Appendix A Creating NetWare Loadable Modules

Device drivers must be converted from source modules into NetWare Loadable Modules before they can be loaded and installed in the NetWare v3.1x or v4.x Operating Systems. Novell Labs can provide a list of compilers which may be used to create object modules from driver source modules. The object modules must be processed by a NetWare linker, either NLMLINKP (which makes use of extended memory) or NLMLINKR (which uses only regular memory.) Both linkers require a developer-created definition file (.DEF) that provides configuration information needed to produce the NLM, including the NetWare internal variables and routines the driver will access once loaded. The NLMLINKP(R) command syntax is as follows:

nlmlinkp *drivername*<Enter>

Where *drivername* is the name of the definition file for the desired driver (It is not necessary to enter the .DEF extension in this command).

NLMLINKP(R) will find all the required information and directives in the definition file, including the names of all object modules which must be linked to form the driver (see Figure A-1 for a sample definition file). NLMLINKP(R) will produce an output file (with a .HAM or .CDM extension) which is the NLM form of the driver module, ready to load into an active NetWare environment. Figure A-1 shows a sample definition file which will direct NLMLINKP(R) to find sample.obj, link it, and produce the output file sample.ham.

Required Header Files

There are four header files that Novell has created to provide the necessary definition statements for HAMs and CDMs. These are:

- NPA.H - Required for HAMs and CDMs. This must be included first.

- NBI.H - Required for NBI aware HAMs. If used, this must be included before NPA_HAM.H. Also, for NBI aware HAMs, a Novell assigned compiler environment variable for the target platform must be declared. Currently, IAPX386 (for Intel platforms) is the only defined environment variable. NBI.H has an imbedded include statement that is associated with a platfom specific .H file based on this environment variable. This file contains all of the pragmas associated with the target platform for the specific compiler. Contact Novell Labs for a current list of supported platforms and their associated environment

variables and .H files. For IAPX386, the file NBIEXP.H must be located in your include path.

- NPA_HAM.H - Required for HAMs
- NPA_CDM.H - Required for CDMs

Please include these files in your modules and in the include path defined for your compiler.

```
.description  "Novell Sample HAM"
output $[f,"",$@,]
input $(OBJS)
type       8
start      HamMain
check      CheckHAM
exit       RemoveHAM
map
debug
reentrant
copyright (Copyright information goes here)
version    xx, yy, z (see keyword description for details)
import
    CCmpB
    CMovB
    CSetB
    CStrLen
    DisableAndRetFlags
    Enable
    HAI_Activate_Bus
    HAI_Complete_HACB
    HAI_Deactivate_Bus
    HAI_preProcess_HACB_Completion
    HAI_Abort_HACB
    InvertLong
    NPA_Add_Option
    NPA_Allocate_Memory
    NPA_Cancel_Thread
    NPA_Delay_Thread
    NPA_Exchange_Message
    NPA_Interrupt_Control
    NPA_Micro_Delay
    NPA_Parse_Options
    NPA_Register_HAM_Module
    NPA_Register_Options
    NPA_Return_Bus_Type
    NPA_Return_Memory
    NPA_Spawn_Thread
    NPA_System_Alert
    NPA_Unload_Module_Check
    NPA_Unregister_Module
    NPA_Unregister_Options
    NPAB_Get_Card_Config_Info
    OutputToScreen
    MapDataOffsetToAbsoluteAddress
    QueueSystemAlert
    SetFlags
#export
#   HamISR,
#   HamExecuteHacb,
#   HamAbortHacb,
```

Figure A-1 Sample Definition File

Definition File Keywords

The .DEF file keywords can occur in any order, and the required keywords are indicated. The following keywords are defined for use in the definition file to direct NLMLINKP(R) in creating NetWare v3.1x & v4.0 Loadable Module device drivers:

Type (Required) Specifies the type of loadable module as indicated below, and implicitly determines the extension to append on the output file.
1 = Lan Driver (.LAN)
2 = Disk driver (.DSK)
3 = Name space module (.NAM)
4 = Utility (.NLM)
8 = HAM
9 = CDM

Description (Required) Specifies a string that describes the loadable module to be created. The console command "MODULES" will output this string to describe this module. The description can be 1-127 bytes long, must be enclosed in double quotes and may not include a null, double quote, carriage return, or new-line. The description should contain the indicated fields in the following order format: "company or product name, description"

Output(Required) Specifies the name of the output file (the extension will be added by the linker as specified above).

Input(Required) Specifies the name of the input .OBJ file(s).

Start(Required) Specifies the name of the loadable module's initialization procedure. When the file server supervisor uses the "LOAD" console command to load the module, NetWare calls this procedure.

Exit(Required) Specifies the name of the loadable module's exit procedure. This procedure is called when the file server supervisor enters the "UNLOAD" console command.

Check(Required) Specifies the name of the loadable module's check procedure. The console command "UNLOAD" calls an NLM's check procedure (if it exists) before unloading the module. The check procedure is required for disk drivers since it must indicate to the OS if any disks are locked (module may not be safely unloaded).

Copyright(Required) Inserts the Novell default copyright or the copyright for a third-party developer. Usage: COPYRIGHT " " Novell default; COPYRIGHT "company" third party copyright message.

Version (Required) Usage: VERSION XX,YY,Z XX - Major Version number, YY - Minor Version number, Z - Revision number

Reentrant(Optional) Specifies that the loadable module is reentrant (i.e., two or more processes may

use the code at the same time). This keyword is mutually exclusive (both cannot be specified) with the keyword MULTIPLE.

Multiple(Optional) Specifies that more than one code image of the loadable module may be loaded into file server memory. This keyword is mutually exclusive (both cannot be specified) with the keyword REENTRANT.

Custom(Optional) Specifies that a custom data file is to be appended to the output file. This keyword should be followed by the file name of the custom data file.

Map(Optional) Directs the linker to create a map file.

Import(Required) Specifies that a list following the keyword will contain variable and procedure names that are external to the object files. These variables and procedures are case sensitive NWPA and/or NetWare v4.xx Operating System variables and procedures (or variables and procedures from other loadable modules which must have been previously loaded) which will be linked to the module after it has been loaded and before it begins initialization.

Export(Optional) Specifies that a list following the keyword will contains case sensitive variables and procedure names resident in the loadable module to be made available to other loadable modules.

Module(Optional) Specifies loadable modules that must be loaded before the current loadable module is loaded. The loader will attempt to find and load any modules not already in the server memory. If it cannot, the current module will not be loaded.

Debug(Optional) Specifies that the linker will include debug information in the output file.

@ operator(Optional) "@" is an operator that can be used with the Input, Import, and Export directives. The @ operator indicates that the list is to be read from a file, and can be nested. The file specifier, including path, must immediately follow the @ operator. Syntax: IMPORT @file.txt