PRTHNDLR.EXE
- PRTHNDLR.VBP
- SESSIONF.FRM

Using the Sample Data Filter

The INFOConnect OCX includes a sample data filter for 1100/2200 Series hosts.

To use the sample data filter in your application, follow these steps:

- 1 Run Visual Basic.
- 2 From the File menu, click Add File.
- 3 Double-click PRTHNDLR.BAS.
- 4 Double-click the INFOConnect OCX object on the form.
If this object is not on the form, add it as described in [“Adding the INFOConnect OCX to a Project”](#) on page 12.
- 5 From the Proc list box, click ReceiveDone.
- 6 Type the following code in the ReceiveDone event window:

```
Sub Icsctl321_ReceiveDone (RcvData As String, Length _  
As Long)  
  
AddReceiveBufferToTerminalScreenList RcvData, _  
Len(RcvData)  
  
End Sub
```
- 7 In your application, make the data's display area a list box and name it TerminalScreenList.

RptMsg Messages

B

In This Appendix

This appendix lists the error messages handled by the RptMsg event and includes the following headings:

INFOConnect OCX Errors	62
INFOConnect Termination Errors	64
INFOConnect Severe Errors	67

The information in “INFOConnect Termination Errors” and “INFOConnect Severe Errors” is taken from the *INFOConnect Development Kit Basic Programming Reference Manual*, Appendix C, “Errors and Results,” published by Unisys® Corporation.

INFOConnect OCX Errors

For these errors, the text **in this type** is what appears in the message box, and the subsequent text provides additional information and suggestions for recovering from the error.

When you create your application, you can either trap these error messages and create an error handling routine to respond to them, or you can display them to the user.

83886081

Error adding to PathListBox.

The control was unable to add a path name to the list or combo box specified by the list box handle, hWndVBList. The box could be full, the handle could be incorrect, or the system could be low on resources. Check the handle, close any unneeded applications, and try again.

83886082

Failed to LockBuffer hMemPathList.

The control was unable to allocate memory for an internal buffer. Close any unneeded applications and try again.

83886083

hMemPathList is NULL: Control was unable to allocate a buffer of adequate size.

The control was unable to allocate memory for an internal buffer. Close any unneeded applications and try again.

83886084

hWndList is NULL; Invalid handle passed from the user's program.

The hWndVBList property holds an undefined handle for a list or combo box. As a result, the control can't fill the list box with INFOConnect path names. Make sure the handle assigned to hWndVBList is for a combo or list box and try again.

83886085

IcGetPathNames returned an error!

The INFOConnect Manager reported an error to the custom control. No paths might exist, the INFOConnect database might be corrupted, or there could be a problem allocating system resources because of low memory or resource space. Make sure the database includes at least one path, close any unneeded applications, and try again.

83886086

Number of paths is 0.

Either all the paths are being used, there are no paths in the INFOConnect database, the INFOConnect Manager is having trouble accessing the database, or the database has become corrupt. Verify that there are paths defined and available.

INFOConnect Termination Errors

Termination errors indicate that the particular request failed and that all other requests on the associated session will also fail. If the default error procedure is called, the error message appears and the session closes automatically. If the default error procedure is not called, you must close the session manually.

For these errors, the text in `this` type is the message that appears on the screen (if one appears), and the subsequent text provides additional information and suggestions for recovering from the error.

12582913

IC_ERROR_BADSESSION

Invalid session handle detected at *string*. Session must be terminated.

An underlying layer of Connectivity Services received a handle to a session that is not a valid session handle or, if required, the handle of an established session. Once the error message appears, the session closes through the close session procedure. If the default error procedure is called, it closes the session automatically.

For debugging purposes, an INFOConnect library returning this error result must first call `IcSetSessionError` with the `lpInsert1` parameter pointing to a string that identifies the location in the code where the error was detected (such as `IcLibXmit`).

Trace the session, recreate the error, and send the resulting TRACE.LOG file to Customer Support.

12583014

IC_ERROR_NODATABASE

Database Not Found. Terminate and restart INFOConnect.

A valid INFOConnect database could not be located. This error occurs when the database was not properly opened or created. During initialization, the INFOConnect shell should have received the specific database error and displayed the error to the user.

Exit the INFOConnect Manager, verify that all INFOConnect command line parameters are correct, and restart the Manager. If this error persists, reinstall Connectivity Services or contact Customer Support.

12583015

IC_ERROR_SHELL_ACTIVE

An INFOConnect Shell is already active. You cannot run multiple shells.

An accessory tried to register itself as the INFOConnect shell through IcInitShell and an INFOConnect shell is already running. Only one INFOConnect shell can be active at a time. Close the application that is causing the error and contact the application's vendor.

12583016

IC_ERROR_TERMINATE_CLEAR

A request has been made to clear this session.

The user clicked a path from the INFOConnect Manager's Active Session window and then clicked Clear. The application can intercept this error and perform its termination routine before allowing the session to close. If the default error procedure is called, the session closes automatically.

Unless the INFOConnect Manager is running in debug mode or `-d` appears on the command line for the Manager, the error text does not appear on the screen. However, if it does, click OK on the default error dialog box to close the session.

12583017

IC_ERROR_TERMINATE_EXIT

A request has been made to close this session so INFOConnect can exit.

A session is being closed because the user is closing the INFOConnect Manager. The application can intercept this error and perform its termination routine before allowing the session to close. If the default error procedure is called, the session closes automatically.

Unless the INFOConnect Manager is running in debug mode or `-d` appears on the command line for the Manager, the error text does not appear on the screen. However, if it does, click OK on the default error dialog box to close the session.

12582912

IC_ERROR_TERMINATE_NOMSG

A request has been made to unconditionally terminate this session.

This error is generated by the Local external interface library if one half of a connected session using a local path is closed. It can also be generated by the IcTELNET service library if a TCP socket of is closed. If the default error procedure is called, the session closes automatically.

Unless the INFOConnect Manager is running in debug mode or -d appears on the command line for the Manager, the error text does not appear on the screen. However, if it does, click OK on the default error dialog box to close the session.

12583018

IC_ERROR_TERMINATE_SHUTDOWN

A request has been made to close this session so workstation can shutdown.

A session is being closed because the user is closing Windows. The application can intercept this error and perform its termination routine before allowing the session to close. If the default error procedure is called, the session closes automatically.

Unless the INFOConnect Manager is running in debug mode or -d appears on the command line for the Manager, the error text does not appear on the screen. However, if it does, click OK on the default error dialog box to close the session.

INFOConnect Severe Errors

Severe errors indicate that a particular request failed. These errors always appear on the screen.

For these errors, the text in `this` type is the message that appears on the screen, and the subsequent text provides additional information and suggestions for recovering from the error.

8389409

IC_ERROR_ACCESSORY_FAILED

Accessory `accessory_name` execution failed. Verify that this is a valid Windows code file.

The specified accessory could not be run. (For more information, refer to the `IcOpenAccessory` and `IcRunAccessory` functions in the *INFOConnect Development Kit Basic Programming Reference Manual*.) Verify that the specified file is a valid Windows code file.

8389408

IC_ERROR_ACCESSORY_NOT_FOUND

Accessory `accessory_name` not found. Verify the accessory installation, the file name, and the DOS path.

An attempt was made to run an INFOConnect accessory through `IcOpenAccessory` or `IcRunAccessory` and the file or the DOS path to that file could not be found. Verify that the file exists in the location specified in the INFOConnect database, and modify the database if needed.

8389117

IC_ERROR_ALREADYCLOSED

The INFOConnect Communications Manager has already been terminated. Close and restart the INFOConnect Shell.

No INFOConnect shell is running. This error occurs only from `IcTerminateShell` when `IcInitShell` has not been called. INFOConnect shell developers must call `IcInitShell` and `IcTerminateShell` in pairs. Contact Customer Support.

8388612

IC_ERROR_BADPARAMETER

Invalid parameter received. Contact the component's vendor for further information.

An INFOConnect procedure received an invalid parameter, such as an unexpected NULL string pointer or a buffer length less than the minimum required by the called procedure. Contact Customer Support.

8389219

IC_ERROR_BADTEMPLATE

Configuration of path template *template_name* is invalid. Choose Modify from Install Path Templates to update the corrupted data.

The specified path template configuration is corrupted. This error can occur as the result of disk corruption. Modify the path template and save it to overwrite the corrupted data.

8389220

IC_ERROR_CHAN_BUSY

Library *library_name* is still busy opening channel *channel_name*. Wait until the channel has opened and try again.

A library's `IcLibOpenSession` routine is being called before a previous call to the library's `IcLibOpenChannel` routine for that same channel has been completed. This can occur if `IcLibOpenChannel` waits for user input. Wait until the channel finishes opening and try to open the path again.

8389111

IC_ERROR_CHANNELINUSE

Channel *channel_name* already in use by *library_name* non-multiplexing library. The requested path cannot be opened at this time.

The library channel specified for the INFOConnect path being opened cannot be opened a second time. This error occurs from `IcOpenSession` when the library channel specified for the path is already being used by the specified non-multiplexing library. Close the path that is using this library channel before opening this path.

- 8389514** **IC_ERROR_COLON_PRESENT**
- Colon(:) not allowed in ID. Correct the ID and try again.
- The name of the component erroneously contains a colon. Modify the component so that its ID does not contain a colon and try again.
- 8389108** **IC_ERROR_INITICS**
- Unable to start INFOConnect. ABORTING.
- Contact the INFOConnect support representative for further information.
- The INFOConnect Manager cannot be run. This error occurs from IcInitIcs when an unknown error occurs during initialization. Try again. If this error persists, reinstall INFOConnect Connectivity Services or contact Customer Support.
- 8389115** **IC_ERROR_INMODIFY**
- Path *path_name* is currently being modified. You cannot establish a session with this path.
- You cannot open a path that is being modified. Save the changes to the path before using it.
- 8388613** **IC_ERROR_INTERNAL**
- Internal error detected at *string*. Contact the component vendor for further instruction.
- A non-fatal internal error has been detected. This error occurs when some layer of Connectivity Services detects an impossible or unlikely state. For debugging purposes, developers returning this error from an INFOConnect library must first call IcSetSessionError with the lpinsert1 parameter pointing to a location identification string. Contact Customer Support for more information.

8389508

IC_ERROR_INVALID_CONFIGREC

Invalid configuration record structure returned. Configuration aborted. Select Configure from the Configure Packages window.

A configuration record was invalid. This error occurs when the structure of the record does not match the structure expected by the INFOConnect database.

Using the INFOConnect Manager, click Packages from the Configure menu, click the package you were using when this error occurred, and click Quick Config to force a data upgrade to occur. If the error persists, contact Customer Support.

8389110

IC_ERROR_INVALIDPATH

Invalid path requested: *path_name*. Verify the path configuration.

You tried to use an invalid path to establish a session. This can occur if a session is configured to use a specific path automatically, and that path has been deleted from the INFOConnect database. Re-create the path or specify a different path for the session.

8388616

IC_ERROR_INVALID_WINCOMBO

Invalid window state combination. Contact the component's vendor for further information.

A request was made to open an INFOConnect accessory with an invalid window state (such as hidden and active, or maximized and background). This error occurs when one of these invalid combinations of window state options is passed to `IcOpenAccessory` or `IcRunAccessory` (through the `-W` option). (For more information, refer to `IcOpenAccessory` or `IcRunAccessory` in the *INFOConnect Development Kit Basic Programming Reference Guide*.) Contact Customer Support.

8388615

IC_ERROR_INVALID_WINOPTION

Invalid window state option. Contact the component's vendor for further information.

A request was made to open an INFOConnect accessory using unknown window state options. This error occurs when an invalid window state option is passed to `IcOpenAccessory` or

IcRunAccessory (through the -W option). (For more information, refer to IcOpenAccessory or IcRunAccessory in the *INFOConnect Development Kit Basic Programming Reference Guide*.) Contact Customer Support.

8389509

IC_ERROR_LIBRARY_CONFIG

The given ID is still configured in a path. Delete this path or reconfigure it without this library.

You tried to delete a library while it is still configured in a path. Delete or reconfigure the path before deleting the library.

8388617

IC_ERROR_MGR_BUSY

Communication queue full. Request ignored.

The INFOConnect Manager message queue is full. The accessory must relinquish control so that some messages can be delivered. Close the accessory to allow messages to be delivered and contact Customer Support.

8389213

IC_ERROR_NEWVERSION

This application requires Version *version_number* of the INFOConnect Communications Manager. Update the ICS software before using this component.

You tried to run a newer version of an INFOConnect accessory or application with an older version of the INFOConnect Manager. Install a newer version of Connectivity Services before using this application.

8389217

IC_ERROR_NOCHANDATA

Channel *channel_name* configuration data for library *library_name* missing. Have the Administrator modify the channel configuration for this library.

The specified library is missing the specified channel configuration data. This can occur as the result of disk corruption. Re-create or modify the library channel and try again.

8389116

IC_ERROR_NOCLOSE

The INFOConnect Communications Manager is not ready to terminate. Be sure that all dialogs are closed.

You tried to close the INFOConnect Manager when it was not possible, such as when the Select Path dialog box is still open. Close all INFOConnect dialog boxes and try again.

8389208

IC_ERROR_NOLIBLOAD

Unable to load *component_name*. The result code is *number*. Verify that this is a valid Windows code file.

The specified INFOConnect component cannot be loaded. Verify that the specified component is a valid Windows code file and that Windows itself is properly installed.

8389215

IC_ERROR_NOLIBRARY

Library *library_name* is not installed. Install the necessary library and try again.

The specified library is not currently installed. This error can occur if the Trace library is removed and you try to trace a session. This error can also occur if the Local library is removed and an application tries to use an accessory. Reinstall the necessary library and try again.

8388611

IC_ERROR_NOMEMORY

Memory Error. Free some memory and try again.

An attempt to allocate or access an INFOConnect memory block failed. This error occurs in low memory conditions. Close any unneeded applications and try again.

8389216

IC_ERROR_NOPATHDATA

Path configuration data for library *library_name* is missing. Modify the path configuration and try again.

The specified library is missing path configuration data. This error can occur as the result of disk corruption. Modify the path configuration and try again.

8389511

IC_ERROR_NOPATHID

Path ID missing. Verify the path ID and try again.

You tried to use a path that does not exist. Re-create the path or use a path that exists in the INFOConnect database and try again.

8389218

IC_ERROR_NOTEMPLATE

Path template *template_name* is not configured. Try to reconfigure the template.

A path is trying to use the specified path template, but it does not exist. This error can occur as the result of disk corruption. Reconfigure the path to use the correct path template, or reconfigure the path template and try again.

8389211

IC_ERROR_NOVERSION

Cannot verify version information. *Name* not loaded. Try reinstalling this product.

The INFOConnect version information cannot be verified. This error occurs when the specified file does not contain the required INFOConnect RCDATA version information in its resource file. It can also occur as the result of memory or disk corruption. Reinstall the specified product and try again. If the error persists, contact Customer Support.

8389118

IC_ERROR_PATHBUSY

Path *path_name* is currently active. Multiple instances of this path are not allowed.

The specified path cannot be used by multiple sessions simultaneously. Close the active session before trying to open another session using that path.

8389516

IC_ERROR_PATHID_EXISTS

Path ID already exists. Use a different ID or rename the existing path.

You tried to add a path using a path ID that already exists in the INFOConnect database. Specify a different name for this path, or delete the path that already uses this name and try again.

8389112

IC_ERROR_PICHANNELINUSE

Channel *channel_name* in use. Not sharable between *library_name* and *library_name* external interface libraries.

The specified library channel cannot be used by both of the specified external interface libraries simultaneously. This error occurs from `IcOpenSession` when you try to use a single multiplexing service library with two different external interface libraries over the same channel. Close the active session before opening a session using the other path.

8389210

IC_ERROR_PIVERSION

Library_name is not a valid INFOConnect external interface library. Library not loaded. Try reinstalling this product.

The specified file cannot be loaded as an external interface library because it does not properly identify itself as an INFOConnect EIL. Reinstall the library before trying to open a session with a path that uses this library. If the error persists, contact Customer Support.

8389113

IC_ERROR_PMCHANNELINUSE

Channel *channel_name* in use. Not sharable between *library_name* and *library_name* service libraries.

The specified library channel cannot be used by both of the specified service libraries simultaneously. This error occurs from `IcOpenSession` when you try to use a single multiplexing service library with two different external interface libraries over the same library channel. Close the active session before opening a session using the other path.

8389209

IC_ERROR_PMVERSION

Library_name is not a valid INFOConnect service library. Library not loaded. Try reinstalling this product.

The specified file cannot be loaded as a service library because it does not properly identify itself as an INFOConnect SL. Reinstall the library before trying to open a session with a path that uses this library. If the error persists, contact Customer Support.

8388618

IC_ERROR_RCV_BUSY

Station is still receiving. Request ignored.

A second request to receive data was made before the first one completed. The accessory should wait for a receive-done or receive-error message before requesting more data. Wait until the accessory receives data for the outstanding request before making another receive request.

8388610

IC_ERROR_REOPEN

Internal Error. Attempt to re-open external interface library. Contact the component vendor for further instruction.

An internal error occurred when you tried to reopen a communications device. Contact Customer Support.

8389609

IC_ERROR_SERVICE_NOT_AVAILABLE

Unavailable service requested: *service_name*. Verify the service name with the component's documentation.

You requested a service that is not supported. Verify that the specified service name is correct by referring to the component's documentation and try again.

8389512

IC_ERROR_SIZE_EXCEEDED

ID length limited to *number* characters. Correct the ID and try again.

The name for the component is too long. This error occurs when an ID exceeds IC_MAX***IDLEN. Type a shorter name and try again.

8389513

IC_ERROR_SPACE_PRESENT

Space not allowed in ID. Correct the ID and try again.

The name for the component contains a space, which is not allowed. Rename the component and try again.

- 8389515** **IC_ERROR_TILDE_PRESENT**
- Tilde(~) not allowed in ID. Correct the ID and try again.
- The name for the component contains a tilde (~), which is not allowed. Rename the component and try again.
- 8388609** **IC_ERROR_TIMERS**
- Too Many Timers. Terminate some timers and retry.
- An attempt to start a Windows timer failed because the maximum number of timers has already been reached. Close some Windows applications that are using the timer resource and try again.
- 8389608** **IC_ERROR_UNKNOWN**
- Unknown error encountered.
- An unknown error occurred. Contact Customer Support.
- 8389114** **IC_ERROR_UNOPENEDSESSION**
- Attempt to use unopened session. Verify the path configuration and clear the session, if necessary.
- The session is not yet available for communication. This error occurs when an application tries to use a session handle that does not exist or has not yet been established. (For more information, refer to the IC_SessionEstablished message type or the E_IC_SESSION_EST event type in the *INFOConnect Development Kit Basic Programming Reference Manual*.)
- Clear the session and verify that the path configuration is correct before reopening the session. You might need to restart the application or restart Windows to clear the session from memory.
- 8389221** **IC_ERROR_UPGRADE_WAIT**
- Library *library_name* is waiting for configuration data upgrade. Select Configure from the Configure Packages window.
- The data for a library whose data record format has changed has not been upgraded. Using the INFOConnect Manager, click Packages from the Configure menu, click the name of the library you were using when the error occurred, and click Quick Config. If

any dialog boxes appear, be sure to complete each dialog box and click OK rather than Cancel.

8389212

IC_ERROR_WRONGVERSION

Current version of INFOConnect does not support this version of *library_name*. Library not loaded. Upgrade the necessary software.

This version of the INFOConnect Manager does not support this version of the specified library. Reinstall the appropriate version of either Connectivity Services or the specified library and try again.

8388614

IC_ERROR_XMT_BUSY

Station is still transmitting. Request ignored.

A second request to transmit data was made before the first transmission completed. The application or accessory should wait for a transmit done or a transmit error type event or message before retransmitting.

Status Messages

C

In This Appendix

This appendix lists the status messages returned by the INFOConnect Manager and includes the following headings:

Status Messages Sent from an Accessory to a Library	80
Status Messages Sent from a Library to an Accessory	83
Status Messages Sent From the INFOConnect Manager to an Accessory	85
UTS-Specific Status Messages	86

The information in this appendix is taken from *INFOConnect Development Kit Basic Programming Reference Manual*, Appendix B, “Status Types and Statuses,” published by Unisys Corporation.

Status messages are in the form of a decimal number (such as 131074). The following sections contain tables that list the status numbers, their names, and a brief explanation of each.

Status Messages Sent from an Accessory to a Library

Blocking Status Messages

IC_STATUS_BLOCKING is an application-initiated status type that controls the blocking mode of those service libraries that support it. The application toggles the blocking mode on and off using the following status messages:

Number	Name	Meaning
262144	IC_BLOCKING_ON	Turns blocking on
262145	IC_BLOCKING_OFF	Turns blocking off

Applications that require blocking should either be altered to support non-blocking interfaces or refuse to support a session over a library that sets `sinfo.block_mode` to `FALSE`.

Connection Status Messages

The IC_STATUS_CONNECT status type commands the external interface library to alter the connection state. Whether a particular message is supported depends on the library. The connection states are identified by one of the following values:

Number	Name	Meaning
131072	IC_CONNECT_EOF	Request that no more data be sent (for use with TCP/IP). This status is supported only if <code>sinfo.reconnect</code> is <code>TRUE</code> .
131073	IC_CONNECT_CLOSE	Close the connection. This status is supported only if <code>sinfo.reconnect</code> is <code>TRUE</code> .
131074	IC_CONNECT_OPEN	Request to reopen the connection. This status is supported only if <code>sinfo.reconnect</code> is <code>TRUE</code> . If the library cannot honor this request, it should return an error from its <code>IcSetStatus</code> procedure.
131077	IC_CONNECT_STATUS	Request that the EIL display status information to the user. This is the status sent when the user clicks the Status button from the INFOConnect Manager Shell.

Refer also to IC_STATUS_CONNECT in “Status Messages Sent from a Library to an Accessory” on page 83.

Function Key Status Messages

This application-initiated status type sends terminal function key messages to the underlying layer of the communication session. The functions of the function keys depend on the libraries and the host connection. The function keys are defined by one of the following values:

Function Key	Value	Purpose
IC_FKEY_BREAK	458752	The Break key
IC_FKEY_1	458753	Function key 1
IC_FKEY_2	458754	Function key 2
.	.	.
IC_FKEY_23	458775	Function key 23
IC_FKEY_24	458776	Function key 24
IC_FKEY_MSGWAIT	458752 (same as Break key)	Uniscope®-specific BEL character key
IC_FKEY_SYSMODE	458775 (same as function key 23)	Uniscope-specific OS/3 system mode key used to put a session in system mode
IC_FKEY_WSMODE	458776 (same as function key 24)	Uniscope-specific OS/3 workstation mode key used to put a session in workstation mode

Focus Shift Status Messages

When multiple communication sessions are active, certain transports (such as COMS) require notification when the focus shifts among the applications associated with the session. This status message provides the mechanism for notifying the transport of such a focus shift. Sending this status message to a transport that doesn't use it will normally not cause any problems.

Appendix C Status Messages

These status messages must be sent by an application to the communication session by calling the set status procedure when `sinfo.focus_notify` is TRUE.

Number	Name	Meaning
196608	IC_REACTIVATE_ON	The application has received focus and <code>sinfo.focus_notify</code> is TRUE.
196609	IC_REACTIVATE_OFF	The application has lost focus and <code>sinfo.focus_notify</code> is TRUE.

A library that must be notified of an application gaining or losing input focus (such as COMS) should set `sinfo.focus_notify` to TRUE in the `IcLibGetSessionInfo` procedure. (The COMS library generates and transmits messages (?on...) when the current window changes.)

Applications without visible windows must either support these status messages or refuse to support a session over a library that requests this type of notification.

Status Messages Sent from a Library to an Accessory

Connection Status Messages The IC_STATUS_CONNECT status messages are typically issued from the external interface library and report the state of the connection. The connection states are defined by one of the following status messages:

Number	Name	Meaning
131074	IC_CONNECT_OPEN	The logical connection is available for bi-directional communication under the current configuration.
131073	IC_CONNECT_CLOSE	The logical connection is not available for bi-directional communication under the current configuration. This is the initial state of the session.
131072	IC_CONNECT_EOF	The logical communication session is physically closed, and no more data will be received (for use with TCP/IP).
131076	IC_CONNECT_ACTIVITY	The physical connection (but not necessarily the session) is functioning as expected.
131075	IC_CONNECT_NOACTIVITY	The physical connection is not functioning as expected.
131082	IC_CONNECT_BROKEN	The other half of two connected sessions has closed. For example, a DosLink session receives this status when its partner session is closed.
131083	IC_CONNECT_JOINED	Two sessions have been connected. For example, this is the status received when two DosLink sessions are connected.
131084	IC_CONNECT_SERVER	The server application (such as the DosLink Server) is ready to interact with the client.

Libraries should send these status messages only when the status of the connection changes.

For more information, refer to IC_STATUS_CONNECT in “[Status Messages Sent from an Accessory to a Library](#)” on page 80.

Appendix C Status Messages

Control Status Messages When initiated from an external interface library, a status of this type makes a request to the application. The requests are defined by one of the following status messages:

Number	Name	Meaning
327681	IC_CONTROL_ACTIVATE	This status requests that the applications window become active for user input. This occurs when the user selects the GoTo button on the user interface window.
327682	IC_CONTROL_RCVREADY	This status requests that the application perform a receive request. It indicates to the application that a received message must be delivered or it might be lost.
327683	IC_CONTROL_RCVAVAIL	This is a notification or advisory status indicating that a message is available but not deliverable due to the state of the application. The session might be blocked until the message is delivered.

Line State Status Messages This external interface library-initiated status type indicates the state of the underlying layer of the session. This status is generally used by T 27 terminal emulators such as T 27.

An IC_STATUS_LINESTATE status message is generated by the external interface library each time the line status changes. Pass the event to the application by calling `IcMgrSendEvent(...)`. Applications receiving this event might or might not process it.

The status messages are defined by one of the following values:

Number	Name	Meaning
2	IC_LINESTATE_XMT	The session is in transmit or transmit/receive mode.
3	IC_LINESTATE_RCV	The session is in receive mode.
4	IC_LINESTATE_LCL	The session is neither transmitting nor receiving.

Status Messages Sent From the INFOConnect Manager to an Accessory

Communication Manager Status Messages This status type originates from the INFOConnect Manager and conveys initialization or termination information. If the INFOConnect Manager closes, the accessory must call the INFOConnect initialization routine before calling any other INFOConnect procedures.

Number	Name	Meaning
393216	IC_COMMMGR_INITIALIZED	Status sent to all Windows applications when the INFOConnect Manager finishes initializing. INFOConnect accessories can now call the initialization routine, if necessary, before establishing INFOConnect sessions.
393217	IC_COMMMGR_TERMINATED	Status sent to all Windows applications when the INFOConnect Manager finishes terminating. All INFOConnect accessories should either close or call the initialization routine before establishing another session.
393218	IC_COMMMGR_QUERYEXIT	Status sent to all INFOConnect communications sessions when the user closes the INFOConnect Shell. If the application does not wish to close the session, it should cancel the exit by calling <code>IcExitOk(FALSE)</code> . Otherwise, call <code>IcExitOk(TRUE)</code> .
393221	IC_COMMMGR_QUERYSHUTDOWN	Status sent to all INFOConnect communications sessions when Windows is exiting. If the application does not wish to close the session, it should cancel the exit by calling <code>IcExitOk(FALSE)</code> . Otherwise, call <code>IcExitOk(TRUE)</code> .
393219	IC_COMMMGR_CANCELEXIT	Status sent to all INFOConnect communications sessions that previously received an <code>IC_COMMMGR_QUERYEXIT</code> status when at least one of the applications called <code>IcExitOk(FALSE)</code> .
393220	IC_COMMMGR_EXIT	Status sent to all INFOConnect communications sessions if <code>IcExitOk(FALSE)</code> is never called. The INFOConnect Manager will then exit.
393222	IC_COMMMGR_REINSTALL	Status posted to all windows by <code>INSTALL.EXE</code> when the INFOConnect Manager is being reinstalled.

UTS-Specific Status Messages

The IC_STATUS_UTS status can send and receive special messages to or from the INT1 and other transport libraries.

The library sends the status messages listed in the following tables to the application. The IC_RESULT_VALUE is interpreted as two subfields: IC_RESULT_SUBTYPE (subtype) and IC_RESULT_SUBVALUE (subvalue). A special macro, IC_MAKE_UTS_RESULT(t, v), is available to create an IC_RESULT from the standard context and from the IC_RESULT_TYPE and IC_RESULT_VALUE.

IC_UTS_SELECTION Status Messages Sent to and from Libraries

Number	Name	Subtype	Subvalue	Meaning
65543	IC_UTS_MSGWAIT	0	0x07	This is the Message Wait status.
65543	IC_UTS_MSGWAIT	0	0x20-0x6F 0x73-0x7F	These status messages request the selection of the given device ID (DID).
65649	IC_UTS_DESELECT_ACTIVITY	0	0x71	This status message is a request to deselect the current device.
65650	IC_UTS_DESELECT_DID	0	0x72	This status message is a request to flush and deselect the current device

Appendix C Status Messages

Status Messages Sent to Libraries

Number	Name	Sub-type	Subvalue	Meaning
69664-69743 and 69747-69759	IC_UTS_DVC_READY	0x10	0x20-0x6F 0x73-0x7F	These status messages indicate that a given device (DID) is ready.
69920-69999 and 70003-70015	IC_UTS_DVC_BUSY	0x11	0x20-0x6F 0x73-0x7F	These status messages indicate that a given device (DID) is busy.
70176-70255 and 70259-70271	IC_UTS_DVC_ERROR	0x12	0x20-0x6F 0x73-0x7F	These status messages indicate that an error has occurred at a given device (DID).
70432-70511 and 70515-70527	IC_UTS_DVC_NOTREADY	0x13	0x20-0x6F 0x73-0x7F	These status messages indicate that a given device (DID) is not responding.
73782	IC_UTS_POC	0x20	0x36	This status message indicates that power-confidence tests have completed (they send DLE 6 to the host).

Troubleshooting

D

In This Appendix

This appendix provides information on troubleshooting. It contains the following headings:

General Troubleshooting Procedures 90

General Troubleshooting Procedures

If you have problems running an application developed using the INFOConnect OCX, do the following:

- 1** Check the prerequisites for each method, event, or property. Some have prerequisites that must occur before the method, event, or property can be invoked.
- 2** Check your connections. Check your cable connections and make sure that they are securely attached.
- 3** Check your system. You might be using peripheral equipment or other software that might not be compatible with your application. Try disabling some of the other memory-resident applications.
- 4** Verify that your application is processing errors correctly. Refer to [Appendix B, “RptMsg Messages”](#) for information about errors generated by the INFOConnect OCX and the INFOConnect Manager. Refer to the online Help for each library for information about the errors generated by that library.
- 5** Consult your distributor. If you cannot identify and solve the problem without assistance, contact your product distributor. Call from a location where you have access to the problem PC..

Glossary

accessory	An application used with or by the INFOConnect Manager. For example, T27 and UTS are both accessories.
active path	An INFOConnect path that is being used by an INFOConnect accessory.
application interface library (AIL)	A library that provides special interaction with applications. Application interface libraries are not included in path templates, and are not used to create paths. For example, the ADMIN32 application interface library sets aside space in the 32-bit INFOConnect database for the groups and users created by the INFOConnect Database Editor.
control	An object that you use to create the user interface. Controls include list boxes, combo boxes, text boxes, check boxes, and radio buttons.
external interface library (EIL)	A library that provides the link between the INFOConnect Manager and an external transport service. For example, the TCP external interface library provides the link with the WinSock TCP/IP stack. Each INFOConnect path must use an external interface library.

Glossary

host	A mainframe, mini-computer, or information hub with which the PC communicates.
inactive path	An INFOConnect path that is not currently being used.
INFOConnect Connectivity Services	A package of INFOConnect products that includes the INFOConnect Manager and other INFOConnect software.
INFOConnect OLE custom control (OCX)	A custom control that consists of methods, events, and properties used to write applications that can send data over INFOConnect paths.
INFOConnect database	<p>A file usually located in the Windows directory that contains information about all the INFOConnect packages that have been installed, as well as all the path templates, library channels, and paths that have been created.</p> <p>Information about 16-bit and 32-bit products are kept in separate INFOConnect databases. The name of the database for 16-bit products is INFOCONN.CFG. The name of the database for 32-bit products is IC32.CFG.</p>
INFOConnect Manager	<p>The application that runs in the background each time you run an INFOConnect product and controls interaction between accessories and transports.</p> <p>You can also use this application to create, modify, and delete paths; set administrator and user-level security; control user access to configuration; and set user preferences. The INFOConnect Manager also lets you switch between sessions, clear sessions, and view session status.</p>
library	A piece of communication software that can be used singly or in combination with other libraries to enable communication between PCs and hosts, and between applications on the PC.
library channel	A named collection of configuration settings that apply to all paths that use that library channel.
path	A named set of configuration options that define an INFOConnect communication connection between a PC and a host. A path

consists of a path template and other configuration data associated with the library(s) listed in the path template.

- path template** A combination of one or more libraries. If a library in the path template can have library channels, the path template can indicate which library channel to use.
- service library (SL)** A library that provides communication processing for an external interface library. For example, the TCP-A service library provides additional processing so that a PC can communicate with an A Series host.
- session** A communication connection between a PC and a host.
- terminal keystroke** A keystroke that a terminal sends to a host. Terminal keystrokes (such as Transmit) typically do not exist on a PC keyboard and must be emulated using various keystroke combinations.

Index

16-bit VBX applications, converting to 32-bit OCX [9-11](#)

A

AboutBox method [24](#)

Accessories

status messages sent from a library [83-84](#)

status messages sent from the INFOConnect Manager [85](#)

status messages sent to a library [80-82](#)

UTS-specific status messages [86](#)

Action property [25-27](#)

AutoReceive property [28-29](#)

B

BAS files [12](#)

Blocking status messages [80](#)

C

Communication manager status messages [85](#)

Connection status messages [80, 83](#)

Control status messages [84](#)

Custom controls (see OCX)

Custom, in the Properties window [13](#)

Customer Support [90](#)

D

Data filters

overview [2, 58](#)

using the sample data filter [59](#)

E

Error messages

INFOConnect OCX [62-63](#)

INFOConnect severe errors [67-77](#)

INFOConnect termination errors [64-66](#)

Event procedures, defining [14-16](#)

Events

overview [4](#)

ReceiveDone [45](#)

RptMsg [46](#)

SessionClosed [48](#)

SessionEstablished [49](#)

Status [51](#)

TransmitDone [54](#)

UserEvent [55](#)

F

FireUserEvent method [30](#)

Focus shift status messages [81](#)

Function key status messages [81](#)

G

GetPathNames method [31-33](#)

Index

H

Help, online ix
hSession property 34
hWndVBLList property 35

I

ICGLOBAL.BAS 12
ICSCLT32.OCX 2
INFOConnect custom control for Visual Basic
(see VBX applications)
INFOConnect OLE custom control (see OCX)
INFOConnect severe errors 67-77
INFOConnect termination errors 64-66
IsEstablished property 36

L

Libraries
 status messages sent from an
 accessory 80-82
 status messages sent to an
 accessory 83-84
Line state status messages 84

M

Methods
 AboutBox 24
 FireUserEvent 30
 GetPathNames 31-33
 OpenAccessory 39-41
 overview 4
 Receive 43-44
 SendStatus 47
 SetPathList 50
 Transmit 53

O

OC30.DLL 2
OCX
 adding the custom control to a project 12
 converting 16-bit VBX applications to
 32-bit OCX 9-11
 errors 62-63
 overview 2-5
 prerequisites 8
 viewing the version number 24
Online Help ix
Open property 37-38

OpenAccessory method 39-41

P

PathName property 42
Projects, adding INFOConnect OCX to 12
Properties
 Action 25-27
 AutoReceive 28-29
 defining 13
 hSession 34
 hWndVBLList 35
 IsEstablished 36
 Open 37-38
 overview 5
 PathName 42
 StatusMessage 52
 XmitData 56
Protocol handlers (see Data filters)

R

Receive method 43-44
ReceiveDone event 45
 sample code 14
RptMsg event 46
 errors handled by 62-77
 sample code 15

S

Sample application
 reviewing the source code 19-21
 running 21
Sample code
 data filters 59
 ReceiveDone event 14
 RptMsg event 15
 sample application 19-21
 SessionClosed Event 14
 SessionEstablished event 14
 Status event 16
 TransmitDone event 14
 UserEvent event 16
SAMPLE.EXE 21
SendStatus method 47
SessionClosed event 48
 sample code 14
SessionEstablished event 49
 sample code 14
SetPathList method 50

Severe errors 67-77
Status event 51
 sample code 16
Status messages
 blocking status messages 80
 communication manager status
 messages 85
 connection status messages 80, 83
 control status messages 84
 focus shift status messages 81
 from a library to an accessory 83-84
 from an accessory to a library 80-84
 from the INFOConnect Manager to an
 accessory 85
 function key status messages 81
 line state status messages 84
 UTS-specific status messages 86
StatusMessage property 52
Support, technical 90

T

Technical support 90

Termination errors 64-66
Timers 17-18
Transmit method 53
TransmitDone event 54
 sample code 14
Troubleshooting 90

U

UserEvent event 55
 sample code 16
UTS-specific status messages 86

V

VBX applications, converting to OCX 9-11
Version number, OCX 24
Visual Basic (see VBX applications,
 converting to OCX)

X

XmitData property 56

