Chapter 6
Message Design Specification

Abstract

Message sends, receives, and displays messages broadcast to users. It also provides functions to allow application programs to send and receive user broadcast messages.

Contents

Introduction

In the 16-bit clients (Netx and VLMs), messages appear as text on either the top or bottom line of the screen, and in Windows as a dialog box from NWPOPUP.EXE. These messages can be sent from the NetWare server console via the SEND command or from another user using the SEND.EXE program or something similar.

The Message module handles the reception of these user broadcast messages, including displaying the message and prompting the user for input to clear the message. This gives the Client32 Requester the same functionality as the previous DOS requesters. The Message module uses the NiosVid*xxx* services to display the message in a popup window.

The Message module also provides a common set of functions for sending messages to other users. This gives application programs the ability to use a single API for different versions of NetWare and possibly non-NetWare networks.

Design Description

Message is made up of two parts, message handling and the message API. The message handling portion receives and displays user broadcast messages via the NIOS Video Popup services. The message API manages message delivery and local message handling options.

Component-Level Design and Functionality

The message handling portion of the Message module handles user broadcast notifications which originate in the lower-level transports. These notifications let the Requester know that a user has a broadcast waiting at a server. In the event that this user broadcast is a directed broadcast, as in Personal NetWare, then the notification will also include the message itself.

Once the Message module has received the notification of a waiting message, it generates a NESL event to offer other applications the opportunity to retrieve and handle the message. NETX will generate an interrupt 2F to notify applications that a message is waiting and will also support the concept of a message handler callback for compatibility. If either the interrupt

2F or callback indicates the message has been handled then NETX will consume the event. If the event is not consumed the Message module will retrieve and store the message in its message queue.

If the message queue is not empty, a routine will display the message at the head of the message queue using the NiosVideo routines. The display routine will handle user keyboard input and message timeout.

If a message is received which is a duplicate of an already pending message, the second message will be deleted to avoid unnecessary user interruption. This feature defaults to ON, but may be disabled through the REMOVE REDUNDANT MESSAGES option. (See the Configuration File Information section below for more details.)

External API Description

The Message external API includes the functions necessary to send and receive messages as well as manage the local message handling options. The following functions make up the external API for the Message module:

API	Description
SendMessage	Sends a message to one or more workstations via a common server connection.
SendDirectedMessage	Sends a message directly to a single workstation.
GetMessage	Retrieves a user broadcast message stored on a specified connection.
GetDirectedMessage	Retrieves a user broadcast message which has been stored in the workstation.
GetMessageMode	Returns the message handling mode for a specific connection or for the default case.
SetMessageMode	Sets the current message handling mode for a specific connection or for the default case.
GetMessageTimeout	Returns the number of milliseconds configured for the message to be displayed before the client automatically clears the message.
SetMessageTimeout	Sets the number of milliseconds configured for the message to be displayed before the client automatically clears the message.

Structures, Global Variables, Definitions

There is no pertinent information for this section.

Configuration File Information

Message removes duplicate messages. This feature defaults to ON but may be disabled by using the following configuration file parameter in the NET.CFG file:

```
REMOVE REDUNDANT MESSAGES=OFF
```

Deliverables Information

Executables

MESSAGE.NLM	Executable NLM module
MESSAGE.MSG	Message file

Product Source

MESSAGE.C	Source file
MESSAGE.H	Header file
MAKEFILE.MAK	Make file
MESSAGE.IMP	Import list file
MESSAGE.EXP	Export list file
M.BAT	Make batch file

Unit Test Source & Documents

MSGTEST.NLM	Automated unit test NLM which executes each of the exported functions and performs validation checks
MSGTEST.C	Source to automated test