Chapter 1 About This Guide

The **NetWare Peripheral Architecture Functional Specification and Developer's Guide** is a comprehensive guide describing the technical requirements and procedures for building device drivers under the NetWare Peripheral Architecture (NWPA). The purpose of this chapter is to define the audience, outline the presentation format, and explain the printing conventions of this manual to aid in assimilating its information.

1.1 Audience

The **NetWare Peripheral Architecture Functional Specification and Developer's Guide** is intended for developers of device drivers that will interface with file servers using NetWare 3.12 and above. This developer's guide explains general concepts, and it contains technical specifications, procedural outlines, and technical reference information on data structures and APIs used in the NWPA. The examples presented in this manual--such as pseudo code, function prototypes, and data structure definitions--are shown in ANSI C format; therefore, it is assumed that the reader is familiar with the syntax of ANSI C.

1.2 Presentation Format

The **NetWare Peripheral Architecture Functional Specification and Developer's Guide** combines a developer's tutorial and a technical reference into one concise manual. The major sections divide the manual into eight individual sections. Section 1 provides an introduction and overview of the NWPA. Within this section, all of the components of the NWPA are discussed, including a functional diagram showing, at a very high level, how the components interact.

Section 2 builds on the overview by taking a full chapter to discuss each major component in detail. The chapters that discuss the HACB and CDM Message data structures contain an overview of the respective structure's purpose, an ANSI C definition of the structure, and detailed field descriptions.

The chapters that discuss the HAM and CDM program modules contain both an architectural and an operational overview of the respective module. The architectural overview gives a brief description of all local data structures and routines (including their ANSI C prototypes) that need to be defined in the module. The purpose of the architectural overview is to provide the big-picture information that comprises the shell of the module. The purpose of the operational overview is to outline the major procedures in each module showing their functional flow and what calls need to occur in the flow. To prevent frequent detours that might detract from the main

flow, details--such as input arguments to routines and full descriptions of data structures--are not discussed in the overview. These details are provided in the module's corresponding reference section.

Sections 3 - 6 provide technical reference information for the following:

1. Data structures used in the NWPA. Each entry has an ANSI C definition of the structure followed by detailed field descriptions.
2. Routines that HAMs and CDMs are required to implement (*HAM_* and *CDM_* routines).
3. APIs provided by the NWPA (**CDI _** , **HAI_** , **NPA_**and **NPAB_** routines).

HACB Type Zero Functions (HAM Functions), which all HAMs are required to support, and the various types of CDM Messages that the Media Manager can issue to a CDM. Support routines provided by the NetWare OS. Sections 7 and 8 contain the appendices and index, respectively.

1.3 Synopsis of Sections and Chapters

The following list shows the organization of this manual according to section and chapter:

Section 1: Introduction

Chapter 1: About This Guide

This chapter contains a description of the material in this guide.

Chapter 2: Overview of the NetWare Peripheral Architecture

This chapter provides an overview to the NWPA that introduces its components along with a brief description of each. Also, this chapter describes some general OS issues pertaining to the NWPA.

Section 2: Functional Specification and Procedural Outline

Chapter 3: Host Adapter Control Block (HACB)

This chapter describes in detail the Host Adapter Control Block (HACB), which is the data structure containing I/O request information passed between a CDM and a HAM. This chapter also includes a detailed description of each field within the structure.

Chapter 4: Host Adapter Module (HAM)

This chapter defines the Host Adapter Module (HAM) which is the component in the NWPA responsible for providing an interface to the host adapter hardware. This chapter also outlines a HAM's functional flow.

Chapter 5: Custom Device Module (CDM)

This chapter describes in detail the Custom Device Module (CDM) which is the component in the NWPA responsible for translating application I/O requests into functions compatible with the devices attached to the host adapter hardware. This chapter also outlines a CDM's functional flow.

Section 3: NWPA Data Structures

Chapter 6: Technical Reference for NWPA Data Structures

This chapter lists all of the data structures used by a CDM and HAM, including detailed descriptions of their respective fields.

Section 4: NWPA Routines

Chapter 7: Technical Reference for NWPA Routines

This chapter provides functional descriptions of all of the local routines (*CDM_* and *HAM_* prefixes) that developers must implement in a CDM or HAM. Additionally, this chapter contains technical reference information for the APIs provided by the NWPA (**CDI_** , **HAI_**, **NPA_** and **NPAB_** prefixes).

Section 5: HAM Functions and CDM Messages

Chapter 8: HACB Type Zero Functions

This chapter describes the types of host adapter functions that may be issued by the Media Manager or an application. These functions request information about a HAM, a host adapter, or attached devices. A HAM is required to provide the functionality to implement these functions.

Chapter 9: CDM Message Types

This chapter describes the types of CDM Messages that a CDM can receive from the Media Manager. Additionally, this chapter lists Media Manager function and error codes that a CDM must recognize.

Section 6: OS Support Routines

Chapter 10: OS Support Routines

This chapter provides functional descriptions of applicable OS APIs. Some of these APIs are necessary in order to initialize modules, and some are auxiliary routines that merely simplify tasks.

Section 7: Appendices

Appendix A: Creating NetWare Loadable Modules

CDMs and HAMs are NetWare Loadable Modules (NLMs). This appendix provides supplementary information for creating NLMs.

Appendix B: Handling HACB Completion Errors

This appendix gives a more detailed description of the HACB Completion errors. Where applicable, it also suggests courses of action for both the CDM and HAM.

Appendix C: LDI/DDI Specification Supplements

This appendix describes how to write a DDI file for HAMs and CDMs.

1.4 Printing Conventions

Certain printing and style conventions are used in this manual to identify terms, concepts, routines, and their parameters. This section explains these conventions so that these terms can be easily identified along with their respective meanings when they are encountered in the text.

Underlined Bold

This style of type is used to identify titles of publications and words being emphasized in the text.

Italics

This style of type is used to identify words that act as placeholders. Generally, these placeholders are used in naming arbitrary parameters. The developers may name them whatever they choose.

HAM_Routine()

This type and prefix indicate a routine or function specific to the HAM. A HAM is required to implement local routines and functions to make the driver functional with the NWPA. These local routines and functions have specific tasks, and they are expected to interact with other NWPA components. For consistency in referring to these routines in the text and in code examples, this manual gives each a generic name in this type style with a *HAM_* prefix. The actual names developers choose for these routines are arbitrary, but their respective functionality must adhere to the specifications described in this manual.

CDM_Routine()
This type and prefix indicate a routine or function specific to the CDM. A CDM is required to implement local routines and functions to make the driver functional with the NWPA. These local routines and functions have specific tasks, and they are expected to interact with other NWPA components. For consistency in referring to these routines in the text and in code examples, this manual gives each a generic name in this type style with a *CDM_* prefix. The actual names developers choose for these routines are arbitrary, but their respective functionality must adhere to the specifications described in this manual.

CDI_ HAI_ NPA_ NPAB_

This type and these prefixes indicate routines from one of four different API sets provided by the NWPA. These API sets provide CDMs and HAMs with interfaces to the Media Manager.

`Code examples`
This font size and type will be used to indicate code examples throughout the document.

Field names and Structures
This font size and type will be used to indicate names of fields and structures.

"No prefix"

Routines without a prefix are actual NetWare Operating System routines. These are described in Chapter 10.