

NIAKWA DATA MANAGER

INTEL UNIX/C-ISAM ADDENDUM



2nd. Edition - November 1993
COPYRIGHT © 1991-93 Niakwa, Inc.

Niakwa, Inc.
23600 N. Milwaukee Avenue
Mundelein, IL. 60060

PHONE: (708) 634-8700 FAX: (708) 634-8718 TELEX: 3719965 NIAK UB

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES AND PROPRIETARY RIGHTS

The staff of Niakwa, Inc. (Niakwa) has taken due care in preparing this manual. Nothing contained herein shall be construed to modify or alter in any way the standard terms and conditions of the Niakwa Programming Language Support and Distribution License Agreement, the End-User Support Only License Agreement, the Niakwa Software License Agreement and Warranty, or any other Niakwa License Agreement (collectively, the "License Agreements") by which this software package was acquired.

This manual is to serve as a guide for use of the Niakwa software only and not as a source of representations or additional undertakings by Niakwa. The licensee must refer to the License Agreements for Niakwa product and service representations.

No ownership of Niakwa software is transferred by any of the License Agreements. Any use of Niakwa software beyond the terms and conditions of the License Agreements, without the written authorization of Niakwa, is prohibited.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without prior written permission from Niakwa, Inc.

Niakwa is a registered trademark of Niakwa Management Services 1975 Ltd., and is licensed to Bluebird Systems.

Niakwa Programming Language (NPL), Bluebird and SuperDOS are registered trademarks of Bluebird Systems.

All other trademarks are the property of their respective holders.



PREFACE

P.1 NDM Intel UNIX/C-ISAM Addendum

The Niakwa Data Manager (NDM) allows Niakwa Programming Language (NPL) applications to use state-of-the-art native ISAM products to store data while retaining full portability. This section describes the organization of the NDM manuals, the required knowledge for using the NDM, and the notational conventions used to denote command codes and parameters that may be needed with this manual.

P.1.1 How to Use this Manual

The NDM documentation consists of two manuals, the Programmer's Guide and the Platform-Specific Addendum. This Platform-Specific Addendum provides machine and C-ISAM-specific information as it relates to the NDM functions and installation. This manual also documents all C-ISAM-specific configuration requirements, extended NDM features supported, and NDM limitations under C-ISAM.

This should be used with the NDM Programmer's Guide which is intended as a hardware-independent guide for programmers in the correct use of the NDM functions, NDM program development, NPL program conversions, and the NDM Utilities.

This platform-specific addendum is organized by chapters.

Chapter 1 discusses configuring, installing, and starting NDM on Intel UNIX based platforms.

Chapter 2 discusses integrating external routines with the NDM API.

Chapter 3 discusses C-ISAM's limitations and supported toolbox features as they affect NDM.

Chapter 4 discusses some considerations when installing NDM for the end-user.

P.1.2 Prerequisite Knowledge

This Platform-Specific Addendum assumes that the user has at least a basic knowledge of the computer system in use, the native operating system, and the Niakwa Programming Language, and that C-ISAM is installed and operating correctly. Refer to the appropriate systems manuals for more information regarding specific hardware, operating system, NPL, and C-ISAM.

P.1.3 Notational Conventions

The NDM Programmer's Guide and Platform-Specific Addenda use the following notational conventions.

NOTE: Notes provide information of particular importance.



WARNING--Warnings are conditions that require special care by the user. Disregarding this information could result in a serious problem.

HINT: Hints provide helpful comments pertaining to the use of particular features.

For documentation purposes, examples in this manual use both the number and name of the API functions.

Under NPL Release III, only API function numbers can be used when issuing an NDM call. Under NPL Release IV, the API function number or name can be used (but not both at the same time).

In addition, long variable names used for the API parameters can be used under NPL Release IV only. Standard NPL variable names should be used under NPL Release III.

All alpha parameters used in API function descriptions end in \$, while numeric parameters do not.

For example:

```
'31060 NDM_CREATE_FILE (FILE_NAME$, MODE$, INDEX_NUMBER, KEYDESC_TABLE$, RECORD_LENGTH, ISAM_SPECIFIC$, FILE_HANDLE$, RETURN_CODE)
```

would be coded as:

```
GOSUB' 31060 ("EXAMPLE", M$, I, K$, L, A$, H$, R)
```

under Release III, but could be coded as

```
GOSUB'NDM_CREATE_FILE ("EXAMPLE", SHARE_MODE$, INDEX_NUM, KEY_TABLE$, REC_LENGTH$, ISAM_CODE$, FILE_HANDLE$, RETURN_CODE)
```

under Release IV.

TABLE OF CONTENTS

PREFACE

NDM Intel UNIX/C-ISAM Addendum	P-1
How to Use this Manual.....	P-1
Prerequisite Knowledge	P-2
Notational Conventions.....	P-2

INSTALLING NDM

Overview.....	1-1
System Requirements	1-2
Memory	1-2
C-ISAM Requirements.....	1-3
UNIX Requirements.....	1-3
NPL Requirements	1-3
System Setup.....	1-4
.profile.....	1-4
NDM Environment Variable.....	1-4
Number of Open Files	1-5
C-ISAM Installation/Sartup Options.....	1-5
NDM Software Installation	1-5
Diskette Contents.....	1-5
Installing the NDM Software.....	1-7
NPL Security Configuration Using the seci.ndm Script.....	1-8
Starting NDM	1-9

CREATING THE NDM EXECUTABLES

Overview.....	2-1
Contents of NDM UNIX BESDK	2-2
Installing NDM UNIX BESDK.....	2-3
Creating Linked rtic and rtpc Files.....	2-4
The Make Procedure	2-5
Creating the Linked Files when BESDK or C-ISAM is Installed in Another Location.....	2-6
Merging the External Libraries	2-6

Creating the Linked rticx and rtpcx with other External Routines.....	2-8
Using Non-C External Routines	2-8
Using the Merged Library	2-9

C-ISAM CONSIDERATIONS

Overview.....	3-1
C-ISAM Field Classes Supported	3-2
C-ISAM File Limits.....	3-3
C-ISAM-Specific Considerations.....	3-3
Toolbox Features	3-4
Key Types.....	3-5
File Limits.....	3-5
Transaction Processing.....	3-5
Create/Delete Index.....	3-5
C-ISAM-Specific Error Messages.....	3-5
Intel UNIX Features of the NDM Utilities.....	3-6
Ordering/Updating C-ISAM.....	3-6

END-USER CONSIDERATIONS

Overview.....	4-1
Minimum NDM File Requirements	4-2
NDM Error Description File.....	4-2
Providing the NDM Utilities to End-Users	4-2
Data Dictionary Files.....	4-4
Catalog File.....	4-4
Field Type File.....	4-5



CHAPTER 1

INSTALLING NDM

1.1 Overview

This chapter discusses configuring, installing, and starting NDM on Intel UNIX-based platforms.

Section 1.2 discusses configuration requirements.

Section 1.3 discusses system files that may need to be modified.

Section 1.4 discusses C-ISAM installation and startup options.

Section 1.5 discusses NDM software installation and file locations.

Section 1.6 discusses starting NDM.

1.2 System Requirements

The following table illustrates the basic hardware configurations for Intel UNIX systems using NDM software with Informix's C-ISAM.

Operating System	C-ISAM
AT&T UNIX	C-ISAM for AT&T UNIX Ver. 4.00 or greater.
INTERACTIVE UNIX	C-ISAM for INTERACTIVE UNIX Ver. 4.00 or greater.
SCO UNIX	C-ISAM for SCO UNIX Ver. 4.00 or greater.
ALTOS UNIX	C-ISAM for SCO UNIX Ver. 4.00 or greater.

Refer to the *NiakwaNews* for current information on Niakwa NPL-compatible computer systems.

The following is a list of the requirements necessary for NDM.

1.2.1 Memory

It is necessary to link the NDM and C-ISAM libraries with the NPL UNIX RunTime executables to create a combined NDM and NPL executable (refer to Chapter 2 for details). The memory required for the linked executable includes the size of the NPL UNIX RunTime being used (i.e., Release III or IV rti or rtp) plus the size of the NDM UNIX and NDM and the C-ISAM libraries.

The following table shows the memory required by the different components that make up the linked NPL/NDM executable.

Name of File	NPL Release III	NPL Release IV
rti	256K	330K
rtp	165K	212K
NDM and C-ISAM libraries	296K	296K

NOTE: If developer-created external routines are used with the NDM for C-ISAM, the size of the linked executable `rtic` and `rtpc` files may vary. Refer to Chapter 2 of this addendum for more information.

Refer to the C-ISAM manuals for the memory requirements for C-ISAM.

HINT: NDM can be configured to allocate additional memory for internal use. Refer to the NDM Programmer's Guide, Chapter 10, function **31320 NDM_INITIALIZE** for details.

1.2.2 C-ISAM Requirements

Refer to the C-ISAM Installation Guide for information on installing the C-ISAM program files.

C-ISAM uses internal buffers to speed disk I/O. The default value is 16 buffers. This can be changed by setting the environment variable `ISAMBUFS`. It is recommended to use four buffers for each index file to be used simultaneously.

For example, if 30 buffers are needed, enter the following at a system prompt before executing `rtic` or `rtpc`:

```
ISAMBUFS=30
export ISAMBUFS
```

1.2.3 UNIX Requirements

All versions of Intel UNIX currently supported by NPL are compatible with the NDM.

1.2.4 NPL Requirements

NDM is supported only on NPL Revision 3.20 or greater.

1.3 System Setup

For proper operation of NDM, modifications must be made to the user's .profile file.

NOTE: For information on the .profile file, refer to the UNIX system manuals.

1.3.1 .profile

Each user's .profile file must set the directory where the NDM files are stored as an alternate directory. For example, if the NDM files are placed in the /usr/ndm directory, the following must be added to the PATH statement in the user's .profile:

```
:/usr/ndm:.
```

1.3.2 NDM Environment Variable

NDM allows the user the freedom to store NDM files in any subdirectory on the host system's hard drive. This location is then passed to NDM by means of the NDM environment variable. Consequently, the environment variable must be set before executing NDM.

NOTE: This variable must be set before executing NDM, and it is strongly recommended that it be set in the user's .profile file.

The NDM environment variable is set by the UNIX "export" command. This may be entered at a system prompt or added to the user's .profile file.

For example, if NDM files are located in the /usr/ndm directory, enter:

```
NDM=/usr/ndm
export NDM
```

To verify that the NDM environment variable is set, enter the UNIX "set" command at a system prompt and review the output for the appropriate set statement.

NOTE: If the NDM environment variable is not set NDM defaults to the `/usr/ndm` directory.

1.3.3 Number of Open Files

It may be necessary to increase the number of open files allowed to each user by the Intel UNIX operating system when using the Intel UNIX implementation of NDM. Refer to the Intel UNIX operating system manuals for details on increasing the number of open files for each user.

1.4 C-ISAM Installation/Sartup Options

Refer to the C-ISAM documentation regarding platform-specific installation and setup information for C-ISAM.

1.5 NDM Software Installation

This section discusses the contents and installation of the NDM Software for Intel UNIX.

1.5.1 Diskette Contents

Each NDM data file consists of two separate files, one with the extension `.dat` and the other with the extension `.idx`.

NOTE: The extension is in lower case and the file name is in uppercase (i.e., `FILENAME.ext`).

Below is a list of the file names of NDM data files, without extensions, contained on the NDM Development Package Diskette.

CUSTCAT	Catalog file for the sample program.
CUSTDD	Data description file for the sample program.
CUSTKD	Key description file for the sample program.

CUSTREL	Relation file for the sample program.
CUSTEXAM	Data file for the simple example programs.
NDMCAT	Catalog file.
NDMCATD	Data description file for the catalog file.
NDMCATK	Key description file for the catalog file.
NDMDDDD	Data description file for data description files.
NDMDDKD	Key description file for data description files.
NDMERR	Error message file.
NDMERRD	Data description file for the error message file.
NDMERRK	Key description file for the error message file.
NDMFTF	Field type file.
NDMFTFD	Data description file for the field type file.
NDMFTFK	Key description file for the field type file.
NDMKDDD	Data description file for key description files.
NDMKDKD	Key description file for key description files.
NDMREL	Relation file.
NDMRELD	Data description file for the relation file.
NDMRELK	Key description file for the relation file.
NDMTRAN	Translation file.
NDMTRAND	Data description file for the translation file.

NDMTRANK Key description file for the translation file.

Below is a list of the non-data files on the NDM Development diskette:

ndmutil	Shell script to start the NDM Utilities.
NDMUTIL.BS2	The NDM Utilities diskimage.
NDM.HLP	NDM help file. Contains descriptions on the NDM function calls.
NDM.IDX	Index file for NDM.HLP
seci.ndm	Installation shell script for when NPL security problem is encountered.
EX1NDM.OBJ	Simple example program number 1.
EX2NDM.OBJ	Simple example program number 2.
NDMUTIL.OBJ	Boot program used when starting the NDM Utilities.
TESTCUST.OBJ	Boot program for the sample program.

1.5.2 Installing the NDM Software

To install the NDM Development Package diskette, follow the steps below:

NOTE: These instructions assume that drive 0 (fd0) is the diskette drive to be used. If the drive is different on the system being used, be sure to use the proper device designation instead.

1. Login to UNIX as "root"
2. Enter the following UNIX commands:

```
cd /
umask 000
ulimit 65000
```

3. For 5-1/4" 1.2 MB diskettes enter:

```
tar xvfbk /dev/fd096ds15 20 1200
```

For 3-1/2" 720K diskettes enter:

```
tar xvfbk /dev/fd0135ds9 20 720
```

All files on the diskette are automatically copied into the /usr/ndm directory. Once installed, these files can be moved to any directory, provided the NDM environment variable is used. Refer to Section 1.3.2 for more information.

The NPL RS/6000 BESDK libraries must be linked to C-ISAM by the developers with the use of the NDM UNIX BESDK. Refer to Section 2.4 for details on creating the NPL UNIX RunTime/NDM executables.

NOTE: The rti and rtpc files, created by developers with the use of the NDM UNIX BESDK, can be moved to either the directory which contains the NDM software or any directory which the developer chooses, provided the NIAKWA_RUNTIME and NDM environment variables are used. Refer to Section 1.3.2, the Revision 3.20 UNIX Addendum, or the Release III or Release IV NPL UNIX/Xenix Supplement for more information.

1.5.3 NPL Security Configuration Using the seci.ndm Script

Some security problems may appear related to where the rti and rtpc files were made and where they will be used (refer to Section 2.4 for more information). These problems are due to hardware incompatibilities between different UNIX systems and generate a 1704 security error code when attempting to execute the linked Runtimes. If this occurs, the error can be avoided by recalling the present NPL UNIX RunTime and reinstalling it using the linked rti and rtpc files. Niakwa has provided a shell script "seci.ndm" that automatically performs this procedure.

To execute the seci.ndm script, make sure the NDM has been successfully installed as described in Section 1.4.2. Switch to the directory where the NDM files are stored and enter:

```
seci.ndm
```

This script file performs the following:

1. Recalls the NPL UNIX RunTime installed security.

NOTE: The NPL RunTime Gold Key diskette is necessary to complete this step.

2. Renames rtp to rtp.old and rti to rti.old respectively.

3. Copies rtpc and rtic into the directory pointed to by the NIAKWA_RUNTIME environment variable and renames them rtp and rti.
4. Reinstalls the NPL RunTime Security using the new rtp and rti files.

After running this script, the linked RunTimes execute correctly, but attempts to execute rti.old and rtp.old result in a 1704 security error and the NPL Gold Key diskette will be necessary to pass security.

NOTE: Niakwa advises that only the linked RunTimes be used once this script is executed, to avoid confusing users as to which NPL Runtime to use at what time.

1.6 Starting NDM

Before NDM can be executed, NPL must be fully operational on the UNIX system in use and the NDM environment variable must be set to point to the directory where the NDM files are located (refer to Section 1.3.2).

The general form for starting NDM is:

```
rtic <BOOTNAME>
```

or

```
rtpc <BOOTNAME>
```

Other NPL start-up options may also be specified. For example, if the -d option is to be used, the start-up statement is:

```
rtic -d=20 <BOOTNAME>
```

NOTE: The NDM for Intel UNIX executables (rtic and rtpc), created by use of the method described in Section 2.4, should be placed in the directory that is specified by the NDM environment variable. Refer to Section 1.3.2 for information on setting the NDM environment variable.



CHAPTER 2

CREATING THE NDM EXECUTABLES

2.1 Overview

The NDM UNIX BESDK diskette supplies the necessary object files, source code and example make procedures to allow developers to link the NDM API and the C-ISAM libraries to the NPLUNIX BESDK libraries to create the necessary executables `rtic` and `rtpc`. The NDM UNIX BESDK can also be used to integrate developers' own external routines with the RunTime once it has been linked to the NDM API and C-ISAM libraries, to create the `rticx` and `rtpcx` executables.

For the purpose of definition, `rti` and `rtp` refer to the NPL UNIX RunTime executables, `rtic` and `rtpc` refer to the combined NPL UNIX RunTime, NDM API, and C-ISAM libraries executables, and `rticx` and `rtpcx` refer to the combined NPL UNIX RunTime, NDM API, C-ISAM libraries and the developers' own external libraries executables.

NOTE: The linking of NPL UNIX RunTime, NDM API and the C-ISAM libraries must be done by the developer to create the RunTime executables (rtic and rtpc). Niakwa is currently prohibited from distributing prelinked rtic and rtpc files due to the way Informix licenses C-ISAM. Consequently, it is the developers' responsibility to license the correct version of the C-ISAM runtime and link the C-ISAM libraries to the NPL UNIX RunTimes. Once linked, the rtic and rtpc files can be copied to each end user system where an NPL UNIX RunTime is properly installed and where C-ISAM is properly licensed.



WARNING--Informix requires that a valid C-ISAM run time be purchased for each system running C-ISAM. Compliance with this requirement rests solely with the developer.

The make file, included with the NDM UNIX BESDK creates the files rtic and rtpc. If developers already have files using these names or must use the NDM UNIX BESDK to integrate their own external routines to create rticx and rtpcx, it is recommended that they be renamed (using the mv command) before the NDM UNIX BESDK is executed.

Section 2.2 discusses the content of the NDM UNIX BESDK diskette.

Section 2.3 discusses installing the NDM UNIX BESDK.

Section 2.4 discusses creating the necessary rtic and rtpc files from the NPL UNIX RunTime, NDM API and the C-ISAM libraries.

Section 2.5 discusses creating customized rticx and rtpcx files with merged external routines.

Section 2.6 discusses using non-C external routines.

Section 2.7 discusses using the merged library.

2.2 Contents of NDM UNIX BESDK

The NDM UNIX BESDK diskette contains two directories, /ndmx and /ndmexam.

The /ndmx directory contains the files that, when linked with the NPL code, create the customized version of the NPL UNIX RunTime, with C-ISAM libraries and NDM API linked in. This directory contains the following files:

makefile	Commands to create rtic/rtpc.
ndmcisam.lib	NDM routines in compiled form.
ndm_rtp.h	Prototypes for routines used by RTPEXT.
rtp_ext.c	RTPEXT routine.
rtp_main.c	Mainline.

The /ndmexam directory contains the same files merged with the example external subroutine from the NPL UNIX BESDK package, plus the files necessary to compile and link the example. This is intended as an example for developers to merge in their own external routines. Refer to Chapter 10 of the Release III Xenix/UNIX Supplement or Chapter 11 of the Release IV NPL Intel UNIX Supplement for information on the NPL external calls.

The following files are found in the /ndmexam directory:

makefile	Commands to create merged rtic/rtpc
rtp_ext.c	Merged RTPEXT routine
rtp_main.c	Merged mainline

2.3 Installing NDM UNIX BESDK

The /ndmx and /ndmexam directories can be installed at any directory level on the system.

NOTE: The makefile file explicitly points to the /usr/BASIC2C/uextrn directory. If the /usr/BASIC2C/uextrn directory is moved to another location, the makefile must be modified for compiling and linking to succeed.

In the following example, the uextrn directory of the NPL UNIX BESDK is assumed to be in the /usr/BASIC2C directory (the full path name of this directory is /usr/BASIC2C/uextrn under UNIX. It is also assumed that the NDM UNIX BESDK diskette is in drive 0 on the UNIX system and that the NDM UNIX BESDK is to be installed into the /usr/ndm directory. If this is not the case, substitute the correct names.

The steps shown below install the NDM UNIX BESDK.

1. Select the /usr/ndm directory by entering:

```
cd /usr/ndm
```

2. Insert the NDM BESDK diskette into drive 0.

3. For 5 1/4", 1.2 MB diskettes enter:

```
tar xvfbk /dev/fd096ds15 20 1200
```

For 3 1/2", 720K diskettes enter:

```
tar xvfbk /dev/fd0135ds9 20 720
```

2.4 Creating Linked rtic and rtpc Files

This section discusses the creation of the NPL UNIX RunTime executables rtic and rtpc. These files are the linked versions of the NPL UNIX BESDK libraries and C-ISAM.

NOTE: This procedure must be done by the developer to create the NPL RunTime executables rtic and rtpc that contain the NDM API and C-ISAM libraries. Niakwa is currently prohibited from distributing prelinked rtic and rtpc files due to the way Informix licenses C-ISAM. Consequently, it is the developers' responsibility to license the necessary C-ISAM runtime and link the C-ISAM libraries to the NPL UNIX RunTimes. Once linked, the rtic and rtpc files can be copied to each end-user system, where an NPL UNIX RunTime is properly installed and where C-ISAM is properly licensed.



WARNING -- Informix requires that a valid C-ISAM runtime be purchased for each system running C-ISAM. Compliance with this requirement rests solely with the developer.

If the developer is intending on supporting NDM on different flavors of Intel UNIX supported by Niakwa, the following table must be considered when the linked rtic and rtpc files are to be created under one flavor of UNIX and installed under another.

Run Under/ When Made On	SCO UNIX	AT&T UNIX	INTERACTIVE UNIX	ALTOS UNIX
SCO UNIX	OK	OK (1)	OK	OK (1)
AT&T UNIX	Incompatible	OK	Incompatible	Incompatible
INTERACTIVE UNIX	OK	OK (1)	OK	OK (1)

- Ok Linked RunTime files made on a specific flavor of UNIX will execute under a different flavor. For example, a linked NPL/NDM RunTime made under SCO UNIX would execute correctly under INTERACTIVE UNIX.
- Incompatible Linked NPL RunTime files made on a specific flavor of UNIX will not execute under a different flavor. For example, a linked NPL/NDM RunTime made under SCO UNIX will not execute under AT&T UNIX.
- 1 NPL security problems may be encountered due to hardware incompatibility between the operating system where the linked NPL RunTimes were created and where they are to be executed. Refer to Section 1.4.3 for details on how to correct this situation.

2.4.1 The Make Procedure

Before creating the linked rtic and rtpc, the NPL UNIX BESDK, the NDM UNIX BESDK, and the proper C-ISAM library files must be installed on the system. Refer to the Release III or Release IV NPL Intel UNIX Supplements for information on installing NPL and Section 1.5 and Section 2.3 for details on installing the NDM and the NDM UNIX BESDK. Refer to the C-ISAM documentation for information on installing the C-ISAM libraries.

The following steps create the linked versions of `rtic` and `rtpc`:

1. Select the `/usr/ndm/ndmx` directory by entering:

```
cd /usr/ndm/ndmx
```

2. Enter the following command to create the linked version of `rtic` and `rtpc`:

```
make
```

2.4.2 Creating the Linked Files when BESDK or C-ISAM is Installed in Another Location

If the NPL UNIX BESDK directories, the NDM UNIX BESDK directory, or the C-ISAM libraries are not in the standard locations, the makefile in the `ndmx` and `ndmexam` directories must be changed so that the files will be found.

At the start of the makefile in the `ndmx` directory, change the following lines:

```
BDIR=/usr/BASIC2C/uextrn  
CDIR=/usr
```

At the start of the makefile in the `ndmexam` directory change the following lines:

```
BDIR=/usr/BASIC2C/uextrn  
CDIR=/usr  
NDIR=../ndmx
```

where:

BDIR	The name of NPL UNIX BESDK directory.
CDIR	The name of the directory containing the C-ISAM library and include directories.
NDIR	The name of the directory containing the <code>ndmx</code> files on the host system.

2.5 Merging the External Libraries

If a developer has an NPL application that contains external routines as well the NDM external routines, the developer must also link the external libraries to create the `rticx` and `rtpcx` executables.

To merge two external libraries, follow these steps:

1. Copy the files in the NDM UNIX BESDK directory (i.e. the ndmx directory) to the directory in which the merged library is to be created.
2. Modify the rtp_ext.c as follows:
 - Add the C declarations of the other library's routines at the start of the file after the #include statements. The C declarations for the NDM routines are in a file named ndm_rtp.h.
 - Add the NPL definitions for the other library's routines to the beginning of the switch statement. The rtpdef_next_number field of the last definition that is added must contain the DEFFN' number of the first NDM DEFFN'(31010).
 - Modify the rtpdef_next_number field in the default section of the switch statements to be the lowest DEFFN' number of the NPL definitions being added.

For an example of these changes, review the rtp_ext.c file in the /ndmexam directory.

3. Modify the rtp_main.c file as follows:
 - Add the initialization and termination routines for the other library to the start of the file.
 - Add a call to the other initialization routine immediately after the call to ndm_init().
 - Add a call to the other termination routine immediately before the call to ndm_done().

For an example of these changes, review the rtp-main.c file in the /ndmexam directory.

4. Modify the makefile file by adding the statements necessary to compile the file(s) containing the other external routines.
5. Run the make file to compile the necessary files and link them together to create an rtic or rtpc containing the merged external libraries. Refer to Section 2.5.1 for details.

NOTE: The NPL external routine definitions in `rtp_ext.c` are actually calls to a routine (in the same file) called `set_sub()`. The parameters to this routine are the values to be placed into the `rtpdef` structure. The `set_sub()` routine copies them into this structure. This routine is used to make the code size of the external library slightly smaller than if this copying is repeated for each case within the switch statement.

2.5.1 Creating the Linked `rticx` and `rtpcx` with other External Routines

Before creating the linked `rticx` and `rtpcx` executables, the NPL UNIX BESDK and the NDM UNIX BESDK files must have been installed on the system. Refer to the Release III or IV Intel UNIX Supplements for details on installing NPL, Section 1.5 and Section 2.3 for details on installing NDM and the NDM UNIX BESDK.

It is also necessary to make the modification described in Section 2.5 before attempting to create the linked `rticx` and `rtpcx`.

NOTE: The NDM UNIX BESDK creates the files `rticx` and `rtpcx`. If the developer already has files using these names, it is recommended that those files be renamed (using the `mv` command) before the NDM UNIX BESDK is executed.

Follow the steps shown below to create the linked versions of `rticx` and `rtpcx`:

1. Select the `/usr/ndm/ndmexam` directory. Enter:

```
cd /usr/ndm/ndmexam
```

2. Enter the following command to create the linked version of `rticx` and `rtpcx` and the files `mymain` and `MYBOOT.OBJ`:

```
make
```

NOTE: If the NPL BESDK or C-ISAM libraries are installed in non-standard locations, the makefile will require further modifications. Refer to Section 2.4.2 for details.

2.6 Using Non-C External Routines

If the other external routine definitions are written in Pascal or assembly language, it is necessary to put underscores ("`_`") at the beginning of each routine name.

2.7 Using the Merged Library

The merged `rtic`, `rtpc`, `rticx`, or `rtpcx` executable files produced by the above steps contain the NPL UNIX RunTime, C-ISAM libraries, the NDM API and the user external routines (if needed).

To install these linked executables on any system, use the UNIX "tar" command to copy the `rtic`, `rtpc`, `rticx`, or `rtpcx` files to the NDM directory of a system where the NPL UNIX RunTime has been properly installed. To avoid name confusion, rename the `rti` or `rtp` files to some other name. Regardless of the name used, the merged library automatically uses the serial number and security information of the NPL UNIX RunTime installed on the system.

NOTE: An NPL UNIX RunTime must be properly licensed and installed on every system where the merged library is to be used.

If the developer is intending on supporting NDM on different flavors of Intel UNIX supported by NIAKWA, refer to the table in Section 2.4 for considerations to follow when the linked `rtic` and `rtpc` files are created under one flavor or UNIX and installed under another.



CHAPTER 3

C-ISAM CONSIDERATIONS

3.1 Overview

This chapter provides information on features and limitations of NDM that are specific to C-ISAM.

Section 3.2 lists the native C-ISAM field types that are supported.

Section 3.3 provides a list of C-ISAM file limits.

Section 3.4 discusses all known C-ISAM-specific considerations for use with the NDM.

Section 3.5 discusses the various NDM toolbox features that Niakwa supports under C-ISAM.

Section 3.6 discusses C-ISAM-specific error messages.

Section 3.7 discusses Intel UNIX-specific features of the NDM Utilities.

Section 3.8 discusses ordering/updating C-ISAM.

3.2 C-ISAM Field Classes Supported

The key types currently supported are shown in the table below:

Native Field Type Number	Field Class	Native Field Type	Min Size	Max Size	Size Increment
1	A	String	1	32511	1
2	N	Integer	2	2	0
3	N	Float	4	8	4
6	D	Date	4	4	0
7	A	Time	4	4	0
11	N	Long	4	4	0
13	D	Alpha Date	8	8	0
14	A	Alpha Time	8	8	0
15	N	Decimal (C-ISAM)	2	17	1
16	N	No Conversion	1	32511	1

The field class types in the above table are defined as:

A = Alphanumeric
 N = Numeric
 D = Date

The above chart lists all the field types supported as keys by NDM under C-ISAM. Other NDM field types supported on other NDM platforms may also be used as field types. These other field types may not be used as keys.

NOTE: Field Type 16, No conversion, does no conversion from the NPL field type to the C-ISAM field type. Use of this field type is only for applications that need to be ported to platforms which support other string types (i.e. EBCDIC, on the IBM AS/400). This field type is useful for overlay fields or alpha fields that need to be stored in ASCII format.

When porting NDM files (catalog, data dictionary, key description, and data files), to C-ISAM from other NDM platforms, care should be taken to make sure that field types used as keys on another NDM platforms are supported by NDM under C-ISAM. For example, a Btrieve packed decimal field (if used as a key field) must be converted to a C-ISAM decimal field (type 15) when porting from Btrieve to C-ISAM. Refer to Sections 4.13 - 4.14 of the NDM Programmer's Guide for more information on converting native ISAM files from one native ISAM to another.

For further information on the structure of these field types, refer to the C-ISAM documentation.

3.3 C-ISAM File Limits

The file limits for C-ISAM with respect to NDM are defined in the table below.

Description	NDM	C-ISAM
Record Length	4090	32767
Key Length	120	120
No. of keys/file	9	no limit
No. of segs/key	8	8

3.4 C-ISAM-Specific Considerations

The following is a list of C-ISAM-specific issues that must be considered when working with NDM under C-ISAM:

3.5.1 Key Types

NDM key segments can normally be: string or integer. When the key types toolbox feature is enabled, all other field types supported by C-ISAM may be used as key segments. These types are listed in the table in Section 3.2.

3.5.2 File Limits

Using this toolbox feature allows NDM files to use the C-ISAM limits defined in Section 3.3.

3.5.3 Transaction Processing

This feature allows access to the 31430 NDM_TRANSACTION_ABORT, 31440 NDM_TRANSACTION_COMPLETE, and 31450 NDM_TRANSACTION_START NDM API calls. These are used to logically group a series of file operations to treat the series of operations as a single, indivisible operation.

3.5.4 Create/Delete Index

This feature allows access to the 31080 NDM_CREATE_INDEX and 31110 NDM_DELETE_INDEX NDM API calls, which are used to create temporary indices on existing data files.

3.6 C-ISAM-Specific Error Messages

NDM returns both an NDM and Native ISAM return code (error code) for each NDM API call (chapter 11 of the NDM Programmer's Guide discusses all generic NDM return codes). The NDM error codes specific to the Intel UNIX/C-ISAM implementation of NDM can be displayed by using the NDM Utilities and interpreted by referring to the C-ISAM documentation.

3.7 Intel UNIX Features of the NDM Utilities

The following section discusses Intel UNIX specific requirements for the use of the NDM Utilities.

All C-ISAM-specific features of the NDM Utilities are already discussed in Section 4.12 of the NDM Programmer's Guide and Section 2.2 of the NDM Revision 1.20 Addendum.

3.8 Ordering/Updating C-ISAM

The C-ISAM runtime software from Informix is not included with NDM software, but is required for use of NDM. To locate a vendor that sells C-ISAM in your area, you may contact Informix at (800) 274-8184 or, for an authorized dealer, Midwest Software at (708) 392-0091.

Niakwa has attempted to simplify the confusing process of determining which C-ISAM product to order by enclosing the following information needed to order C-ISAM from an Informix Dealer:

- The equipment manufacturer/Model number of the host system (e.g., Compaq DESKPRO 486/25 Model 120)
- The Operating System/Version number (e.g., SCO UNIX 3.2.4)
- The number of users (e.g., 1-32)
- The media size (e.g., 3-1/2" 1.44 MB diskettes)

NOTE: The C-ISAM run time software, unlike Btrieve from Novell which has an unlimited distribution license, must be licensed for each system NDM will be operating on.

Development sites are strongly urged to purchase a copy of the C-ISAM Development Package for an internal development reference. The C-ISAM Development Package contains the C-ISAM run Time software (required for each system running the NDM for Intel UNIX using C-ISAM), plus the C-ISAM Programmer's Manual and documentation.



CHAPTER 4

END-USER CONSIDERATIONS

4.1 Overview

This chapter provides information on the setup of NDM on end-user systems.

Section 4.2 discusses the minimum file requirements for the end-user system.

Section 4.3 discusses the minimum file requirements for the NDM error description file.

Section 4.4 discusses providing the NDM Utilities to end-users.

Section 4.5 discusses the data dictionary files.

Section 4.6 discusses the catalog files.

Section 4.7 discusses the field type file.

4.2 Minimum NDM File Requirements

The NDM executable files (rtic and rtpc) must be created by the developer using the method described in Section 2.4. Once created, these files must be copied into the NDM directory on the end-user's system. No additional files are required.



WARNING -- It is the responsibility of the developer to properly license C-ISAM for use on each end-user system.

There are several conditions that require the developer to provide additional files from the NDM Development Package to the end-user's system. The following sections discuss these conditions and which NDM Development Package files are required.

4.3 NDM Error Description File

If the application uses the API function call 31210 NDM_GET_ERROR_DESCRIPTION, the following files must be provided to the end-user:

NDMERR.dat	List of basic NDM error messages and any customized errors.
NDMERR.idx	List of basic NDM error messages and any customized errors.

4.4 Providing the NDM Utilities to End-Users

If the developer wishes to install the NDM Utilities on the end-user's system, the following files must be provided:

NOTE: Each NDM data file consists of two separate files, one with the extension .dat and the other with the extension .idx. The filenames listed below do not show their extensions. On end-user systems, both the .idx and .dat files for each filename shown must be included.

NDMCAT	Catalog file.
NDMCATD	Data description file for the catalog file.
NDMCATK	Key description file for the catalog file.
NDMDDDD	Data description file for data description files.
NDMDDKD	Key description file for data description files.
NDMERR	Error message file.
NDMERRD	Data description file for the error message file.
NDMERRK	Key description file for the error message file.
NDMFTF	Field type file.
NDMFTFD	Data description file for the field type file.
NDMFTFK	Key description file for the field type file.
NDMKDDD	Data description file for key description files.
NDMKDKD	Key description file for key description files.
NDMREL	Relation file.
NDMRELD	Data description file for the relation file.
NDMRELK	Key description file for the relation file.

The following files must also be provided:

ndmutil	Program to start the NDM Utilities.
---------	-------------------------------------

NDMUTIL.BS2	The NDM Utilities diskimage.
NDM.HLP	NDM Help file. Contains descriptions of the NDM functions calls.
NDM.IDX	Index for NDM.HLP file.
NDMUTIL.OBJ	Boot program used when starting the NDM Utilities.

4.5 Data Dictionary Files

If the application's data dictionary files are accessed under program control, then the following files must be installed on the end-user's system.

NOTE: Each NDM data file consists of two separate files, one with the extension .dat and the other with the extension .idx. The filenames listed below do not show their extensions. On end user systems both the .idx and .dat files for each filename shown must be included.

NDMDDDD	Data description file for data dictionary files.
NDMDDKD	Key description file for data dictionary file.
NDMKDDD	Data description file for key description files.
NDMKDKD	Key description file for key description files.

However, if the data dictionary files are not accessed directly, then the end user does not need these files.

4.6 Catalog File

If the application builds catalogs under program control, then the end-user must have the following files:

NOTE: Each NDM data file consists of two separate files, one with the extension `.dat` and the other with the extension `.idx`. The filenames listed below do show their extensions. On end user systems both the `.idx` and `.dat` files for each filename shown must be included.

NDMCATD Data description file for the catalog file.

NDMCATK Key description file for the catalog file.

4.7 Field Type File

If the application uses 31200 `NDM_GET_DEFAULT_FIELD_TYPE`, then the following file is needed:

NOTE: Each NDM data file consists of two separate files, one with the extension `.dat` and the other with the extension `.idx`. The filename listed below does not show its extension. On end-user systems, both the `.idx` and `.dat` files for the filename shown must be included.

NDMFTF Field type file