



PictureTel LiveDTK V2.1 Sample Application

Visual Basic 5.0

May 1998

[Enhancements to the LiveDTK 2.0 Sample Application](#)

[Overview of Enhancements](#)

[The NetMeeting Wrapper OCX Integration](#)

[Additional Visual Basic 5.0 Sample Application Suggestions](#)

Enhancements to the Visual Basic LiveDTK 2.0 Sample Application

Overview of Enhancements

Two major changes to the LiveDTK 2.0 sample application for Windows 95 are: use of NetMeeting 2.1 for data conferencing, and use of NetMeeting 2.1 T-120 stack for multi-conferencing. In addition enhancements were made to the sample application as to exercise as much of standard DTK features that might be of interest to prospective developers.

Controls embedded in the LiveShare Plus OLE Control (Ispintf.ocx) were replaced by their equivalents available through Microsoft NetMeeting 2.1. The process has been simplified by the use of a NetMeeting Wrapper Control (manager.dll). This particular control is not to be confused with the "NetMeeting OCX". The NetMeeting Wrapper Control was designed to present the NetMeeting API in way that would allow current users of the LiveDTK a simplified, and straightforward upgrade path.

The NetMeeting Wrapper OCX Integration

The following steps integrate the new NMW OCX:

1. The LSP OCX controls were removed from the application. Verify that your project.vbp file no longer contains reference to this .ocx file. Most of these controls appear to users as pushbuttons that provide access to the whiteboard, message board, application sharing, remote control, shared clipboard, and file transfer functionality of LSP. It is important to note that currently NetMeeting does not support Shared Clipboard and Remote Control functionality, hence this features have been dropped from the current sample application. A workaround can be produced for both functions, i.e., for remote control by either sharing the desktop and then enumerating and sharing all the available window handles; and for shared clipboard by opening a transparent data channel (Virtual CommPort), and monitoring the changes in the regular clipboard.
2. The Call Interface OCX is still used to make a physical call but no longer to connect to a data conference. NetMeeting Wrapper Control has to be used to make the NetMeeting on both sides connect. Make sure that Application Sharing functionality of NetMeeting has been installed on both machines. This is a separate installation procedure on the current, NetMeeting 2.1, install. Instead of lspintf.ocx, manager.dll has to be added to the development workspace. If manager.dll has been properly registered it will appear in the list of all available OLE/ActiveX controls. Select it and the place on the same location where lspintf.ocx controls were previously.
3. NetMeeting Wrapper Control is a superset control for the entire LiveShare Plus functionality. It is a transparent control that includes all the relevant multi-conferencing and data sharing functionality in the format formerly used by the lspintf.ocx control, and the conference.ocx control of the LiveDTK 2.0. The control gets activated by having its Init() method called in the Form_Load(). A NetMeeting conference is established by using NetMeeting Wrapper Control's Connect() method when CallConnected event from the Call Interface (callintf.ocx) is received. It can take several seconds for NetMeeting to get fully connected. A new event, ConferenceCreated, was added to the NetMeeting Wrapper Control to signal that NetMeeting conference has been fully established. The NetMeeting conference must be in progress for any of the subsequent calls to the NetMeeting Wrapper Control to succeed.
4. The NetMeeting Wrapper Control replacements for the LSP controls were following:
 - o For the Whiteboard and MessageBoard , the replacement was simply changing the name of the new control to Manage1. LaunchWhiteBoard() and Manage1.LaunchMessage() respectively.
 - o The File Transfer migration is not quite so straightforward. Part of the problem is that NetMeeting 2.1 does not have a FileTransfer User Interface component, as is available through LiveShare Plus 4.0. Cftintf1.LaunchFileTransfer is no longer available. All its functionality now has to be added. Nevertheless, NetMeeting Wrapper Control provides for a SendFiles method that has similar syntax as CftIntf1.SendFiles. You will have to add a custom file selection dialog that would return a string with the paths and names of all the files destined for transfer. Manage1.SendFiles can send only one file at the time. Repeat this call if multiple files have been selected. Unfortunately, NetMeeting does not operate with a folder, as has been the case with LiveShare Plus 4.0.
 - o Migration of the Share and Unshare LSP controls is different in the sense that NetMeeting does not have an embedded SelectAppToShare method equivalent. To obtain the former LSP behavior, a dialog box has to be created, and/or the behavior of the Mouse object has to be tracked, in order to figure out which application the user wants to be shared or unshared. NetMeeting Wrapper Control supports a ShareableApps() method call in order to return a list of

shareable applications. `ShareApp()` method with the window handle of the selected item functions similarly as the LSP call `CshareIntf1.ShareApp`. There is a similar correspondence with the `UnshareApp()` method. In the sample application, we maintain only one list of shareable/unshareable applications, and we rely on the user to keep track of these. You may want to keep track of the shared application, in order to properly dispose of them at the later stage.

5. The NetMeeting Wrapper Control automatically disconnects from the NetMeeting conference when the call is disconnected, so no additional work is required.

Additional Visual Basic 5.0 Sample Application Suggestions

1. Allow PictureTel events (such as `CallConnected`, `ChannelConnected`, `CallIdle`, etc.) to return, that is if much code is associated with the activity of any single event, start a message queue or a timer control that would allow PictureTel event to return. This is the best way not to hinder the performance and the stability of the system.
2. To create your own custom call answering box, following properties have to be set:

```
CallIntf1.AutoAnswer = False  
Incoming1.AutoAnswer = False  
Incoming1.AutoShowDialog = False  
CallProg1.AutoShowDialog = False
```

System will wait after `CallOffering` event has been received.

If you want the ringer to continue ringing until the user makes a decision on the further progress of a call, you can continue to cycle through the `CallIntf1.Ring` method.

`CallIntf1.AnswerCall(hCallHandle)` or `CallIntf1.RejectCall(hCallHandle)` will then accept or reject the call.

3. To operate the three addressbook controls, the following properties have to be linked:

```
ABList1.punkDataCtrl = ABData1.Object  
ABEntry1.punkDataCtrl = ABData1.Object  
ABEntry1.PunkListCtrl = ABList1.Object
```

4. `ABData1.GetLocationNumberData` method of the addressbook returns formatted dialing number data. Make sure you reformat it into purely numbers before passing the data to `CallIntf1.Connect` or `CallIntf1.ConnectToConference` methods.
5. In `CallIntf1.Connect` method, `ChannelRate` setting can default to 0, or 64K call per each channel. PictureTel system will automatically downgrade the call to 56K per channel if a restricted switch is encountered anywhere on the route. `HConnection` parameter in this method should not be used as a valid identifier of the call handle. Use `ConnectionCreate`, or `CallConnected` events for proper information on the current call handle.

6. Maintain the list of debug and error messages, as well as a list of all the events received by your PictureTel system. This information will be later on of much use when qualifying your problems to your PictureTel support professionals.

For further information on the NetMeeting Wrapper Control, refer to Using the NetMeeting Wrapper OCX.



Copyright © 1998, PictureTel Corporation. All rights reserved.