
G3 Fax CAPI ActiveX OCX - Interface Definition

OCX (C) 2002 Jan Florian Wagner

1. Intro

This OCX is licensed with terms as specified in the LGPL licence (see end of this document). If you use this OCX or the source code in any public or commercial program or package, you have to include the complete OCX source code. The copyright and licence notices must be left intact.

Check once in a while if a newer version of this OCX is available on my homepage at <http://www.hut.fi/~jwagner/cthp/>. If you find any bugs or have made corrections or enhancements to the source code, feel free to email me (jwagner@cc.hut.fi).

2. System requirements

- PC running on Win95 or newer
- ISDN card, including the CAPI2.0 drivers from the card manufacturer
- Windows programming platform which supports ActiveX (Visual Basic, Borland C++, MS Visual C++, Visual Fox Pro, MS Access, etc.)

3. Features

Version 1.5.0:

- display calling number and called number
- display call type
- update your PC clock to ISDN network time (outgoing calls)
- get ISDN card info (supported supplementary services)
- send a G3 fax file from an existing SFF file
- receive a G3 fax, stored as a SFF file
- notifies about call setup and fax transmission progress
- optional fax status window with a progress bar
- includes complete OCX source code with plenty of comments, and some example code for using the OCX.

The OCX should work with VB4.0 and higher, and all other programming languages with OCX support.

All questions, suggestions, improvements, feedback, errors found, info on how you used this OCX and if you liked it, are welcome! jwagner@cc.hut.fi

4. The CAPI OCX interface

4.1 Events:

IncomingCall (*CalledNr, CallingNr, Calltype*)

Indicates that there's an incoming call (any call type).

IncomingPhoneCall (*CalledNr, CallingNr*)

Indicates that there's an incoming phone call, or an analog network call (which *could* be an analog fax call).

IncomingFaxCall (*CalledNr, CallingNr*)

Incoming G3 fax call. The call is answered and receiving starts if the OCX property AcceptFax is set to True.

FaxReceived ()

The fax has been received, the line is still open. The fax is stored in the file given by the OCX property SaveToFile.

FaxSent (*fullSuccess As Boolean*)

Indicates successful transmission of a SFF fax file over a connection. If fullSuccess is True, the fax was transmitted completely, if False, only partially or there was some other transmission error.

CallProgress (*Info\$*)

Passes information about the call progress to the client application. T.ex. "connecting", "b3 disconnected" etc.

Disconnected ()

The call has been disconnected.

DebuggingMessage (*msg\$*)

Info on internal CAPI message handling of the OCX, facility info and more. Useful for debugging, mainly.

4.2 Properties:

AcceptFax [Boolean] {write}

Set to True to answer to incoming fax calls. Can be set during the IncomingFaxCall event where the caller number may be checked (t.ex. against a blocklist).

CalledNumber [String] {read}

The number that was called by the incoming call (your own number).

CallerNumber [String] {read}

The originating number of the of incoming call (callers number).

CAPIVersion [String] {read}

CAPI version, should return "2.0"

CAPISerial [String] {read}

Serial number of the CAPI or ISDN adapter.

SaveToFile [String] {write}

The path and filename of a new file, into which an incoming fax should be stored.

ShowAllCalls [Boolean] {write}

When set to true, and after this calling startListen(), all types of incoming calls will be indicated (useful as CallerID). If set to False, only fax calls are shown.

ShowStatusWindow [Boolean] {write}

Set to True to show a small fax status window with a transmission progress bar, call numbers, etc. The status window appears when fax transmission starts.

ManufacturerVersion [String] {read}

Card manufacturers release version of their CAPI 2.0 driver, for example "3.34" for the Teles CAPI 3.34.

UseNetworkTime [Boolean] {write}

Set to true if you want to update the PC clock to the ISDN network time. Only works for outgoing calls, and only if the ISDN network supports the feature.

4.3 Methods:

StartListen () As Boolean

returns: True on success

Listen for incoming calls. If the ShowAllCalls property is set to False, only fax calls are shown.

StopListen () As Boolean

returns: True on success

Stop listening for incoming calls.

StartCapi () As Boolean

returns: True on success

Initializes the OCX – call it first. This function registers the client application, reads out CAPI version information and checks the ISDN controller profile.

StopCapi()

Unregister application.

Message2Text (MainCmd As Byte, SubCmd As Byte) As String

returns: string description of message

Returns the name of the CAPI message specified by MainCmd and SubCmd. Useful for debugging.

SendFax (CalledNumber\$, OwnNumber\$, FaxHeader\$, SenderName\$, FileToSend\$) As Boolean

returns: True if successful

Initiates a new fax call to CalledNumber. FaxHeader and SenderName should contain a short text with identification info. FileToSend is the path and name of the SFF file to be transmitted.

isConnected () As Boolean

isConnectedWith () As String

isConnecting () As Boolean

isDisconnecting () As Boolean

isSending () As Boolean

isReceiving () As Boolean

Some useful functions to check the current status of the OCX.

supports_3PTY (c As Integer) As Integer
supports_CallDeflect (c As Integer) As Integer
supports_CallForward (...) As Integer
supports_CallTransfer (...) As Integer
supports_DTMF (...) As Integer
supports_HoldRetrieve (...) As Integer
supports_SupplementaryServices (...) As Integer
supports_SuspendResume (...) As Integer

returns: -1=wrong controller, 0=not supported, 1=supported

Some useful functions to check the DTMF and supplementary services capabilities of the ISDN card and driver.

The parameter *c* is the number of the controller, usually 1. For example
result = supports_3PTY(1)

5. Usage examples

Here's a very short description on how to get started with the OCX in your own programming project. In addition to this, there is an example VB program included with the OCX source code. The demo source should demonstrate most of the OCX features. Here is a quick start-up guide:

First, start the OCX.

```
Call FaxCapi32.startCapi()
```

Decide whether to listen for all calls (CallerID) or just show fax calls.

```
FaxCapi32.ShowAllCalls = False      '// only fax calls
```

Tell the OCX to listen for incoming calls.

```
Call FaxCapi32.startListen()
```

For receiving a fax, just wait for an incoming fax call, which will then cause the event FaxCapi32.IncomingFaxCall(...). Inside this event, set FaxCapi32.AcceptFax = True, and specify where to save the fax file. You can also set ShowStatusWindow to display a call progress dialog window.

```
Sub FaxCapi32_IncomingFaxCall(CalledNr As String, CallingNr As String)
    FaxCapi32.AcceptFax = True
    FaxCapi32.SaveToFile = "C:\DEMO.SFF"
    FaxCapi32.ShowStatusWindow = True
End Sub
```

After the fax has been received, the OCX generates a FaxCapi32.FaxReceived() event. Finished.

For sending a fax, use

```
FaxCapi32.SendFax("123-called-nr", "00-own-nr", "Nice fax title", "It's me", "C:\SEND-THIS.SFF")
```

After the fax has been sent, there's an FaxCapi32.FaxSent(bSuccess) event. If bSuccess=True, sending the fax was successful.

To use the OCX for simple caller ID: all non-fax calls will generate an FaxCapi32.IncomingCall(...) if you use

```
FaxCapi32.ShowAllCalls = True
FaxCapi32.startListen()
```

6. Licence info

```
'//
'//      ISDN G3 Fax CAPI ActiveX Control, Version 1.5.0
'//
'//      (C) 2002 by Jan Wagner
'//      jwagner@cc.hut.fi
'//      http://www.hut.fi/~jwagner/cthp/
'//
'//      - the user is granted a licence according to
'//      terms specified under the GNU Library General Public Licence
'//      ('licence-lgpl.htm' in the 'Licence' directory)
'//
'//      - the full OCX source code must be included in all redistributions
'//
'//      - without a legal, written permission from the author, the sale,
'//      rent, and/or lease of this OCX is explicitly forbidden. It may
'//      be used in commercial programs, though, if the GNU LGPL is obeyed.
'//
'// This library is free software; you can redistribute it and/or
'// modify it under the terms of the GNU Library General Public
'// License as published by the Free Software Foundation; either
'// version 2 of the License, or (at your option) any later version.
'//
'// This library is distributed in the hope that it will be useful,
'// but WITHOUT ANY WARRANTY; without even the implied warranty of
'// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
'// Library General Public License for more details.
'//
'// You should have received a copy of the GNU Library General Public
'// License along with this library; if not, write to the
'// Free Software Foundation, Inc., 59 Temple Place - Suite 330,
'// Boston, MA 02111-1307, USA.
'//
```