Chapter 5
Popup Video Services

The Popup Video Services provide message boxes, input dialog boxes, and status dialog boxes which allow other 32-bit client modules to interact with the user in a OS-independent fashion. These functions are cross-platform, so that an NLM calling them does not need to worry about interaction with the operating system or other devices.

This chapter describes the Popup Video Services from the user's perspective, and describes their intended use and the API for using them.

Popup Video Introduction

NIOS provides a layer of operating-system independence for system-level programs, including the NetWare Client. Consequently NIOS includes functions such as file I/O and memory management, and would be incomplete without a set of user-interface functions. This is where Popup Video Services come in.

Popup Video Services includes three types of popups: message boxes, input dialog boxes, and status dialog boxes. These objects will operate much like the corresponding Windows objects, with possible restrictions in the non-Windows environments.

Message Box Architecture

Popup Video Services provides the programmer with tools for communicating with the user. These tools are the boxes described in the following sub-sections.

Message Box

Following is a diagram depicting the basic elements of the message box and an example of how the user would see it.

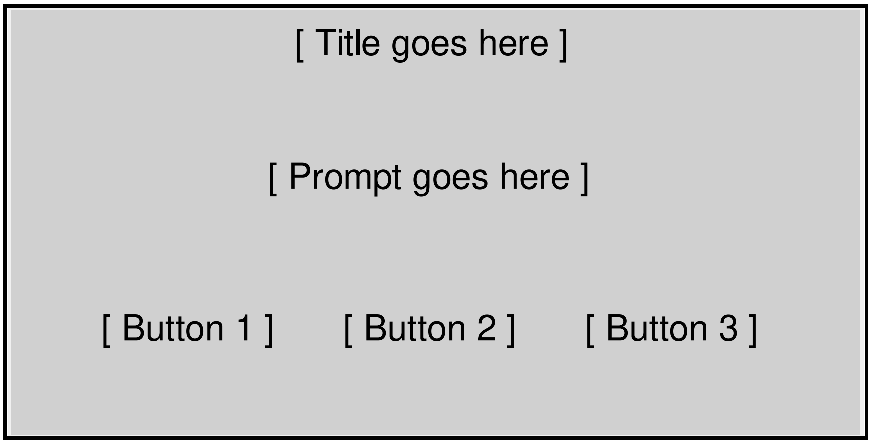
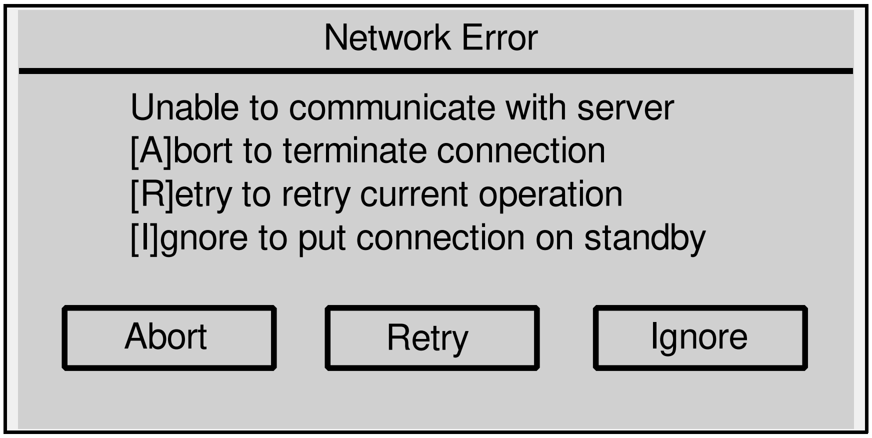


Figure 5.1: Message Box Elements



Network Error

Unable to communicate with server
[A]bort to terminate connection
[R]etry to retry current operation
[I]gnore to put connection on standby

Figure 5.2: Example Message Box

Input Box

Following is a diagram depicting the basic elements of the input dialog box and an example of how the user would see it.

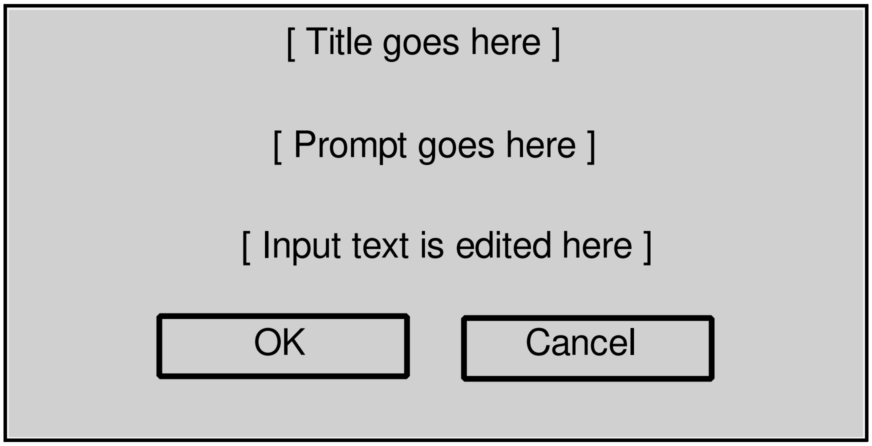
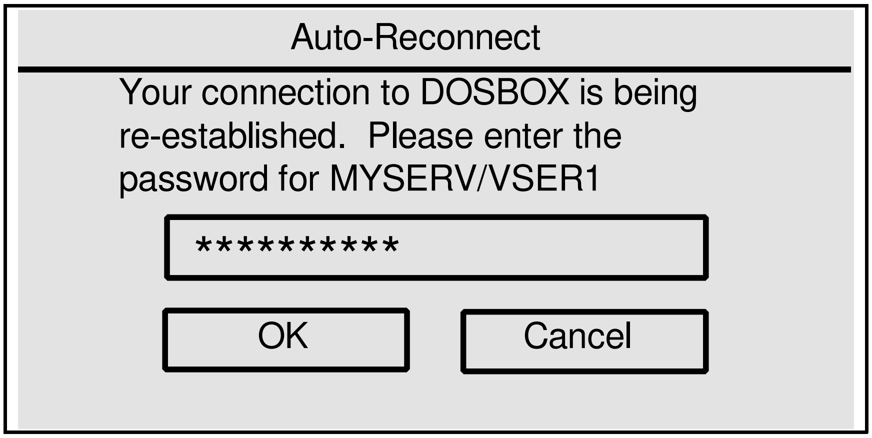


Figure 5.3: Input Box Elements



Auto-Reconnect

Your connection to DOSBOX is being re-established. Please enter the password for MYSERV/VSER1

****

Figure 5.4: Example Input Box

Status Dialog Box

Following is a diagram depicting the basic elements of the status dialog box and an example of how the user would see it.

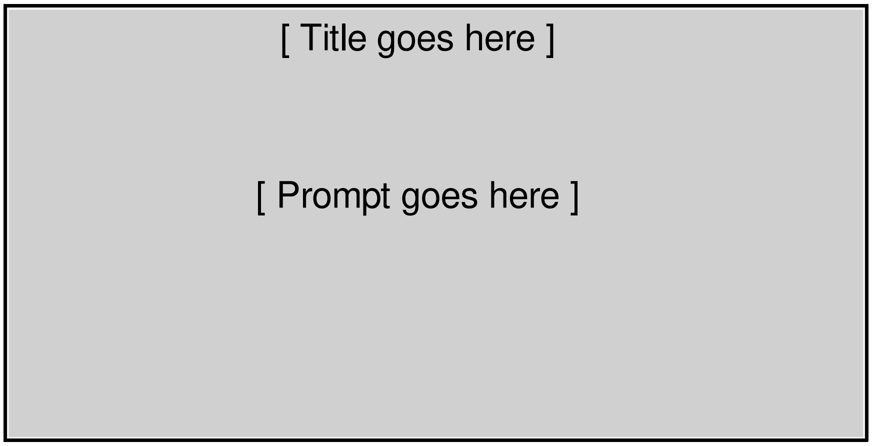
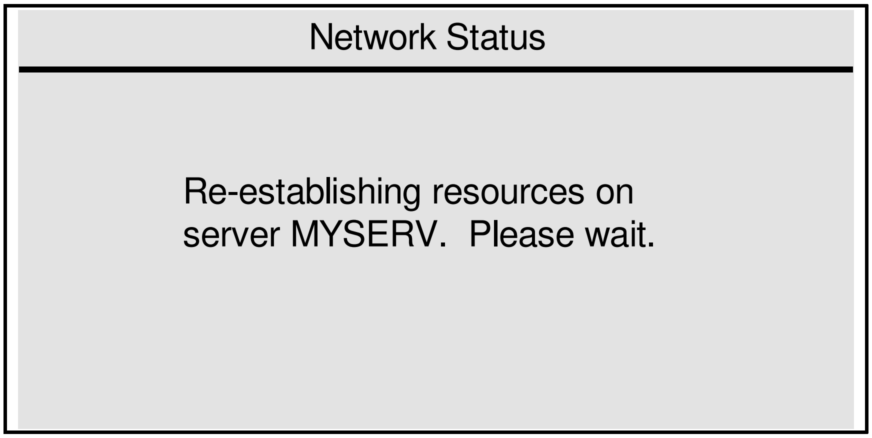


Figure 5.5: Status Dialog Box Elements



Re-establishing resources on
server MYSERV. Please wait.

Figure 5.6: Example Status Dialog Box

Popup Video Design Description

GUI Environment

When an NLM calls one of the Popup Video routines, it need not worry about what video mode the machine is in, and so on. That will all be handled by the popup routine.

Note: Currently, the DOS-only (this is not a DOS box under MS Windows) graphics modes are unsupported.

In order to provide support for the native operating environment GUI (for example, Windows or OS/2) an application registers to handle popups when the GUI environment is active. For example, under Windows, NWPOPUP.EXE registers with Video Popup Services to handle the message box, input dialog box, and status dialog box routines. Then, whenever an NLM makes a request of the Video Popup Services, a Windows dialog box is displayed.

The routines in the Popup Video Services are designed after the Windows routines of the same function so that they will operate naturally in a Windows or Windows-like environment. This should port well to operating systems like OS/2, which include a native GUI.

DOS Text Mode

In DOS text mode the Popup Video Services will pop up a display window that overlays the existing textual data. The data under the window will be restored when the window is closed. The "buttons" in this mode are textual but they are fashioned after the Windows buttons as closely as is feasible. DOS text mode does not support the icons that may be selected as part of the button mask, nor does it support the mouse device.

DOS Graphics Mode

Due to the number of DOS graphics modes available and the lack of a common interface for these modes, the user interface primitives provide only limited support for the DOS graphics mode. Although the functions are capable of writing to the video in DOS graphics mode, it is impossible to save the video data in all of the possible modes. However, we will attempt to support several popular video modes and add others as the need arises.

As a catchall, a BIOS video out routine will be capable of writing to the screen, regardless of the mode, but there will be no restoration of the screen. This is identical to the way DOS handles critical errors in graphics mode.

Windows DOS Boxes

The Popup Video Services are callable in the context of a Windows DOS box. The state of the DOS box could be full-screen, windowed, or miniaturized (icon only). Windows treats all of these cases the same when a critical error occurs in a DOS box. When this occurs, the DOS box switches to full screen mode and displays the error. The Popup Video Services simulates this behavior by using the Windows message mode (that is, full screen text for VxDs).

Popup Video API Calls

Popup Video Services provide the following functions.

Function	Description	Remarks
NiosVidMessageBox	Allows the caller to display a title, a prompt, and a variety of buttons. GUI mode may also support icons.	This is the recommended method of handling critical errors. (*This may not be supported in DOS Graphics mode.*)
NiosVidInputDialogBox	Allows the caller to display a title, a prompt, and an editable string. OK and Cancel buttons are pre-selected.	If the button selected is Cancel, then the editable string is left unchanged. (*This may not be supported in DOS Graphics mode.*)
NiosVidCreateStatusBox	Allows the caller to display a title and prompt in a modeless (asynchronous) dialog box. No buttons are displayed as there is no user interaction.	This is the recommended method of displaying status information. (*This may not be supported in DOS Graphics mode.*)
NiosVidDestroyStatusBox	Destroys a modeless (asynchronous) dialog box.	*This may not be supported in DOS Graphics mode.*
NiosVidUpdateStatusBox	Allows the caller to update the title and prompt for a modeless (asynchronous) dialog box.	*This may not be supported in DOS Graphics mode.*
DosVid16DeregisterGuiCB	Cancels a GUI callback. This function is Windows-specific.	There are some ring-3/ring-0 issues which are involved in using this function.
DosVid16RegisterGuiCB	Registers a GUI callback. This function is Windows-specific.	There are some ring-3/ring-0 issues which are involved in using this function.

A driving concept behind the design of the NIOS Client was creating modular, reusable, and platform-independent software.

To that end, all NIOS Client modules are in an NLM executable format and can be loaded and unloaded as necessary.

NIOS Client system modules:

- Are dynamically loadable and unloadable.
- Use NLM executable format.
- Run exclusively in a 32-bit flat memory model.
- Allocate memory that is guaranteed not to move or be discarded.
- Are fully language enabled.
- Are configured by utilities that use a configuration file.
- Require little or no static configuration information from the user. For example, it is no longer necessary to supply the number of open IPX sockets. As more sockets are opened, IPX dynamically allocates more memory to handle the additional sockets.
- Are written in C and Assembly.
- Run on single-processor Intel 386/486/Pentium systems.

Also included in this chapter is a functional description of the new NIOS logging feature.

NIOS Client Model

This section describes the cross-platform NIOS client model as illustrated in Figure 3.1; the sections that follow discuss each component of the model.