NIAKWA DEVELOPMENT TOOLS

DATA MANAGER

IBM AS/400
USING OS/400

PLATFORM-SPECIFIC
ADDENDUM

1st Edition - October 1993

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 (NPL) Support and Distribution License Agreement, the End-User Support Only License Agreement, the Niakwa Software License Agreement and Warranty and 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.

PREFACE

P.1 The IBM AS/400 Platform Specific Addendum

The IBM AS/400 Platform-Specific Addendum for the Niakwa Data Manager (NDM) is designed as an aid to the installation, operation and operating system-specific features of NDM in the following environment:

Niakwa Server:

Hardware	Operating System	Operating Environment
IBM AS/400	OS/400 Ver. V2R2M0 or greater	AS/400 ICF with PC Support/400

NDM Client System:

Hardware	Operating System	Operating Environment
IBM-Compatible PC (80286 or greater)	MS-DOS 5.0 or greater	MS-DOS

P.1.1 Prerequisite Knowledge

This guide assumes at least a basic knowledge of the IBM AS/400, an IBM-compatible Personal Computer (PC), MS-DOS 5.0 or greater and NPL Release IV or greater for the IBM AS/400.

P.1.2 How to Use the NDM Documentation

The NDM documentation consists of two manuals, the Programmer's Guide and the Platform-Specific Addendum. This Platform-Specific Addendum provides IBM AS/400-specific information as it relates to the NDM functions and installation. This manual also documents all IBM AS/400-specific configuration requirements, extended NDM features supported, and NDM limitations on the IBM AS/400. 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 NDM utilities. The NPL IBM AS/400 Supplement, which discusses IBM AS/400-specific information about NPL, should also be referenced.

Additionally, the NDM IBM AS/400 Release Notes provide information pertaining to this platform-specific addendum that was not available at press time.

Refer to the Preface in the NDM Programmer's Guide for more information on the use of the NDM documentation.

P.1.3 How to Use this Addendum

The AS/400 Platform-Specific Addendum discusses the operation of NDM on the IBM AS/400.

P.1.4 Notational Conventions

This guide uses the following notational conventions:

NOTE: Notes provide information of particular importance.

WARNING--Warnings are special conditions that require extra care by the user.

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

Examples in this manual use both the number and name of the API functions. When making API calls using NDM either the number or the name can be used, but not both in the same call.

Release IV long variable names can be used as API parameters.

All alpha parameters used in API function descriptions will end in $ (i.e., MODE$) while numeric parameters do not (i.e., RETURN_CODE).

For example:

```
31060 NDM_CREATE_FILE
      (FILE_NAME$, MODE$, INDEX_NUMBER, KEYDESC_TABLE$, RE-
      CORD_LENGTH, ISAM_SPECIFIC$, FILE_HANDLE$,
      RETURN_CODE)
```

Could be coded as:

```
GOSUB’ 31060 ("EXAMPLE", M$, I, K$, l, A$, H$, R)
```

or

```
GOSUB’NDM_CREATE_FILE ("EXAMPLE", FILENAME$,SHAREMODE$,INDEXNUM,
KD_TABLE$,REC_LENGTH,ISAM_SPECIFIC$,FILE_HANDLE$,RC)
```

TABLE OF CONTENTS

PREFACE

INTRODUCTION

INSTALLING NDM

INTEGRATING EXTERNAL ROUTINES

IBM AS/400 CONSIDERATIONS

END-USER CONSIDERATIONS

CHAPTER 1

INTRODUCTION

1.1 Overview

This Addendum discusses NDM-specific information in regards to the IBM AS/400 implementation of NPL.

NOTE: **Refer to the Niakwa Data Manager Programmer's Guide for any general details relating to the use of NDM.**

Refer to the NPL AS/400 Supplement for NPL-specific information.

The following is a summary of material covered in this addendum:

Chapter 2 discusses installing NDM.

Chapter 3 discusses integrating external routines.

Chapter 4 discusses NDM considerations with respect to the IBM AS/400.

Chapter 5 discusses end-user considerations.

1.2 Technical Overview

The IBM AS/400 implementation of NPL and NDM for the IBM AS/400 uses a client/server technique referred to as "Cooperative Processing." In the Cooperative Processing environment, NPL and the application is executed on a PC connected to the IBM AS/400. The NDM program communicates with the IBM AS/400 using the Advanced Program-to-Program Communications (APPC) protocol, using PC Support/400. The PC is connected to the IBM AS/400 with an APPC-supported physical link such as token-ring or ethernet. Operation of the application is exactly the same as operation on a stand-alone PC or PC-based network. However, the application's data is stored on the IBM AS/400 in a native AS/400 database format compatible with all native AS/400 utilities and programs.

NOTE: NPL program diskimages are typically stored on the PC or the PC-based network. Optionally diskimages could be stored on the IBM AS/400 in shared file folders.

1.2.1 Product Components

NPL and NDM for the IBM AS/400 consist of three primary components:

NPL RunTime

The IBM AS/400 implementation of NPL is nearly identical to the standard Release IV MS-DOS version of NPL. All MS-DOS Release IV features are fully supported, as well as several features specific to the IBM AS/400. The security used with the IBM AS/400 NPL RunTime is different than the security used in the MS-DOS based versions of NPL.

NDM API

Fully compatible with other platforms, but uses the AS/400 database access method as its native ISAM. In addition, NDM is heavily optimized for the IBM AS/400. This addendum discusses the details of this NDM implementation.

NDM Server

A special NDM server task developed by Niakwa. This program is automatically executed on the AS/400 when NPL is started. The server task responds to all requests issued by NPL and NDM. This addendum discusses the details of the NDM server.

CHAPTER 2

INSTALLING NDM

2.1 Overview

This chapter discusses the hardware and software requirements of the IBM AS/400 and associated workstations. It also outlines the steps necessary for proper installation of both the client and server NDM software.

NOTE: The IBM AS/400 Supplement for NPL discusses the specific NPL and NDM installation requirements. Refer to the NPL IBM AS/400 Supplement for more detailed installation information and procedures.

Section 2.2 discusses system requirements for NDM on the IBM AS/400.

Section 2.3 discusses the NDM Development Software on the IBM AS/400.

Section 2.4 discusses starting NDM on the IBM AS/400.

2.2 System Requirements

The following section illustrates the system requirements for NDM to operate correctly in the IBM AS/400 client/server-based environment.

The configuration requirements for the client workstation and the AS/400 server are the same as documented in Chapter 2 of the NPL IBM AS/400 Supplement. Refer to Chapter 2 of the NPL IBM AS/400 Supplement for details.

2.3 NDM Development Software

The NDM Development Software for the IBM AS/400 is provided on floppy diskette and 1/4" tape cartridge. The floppy diskette contains the NDM client software and the tape contains the Niakwa server software and NDM data files.

The following sections discuss the installation procedures for the workstation(s) and for the IBM AS/400 server that apply specifically to NDM. Refer to Chapter 2 of the IBM NPL IBM AS/400 Supplement for details on installing NPL for use with the IBM AS/400.

2.3.1 Installing Client Software

Refer to Chapter 2 of the NPL IBM AS/400 Supplement for details on installing the NDM and NPL software on the workstation's local hard drive.

Diskette Contents

The NDM files contained on the NDM Development Software diskettes are shown below:

NDMUTIL.BAT	NDM utilities MS-DOS batch file.
NDMUTIL.BS2	NDM utilities diskimage.
EX1NDM.OBJ	Simple example number 1
EX2NDM.OBJ	Simple Example program number 2

NDMUTIL.OBJ	NDM utilities boot program.
TESTCUST.OBJ	Boot program for the sample program.
NDMA400.QLB	NDM for IBM AS/400 quick library.

Refer to Chapter 2 of the NPL IBM AS/400 Supplement for details on installing the NDM and NPL software on the PC-client's local hard drive.

2.3.2 Installing Server Software

Refer to Chapter 2 of the IBM AS/400 Supplement for NPL for details on installing the server software on the IBM AS/400.

Tape Cartridge Software

The contents of the tape cartridge for the NDM AS/400 software is listed below:

NOTE: Each NDM file listed below consists of the actual file (as shown) and up to two (3) additional files ending with the characters "01" or "02" or "03". These additional files (not shown) are logical files storing the indices for the listed file.

Only the files that are specific to NDM are shown below. Refer to Chapter 2 of the NPL IBM AS/400 Supplement for a complete list of files on the tape.

CUSTCAT	Catalog file for the sample program.
CUSTDD	Data description file for the sample program.
CUSTEXAM	Data file for the simple example programs.
CUSTKD	Key description file for the sample program.
CUSTREL	Relation file for the sample program.
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.

2.4 Starting NDM

Before NDM can be executed, the NPL and NDM software must be installed on the workstation and the IBM AS/400. Refer to Chapter 2 of the IBM AS/400 Supplement for NPL for details.

The general form for starting the standard MS-DOS implementation NDM from an MS-DOS prompt is:

```
RTIA4.EXE /XNDMA400
```

or

```
RTPA4.EXE /XNDMA400
```

Refer to Chapter 2 of the IBM AS/400 Supplement for NPL for additional details.

NOTE: NPL can be started stand-alone without the quick library (NDMA400), but the PC client must still have an active connection to the IBM AS/400 to pass security.

Applications must be modified to execute the external call **31135 NDM_ESTABLISH_CONNECTION** prior to executing any other NDM calls. Refer to Section 4.11 for details.

The NIAKWA_RUNTIME and NDM environment variables must be set for NPL and NDM to work properly. Refer to Chapter 2 of the NPL IBM AS/400 Supplement for details.

The NDM environment variable is used to specify the IBM AS/400 default NDM library. It is not an MS-DOS directory name.

CHAPTER 3

INTEGRATING EXTERNAL ROUTINES

3.1 Overview

The NDM Client BESDK diskettes supply the necessary object files, source code and example MAKE procedures to allow developers to integrate their own external routines with NDM for the standard (MS-DOS) and MS-Windows implementation of NDM for the IBM AS/400 (two BESDK diskettes are provided).

If an application must use other external routines with NDM, it is necessary to merge these external libraries with NDM into one library (only one external library can be used when starting NPL.

Section 3.2 discusses the NDM MS-DOS Client BESDK diskette.

Section 3.3 discusses the NDM MS-DOS Client BESDK.

Section 3.3 discusses the contents of the NDM MS-DOS Client BESDK

Section 3.4 discusses installing the NDM MS-DOS Client BESDK.

Section 3.5 discusses merging installed libraries.

Section 3.6 discusses Using Non-C External Routines.

Section 3.7 discusses using the merged library.

3.2 NDM MS-DOS Client BESDK

The AS/400 implementation of NDM using a MS-DOS client PC uses the quick library mechanism of the Microsoft Linker. This allows the standard NPL program to load a specified set of routines after NPL starts.

The following sections discuss the details in creating and linking external routines for use with the AS/400 implementation of NDM and NPL using quick libraries (QLB's).

3.3 Contents of the NDM MS-DOS Client BESDK

The NDM MS-DOS Client BESDK diskette contains the following files:

\NDMA	This directory contains the files that, when linked, create the NDM quick library.
MAKEQLB.BAT	Batch file to make the library.
RTP_EXT.C	RTPEXT source code.
RTP_MAIN.C	Mainline.
NDMA400.LIB	NDM routines.
NDMA400.LNK	Response file for the linker.

\NDMAEXAM This directory contains the same files merged with the example external subroutine from the NPL BESDK package, plus the files necessary to compile and link the example. Refer to Chapter 11 of the NPL AS/400 Supplement for information on the NPL external calls.

MAKEQLB.BAT Batch file to make the merged library.

RTP_EXT.C Merged RTPEXT routine.

RTP_MAIN.C Merged mainline.

NDMA400.LNK Response file for the linker.

To make the example merged library, once the NDM MS-DOS client BESDK has been installed, select the NDMAEXAM directory and enter:

```
MAKEQLB
```

3.4 Installing the NDM MS-DOS Client BESDK

The NDMA and NDMAEXAM directories should be installed at the same level as the DOSCEXAM directory of the NPL BESDK directory on the client PC. The MAKEQLB file assumes that this is where the files are located. If these directories are moved to another location, these files must be modified for compiling and linking to succeed.

In the following example, the DOSCEXAM directory of the NDM MS-DOS client BESDK is assumed to be in the C:\BASIC2C directory (the full path name of this directory is C:\BASIC2C\DOSCEXAM). It is also assumed that the NDM MS-DOS Client BESDK disk is in drive A: on the client PC. If this is not the case, substitute the correct names for the system in use.

To install the NDM MS-DOS Client NDM BESDK, follow the steps below:

1. Create the NDM MS-DOS Client BESDK directory, on the client PC, by entering:

   ```
   MD C:\BASIC2C\NDMA
   ```

2. Insert the NDM MS-DOS BESDK diskette into drive A and copy the files from the A:\NDM directory to the client PC hard drive by entering:

   ```
   COPY A:\NDMA\*.* C:\BASIC2C\NDMA
   ```

3. Create the NDMEXAM directory on the client PC hard drive by entering:

   ```
   MD C:\BASIC2C\NDMAEXAM
   ```

4. Copy the files from the A:\NDMEXAM directory to the directory on the client PC hard drive by entering:

   ```
   COPY A:\NDMEXAMA\*.* C:\BASIC2C\NDMEXAMA
   ```

NOTE: It is not necessary to create the above directories in the \BASIC2C directory. It is only necessary to create the NDM MS-DOS Client BESDK directories on the same level as the DOSCEXAM directory from the AS/400 BESDK. Refer to Chapter 11 of the NPL IBM AS/400 Supplement for details.

3.5 Merging Installed Libraries

To merge two external libraries, follow these steps:

1. Copy the files in the NDM MS-DOS BESDK directory to the directory in which the merged library is to be created.

2. Modify the RTP_EXT.C file as follows:

   - Add the C declarations of the other library’s routines at the start of the file, before the C declarations for the NDM routines.

   - 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 statement 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 ndms_init().

   - Add a call to the other termination routine immediately before the call to ndms_done().

   For an example of these changes, review the RTPMAIN.C file in the NDMEXAM directory.

4. Modify the MAKEQLB.BAT file by adding the statements necessary to compile the file(s) containing the other external subroutines.

5. Modify the NDM.LNK (should be NDMA400.LNK) file by adding the names of the object files that will be produced by compiling the other external subroutines. These names should be added after the second line of the file (each name on a separate line followed by a plus (+)).

6. Execute the MAKEQLB.BAT file to compile the necessary files and link them together. This creates the NDM.QLB file which contains the merged external libraries.

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; set_sub() 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.

3.6 Using Non-C External Routines

If the other external routine definitions are written in Pascal or assembly language, it is necessary to do one of the following:

1. Convert these routines to C.

2. Convert the RTP_EXT.C and RTP_MAIN.C to the other language. This should be a simple line-by-line translation. However, depending on the other language and compiler, it may be necessary to put underscores ("_") at the beginning of each routine name.

3.7 Using the Merged Library

The NDMA400.QLB file produced by the above steps contains the user external routines and the NDM API routines.

To install this on any system, copy the NDMA400.QLB file to the NDM directory of a client system. To avoid name confusion, rename the NDMA400.QLB file to the some other name.

CHAPTER 4

IBM AS/400 CONSIDERATIONS

4.1 Overview

This chapter discusses NDM features and considerations specific to NDM on the IBM AS/400.

Section 4.2 discusses supporting multiple indexed files.

Section 4.3 discusses multiple record blocking.

Section 4.4 discusses IBM AS/400 field naming.

Section 4.5 discusses IBM AS/400 file naming.

Section 4.5 discusses data dictionary files.

Section 4.6 discusses IBM AS/400 field classes supported.

Section 4.8 discusses IBM AS/400 file limits.

Section 4.9 discusses IBM AS/400-specific considerations.

Section 4.10 discusses toolbox features.

Section 4.11 discusses IBM AS/400-specific NDM API calls.

Section 4.12 discusses IBM AS/400-specific error codes.

Section 4.13 discusses the NDM Utilities function under the AS/400 implementation of NDM.

4.2 Supporting Multiple Indexed Files

The IBM AS/400 implementation of NDM does not directly support multiple indices. This is because an IBM AS/400 object (physical or logical file) can have, at most, one index. However, a physical file can have several views (logical files) into the file, each with its own index. Thus, to support multiple indices, a separate logical file must be created for each index used by an NDM data file.

4.2.1 Creating Multiple Indices

The IBM AS/400 implementation of NDM provides several methods for creating separate index files (logical files) for each index.

The index file name, for a particular NDM index, can be provided by the user at the time of creating the key description file. The index file name is entered into the INDEX FILE NAME field of the key description file. For each index needed, a separate physical index file must be created.

NOTE: The NDM Utility, Access Key Description File, has been updated to allow the input of this new field. Refer to Section 4.13 for details on the use of this utility.

If no file name is provided in the INDEX FILE NAME field of the key description file, NDM automatically generates a file name for the index. The generated file name s then automatically stored in the INDEX FILE NAME field of the key description file.

When the **INDEX FILE NAME** is generated by NDM, it is based on the name of the associated NDM data file and two additional digits (numbers) based on the index number.

For example, if no INDEX FILE NAME is specified for index number one (1) of an NDM data file name, CUSTOMER, NDM creates the following file for index number one (1) for this file:

CUSTOMER01

Accordingly, index numbers two and three, if they existed, would be named:

CUSTOMER02
CUSTOMER03

In all cases, the logical files are created when the **31060 NDM_CREATE_FILE** or the IBM AS/400-specific NDM call **31055 NDM_CREATE_DEFINED_FILE** is issued.

NOTE: Refer to Section 4.2.3 for details on creating NDM temporary index files.

HINT: To maintain compatibility with existing IBM AS/400 database files, the developer is able to override the default index file name by explicitly assigning a file name to use.

4.2.2 Using Index Files

When an NDM data file is opened using the **31350 OPEN_FILE_FROM_CATALOG** or **31340 NDM_OPEN_FILE,** the association between the NDM data file and its index files is done automatically by NDM and the current index is set to the index number specified by the call.

Either of the following NDM calls could be used to set (change) the current index of an open NDM data file:

31402 NDM_SET_INDEX _FILE_ NAME (AS/400 Specific NDM call)
31400 NDM_SET_CURRENT_INDEX

When using the **31402 NDM_SET_INDEX_FILE_NAME**, the index number passed to this call may be that of any existing index . The file name passed must be that of an existing file on the IBM AS/400 and that file must be a dependent logical of the base file.

NOTE: If the index number is that of an existing index and the file name does not exist, the call returns an NDM 115 error, Index Number Out Of Range.

Refer to Section 4.11 for details on the 31402 NDM_SET_INDEX_FILE_NAME call.

NOTE: The operations of opening and closing the physical index files are performed automatically by NDM and are transparent to the user. This is done to preserve the portability of the NDM application.

4.2.3 Creating Temporary Indices on the IBM AS/400

31080 NDM_CREATE_INDEX is fully supported on the IBM AS/400. By default, **31080 NDM_CREATE_INDEX** generates the name of the logical file for the index as described in Section 4.2.1. If more control over naming of a temporary index file is desired, new IBM AS/400 specific functions **31235 NDM_GET_INDEX_FILE_NAME** and **31402 NDM_SET_INDEX_FILE_NAME** may be used.

The index number passed to **31402 NDM_SET_INDEX_ FILE_NAME,** when creating a temporary index**,** must be the next available index number for the file and the file name must not be that of an existing file on the system. The new index name is stored until the next call to **31080 NDM_CREATE_INDEX** for that file handle. Only one new index name may be preset at a time for each open file handle.

4.3 Multiple Record Blocking

Record blocking by NDM can significantly reduce the number of messages passed between the PC workstation client and the IBM AS/400 server for some applications.

The **31415 NDM_SET_READ_BLOCKING** call can be used to specify the blocking factor. The only modification necessary to the NDM application is to add this call to the application's startup routine. Refer to Section 1.5 of the NDM Revision 1.20 Addendum for details of this call.

NOTE: On the AS/400 implementation of NDM, a performance gain always occurs when the 31370 NDM_READ_BY_POSITION call is issued and the POSITION parameter of this call is equal to 1 or - 1 (positive one or negative 1). Best performance is gained when matching the size of the buffer to the size of the communication block size. Refer to the AS/400 documentation of information on increasing the communication block size. Refer to Section 1.5 of the NDM Revision 1.2 Addendum for details on the 31415 NDM_SET_READ_BLOCKING call.

4.4 IBM AS/400 Field Naming

Unlike other NDM-supported access methods, all fields in an IBM AS/400 database record are defined at the native level, not just the key fields. In addition to the standard attributes (i.e., length, position, type, etc.), a name must be specified for each field in an IBM AS/400 database record. These field names are used by the native IBM AS/400 utilities (such as the Data File Utility (DFU)) so meaningful names are useful. The only way an NDM application can specify field names is through a data description file.

NOTE: For this reason, it is highly recommended that the NDM application use the NDM data description files. Create files by calling 31070 NDM_CREATE_FILE_FROM_CATALOG or 31055 NDM_CREATE_DEFINED_FILE as opposed to calling 31060 NDM_CREATE_FILE.

4.4.1 NDM-Generated Field Names

To maintain compatibility with existing NDM applications, **31060 NDM_CREATE_FILE** is supported on the IBM AS/400. This call uses the key description table to break the record into segments and generates automatic names for them.

Each key segment in an IBM AS/400 database record must be a distinct field. The field names generated by NDM are of the form:

```
FIELD###
```

where ### are sequential numbers beginning at 001.

NOTE: The generation of these field names involves more overhead than using the field names specified in the NDM data description file. As such, 31060 NDM_CREATE_FILE takes longer than the corresponding create, using the data description file.

Because overlapping fields are not supported, if two key segments overlap, they are broken down into two fields, the overlapping part and the non-overlapping part.

To illustrate the automatic naming used by NDM, consider the following example:

NDM key table:

Index	**Segment**	**Offset**	**Length**
1	1	10	20
1	2	50	15
2	2	20	15

Length of example record = 100

IBM AS/400 example database definition:

Field Name	**Length**
FIELD001	10
FIELD002	10
FIELD003	10
FIELD004	05
FIELD005	15
FIELD006	15
FIELD008	35

Key 1 has the following fields:

FIELD002
FIELD003
FIELD006

Key 2 has the following fields:

FIELD003
FIELD004

4.4.2 Data Dictionary-Specified Fields

Although the IBM AS/400 implementation of NDM can use any valid NDM field names, the field naming conventions used in the IBM AS/400's internal representation of files are more restrictive.

The following are IBM AS/400 field naming conventions:

- The first character must be "alphabetic" (i.e., "A".."Z", "@", "#", "$").
- Subsequent characters must be "alphanumeric" (i.e., "A".."Z", "0".."9", "@", "#", "$", "_").

NDM field names which are ten (10) characters or less in length and contain no internal spaces are converted to upper case and used as the IBM AS/400 field names. NDM field names which do not meet those criteria are replaced with automatic NDM-generated field names as specified above. In this case, the original NDM field name is stored as text associated with the IBM AS/400 field. Use of the IBM AS/400-compatible field names makes it easier to use the IBM AS/400-native programs to access NDM-created files.

WARNING -- Because fields are automatically converted to upper-case, it is possible for multiple fields to be converted to the same value, which would cause an error (i.e., "aaaa", "aaaA", "aaAa" would all be converted to "AAAA").

4.5 IBM AS/400 File Naming

The format for a native file name used by the IBM AS/400 implementation of NDM is as follows:

```
[<library-name>/]<file-name>.
```

where:

library-name Optional, but, if specified, must be an existing library. If a library name is not specified, the library list is searched for the specified file name. If no library name is specified when creating or duplicating a file, the newly-generated file is placed in the library QGPL or the first specified library on the list of libraries associated with the user's login. A valid library name is one to ten characters in length and must start with a letter.

file-name A valid IBM AS/400 file name is one to ten characters long and must start with a letter. File names should be restricted to eight characters in length so that the logical files used for indexes can use the base file name followed by a two-digit number.

The following are IBM AS/400 file naming conventions:

- The first character must be "alphabetic" (i.e., "A".."Z", "@", "#", "$").
- Subsequent characters must be "alphanumeric" (i.e., "A".."Z", "0".."9", "@", "#", "$". "_").

4.6 Data Dictionary Files

The layout of certain of the NDM data dictionary files has been modified for the IBM AS/400 implementation of NDM. The following sections discuss these modifications.

4.6.1 Catalog File Layout

The following table shows the catalog file layout for the IBM AS/400 implementation of NDM.

Start	Length	Type	Field Name
1	32	A	FILE TITLE
33	72	A	FILE NAME
105	32	A	DATA DESCRIPTION FILE TITLE
137	32	A	KEY DESCRIPTION FILE TITLE
169	128	A	ISAM SPECIFIC
297	184	A	RESERVED FOR NIAKWA EXPANSION
169	1	N	ISAM CODE
170	2	N	PAGE SIZE *
172	2	N	PREALLOCATION PAGES *
174	1	A	DATA COMPRESSION **
175	1	A	ALLOCATE ***
176	4	N	INITIAL RECORDS ***
180	2	N	INCREMENT SIZE ***
182	2	N	MAX INCREMENTS ***
184	2	N	FILE WAIT TIME ***
186	2	N	RECORD WAIT TIME ***

*	Btrieve use only.
**	C-ISAM use only
***	IBM AS/400 specific

Description of IBM AS/400-Specific Field Names

The following are descriptions of the field names used in the catalog file layout that are unique to the IBM AS/400 implementation of NDM. Refer to Chapter 3 of NDM Programmer’s Guide for a complete description.

ISAM CODE The code for the native ISAM being used be NDM. Possible values for this field are:

Code	Native ISAM
1	Btrieve
2	C-ISAM
5	SDtrieve
6	AS/400

ALLOCATE Indicate if space for the initial number of records should be pre-allocated. Valid values are "N" and "Y".

INITIAL RECORDS The number of records the initial allocation can hold. The default value is 0.

INCREMENT SIZE The number of records the secondary allocation units can hold. A "0" indicates the file is created with NOMAX as the size parameter. If "0" is specified, then fields ALLOCATE, INITIAL RECORDS and MAX INCREMENTS are ignored. The default value is 0.

MAX INCREMENTS The maximum number of records the secondary allocation units can hold. The default value is 0.

FILE WAIT TIME The wait time (in seconds) for file operations (i.e., OPEN) if the file is locked. The default value is 0.

RECORD WAIT TIME The wait time (in seconds) for record operations (i.e., READ) to wait if the record is locked. The default value is 0.

4.6.2 Data Dictionary File Layout

The data dictionary file layout for the IBM AS/400 implementation of NDM is the same as documented in Chapter 3 of the NDM Programmer's Guide.

4.6.3 Key Description Layout

The following is the layout of the key description file for the IBM AS/400 implementation of NDM.

Start	Length	Type	Field Name
1	32	A	FIELD NAME
33	1	A	ASCENDING
34	1	A	UNIQUE
35	2	N	INDEX NUMBER
37	2	N	SEGMENT NUMBER
39	1	A	KEY COMPRESSION
40	34	A	INDEX FILE NAME ***
74	21	A	RESERVED FOR NIAKWA EXPANSION

*** IBM AS/400-Specific

NOTE: The following are descriptions of the field names used in the catalog file layout that are unique to the IBM AS/400 implementation of NDM. Refer to Chapter 3 of NDM Programmer's Guide for a complete description.

INDEX FILE NAME The name of the native file that holds the key information for this index. This field is valid only for index description entries. This field is currently used only on the IBM AS/400 implementation of NDM. For the IBM AS/400, this field should contain the name of a logical file based on the physical file specified in the catalog file.

NOTE: If no entry is provided by the user, in the INDEX FILE NAME field, NDM automatically creates a file name and stores it in this field when the data file is created.

4.7 IBM AS/400 Field Classes Supported

The tables in the following sections describe the various field types supported by the IBM AS/400 implementation of NDM.

4.7.1 Standard Field Types

The following table displays the standard field types supported by the IBM AS/400 implementation of NDM.

Number	Field Class	Name	Min Size	Max Size	Size Increment
1	A	STRING	1	32511	1
2	N	INTEGER	2	2	0
3	N	FLOAT	4	8	4
4	A	LSTRING	1	256	1
5	A	ZSTRING	1	32510	1
6	D	DATE	4	4	0
7	A	TIME	4	4	0
8	N	PACKED DECIMAL	1	255	1
9	N	NUMERIC (BTRV)	1	255	1
10	N	UNSIGNED BINARY	2	4	2
11	N	LONG	4	4	0
12	N	BFLOAT	4	8	4
13	D	ALPHA FLOAT	8	8	0
14	A	ALPHA TIME	8	8	0
15	N	DECIMAL (C-ISAM) ***	2	17	1
16	A	NO CONVERSION	1	32511	1
17	N	ZONED DECIMAL	1	21	1
18	A	TIMESTAMP YYY-MM-DD-HH.MM.SS.UUU	26	26	0

*** Not Supported by the IBM AS/400

Description of Fields

1 Alpha-numeric string. The Data Description Source (DDS) code for this type is A (alpha). The data is converted to EBCDIC form.

2 Standard 2’s complement numeric field. The DDS code for this type is B (binary).

3 Floating point numeric. The DDS code for this type is F (float).

4 Long alpha-numeric string. The DDS code for this type is A (alpha). The data is converted to EBCDIC form. Not supported for key types.

5 Null-terminated string. The DDS code for this type is A (alpha). The data is converted to EBCDIC form. Data beyond the null terminator is meaningless. The IBM AS/400 does not directly support null-terminated strings, so this field is implemented in the same way as a regular string (field type 1).

6 Packed date. This field is a stored as YYYYMMDD in binary-coded decimal (BCD) format (packed decimal with no sign). Because the IBM AS/400 does not support either a 4-byte date or BCD field, the DDS code for this type is H (hexadecimal).

7 Packed time. This field is stored as HHMMSS in BCD format. Because the IBM AS/400 does not support either a 3-byte time or BCD field, the DDS code for this type is H (hexadecimal).

8 Standard packed decimal numeric field. The DDS code for this type is P (packed).

9 Numeric Btrieve. The DDS code for this type is H (hexadecimal). The data is converted to EBCDIC form. Not supported for key types.

10 Standard unsigned binary. The DDS code for this type is B (binary).

11 Long floating point numeric. The DDS code for this type is F (float).

12 Btrieve Floating point. The DDS code for this type is H (hexadecimal). Not supported for key types.

13 Alphanumeric date. This field is stored as YY/MM/DD in EBCDIC format, the DDS code for this type is A (alpha).

14 Alphanumeric time. This field is a stored as HH:MM:SS in EBCDIC format,. the DDS code for this type is A (alpha).

15 Not currently supported by the IBM AS/400 implementation of NDM (C-ISAM numeric).

16 No conversion. This field does no conversion from the NPL field type. This is useful for overlay fields or alpha fields that need to be stored in ASCII format. The DDS code for this type is H (hexadecimal).

17 Zoned decimal. This field type is the same as a NPL type IBM display field. The DDS code for this type is S (zoned).

18 Timestamp. This is a timestamp of the form YYYY-MM-DD-HH.MM.SS.UUUUUU in EBCDIC format. The DDS code for this type is Z (timestamp).

4.7.2 Date Field Types

The following table displays the date fields supported by the IBM AS/400 implementation of NDM.

Number	**Field Class**	**Name**	**Min Size**	**Max Size**	**Size Increment**
50	D	ISO DATE (YYYY-MM-DD)	10	10	0
51	D	USA DATE (MM/DD/YYYY)	10	10	0
52	D	EUR DATE (DD.MM.YYYY)	10	10	0
53	D	JIS DATE (YYYY-MM-DD)	10	10	0
54	D	DATE (MM/DD/YY)	8	8	0
55	D	DATE (MM-DD-YY)	8	8	0
56	D	DATE (MM.DD.YY)	8	8	0
57	D	DATE (MM.DD.YY)	8	8	0
58	D	DATE (MM DD YY)	8	8	0
59	D	DATE (DD/MM/YY)	8	8	0
60	D	DATE (DD-MM-YY)	8	8	0
61	D	DATE (DD.MM.YY)	8	8	0
62	D	DATE (DD,MM,YY)	8	8	0
63	D	DATE (DD MM YY)	8	8	0
64	D	DATE (YY/MM/DD)	8	8	0
65	D	DATE (YY-MM-DD)	8	8	0
66	D	DATE (YY.MM.DD)	8	8	0
67	D	DATE (YY,MM,DD)	8	8	0
68	D	DATE (YY MM DD)	8	8	0

Number	Field Class	Name	Min Size	Max Size	Size Increment
69	D	DATE (YY/DDD)	8	8	0
70	D	DATE (YY-DDD)	8	8	0
71	D	DATE (YY.DDD)	8	8	0
72	D	DATE(YY,DDD)	8	8	0
73	D	DATE (YY DDD)	8	8	0

The date fields all have a DDS code of L (date). Each date field also has a DDS specification for the format "DATFMT". The first four date types have the following DATFMT specifications:

DATFMT(*ISO)	ISO date format
DATFMT(*USA)	IBM USA Standard date format
DATFMT(*EUR)	IBM European Standard format
DATFMT(*JIS)	Japanese Industrial Standard

The remaining date formats have a specification of *MDY, *DMY, *YMD, *JUL. For each of these types, a date separator ("DATSEP") is also specified and can be "/", "-", ".", "," or " ".

4.7.3 Time Field Types

The following chart lists the time field types supported by the IBM AS/400 implementation of NDM.

Number	Field Class	Name	Min Size	Max Size	Size Increment
74	A	ISO TIME (HH.MM.SS)	8	8	0
75	A	USA TIME (HH:MM AM/PM	8	8	0
76	A	EUR TIME (HH.MM.SS)	8	8	0
77	A	JIS TIME (HH:MM:SS)	8	8	0
78	A	TIME (HH:MM:SS)	8	8	0
79	A	TIME (HH.MM.SS)	8	8	0
80	A	TIME (HH MM SS)	8	8	0

The time fields all have a DDS code of T (time). Each time field also has a DDS specification for the format "TIMFMT". The first four time types have the following TIMFMT specifications:

TIMFMT(*ISO)	ISO date format
TIMFMT(*USA)	IBM USA Standard date format
TIMFMT(*EUR)	IBM European Standard format
TIMFMT(*JIS)	Japanese Industrial Standard

The remaining time formats have specification of *HMS. For each of these types, a time separator ("TIMSEP") is also specified and can be ":", ".", or " ".

4.8 AS/400 File Limits

Description	**NDM**	**IBM AS/400**
Record Length	4090	32766
Key Length	120	255
No. of keys/file	9	100
No. of segs/key	8	119*

* NDM on the IBM AS/400 sets a maximum key table size. This imposes an effective restriction on the number of keys-per-file and the number of segments-per-key. The maximum number of segments allowed for a key by the IBM AS/400’s database system is 119.

NOTE: **Enable the appropriate NDM Tool Box features, using 31420 NDM_SET_TOOL_BOX_STATUS, to use the higher file limits available with the IBM AS/400 implementation of NDM. Refer to Section 4.10 for details.**

4.9 IBM AS/400 Considerations

The following is a list of IBM AS/400-specific issues that must be considered when working with the IBM AS/400 implementation of NDM.

When creating an entry in the catalog file, the following sub fields in the ISAM-specific field must be defined:

- ISAM CODE must be set to 6 for AS/400
- PAGE SIZE is not used
- PREALLOCATION PAGES is not used.

The following fields may also be used. Refer to Section 4.6.1 for the description and legal values for these fields:

ALLOCATE
INITIAL RECORDS
INCREMENT SIZE
MAX INCREMENTS
FILE WAIT TIME
RECORD WAIT TIME

NOTE: An NDM application on the IBM AS/400 must first issue the NDM 31135 NDM_ESTABLISH_CONNECTION before any other NDM calls.

4.10 Toolbox Features

Most native ISAMs provide features that are not supported by all NDM-supported native ISAMs. Therefore, to maximize portability of NDM applications, these features are not normally supported under NDM. However, applications unconcerned about portability may make use of these features through an NDM feature called the toolbox. The toolbox consists of a set of NDM features, each of which may be individually enabled or disabled through a call to API function **31420 NDM_SET_TOOLBOX_STATUS**. Therefore, non-portable applications are easily identified by their call(s) to this special API function.

For more information regarding the NDM toolbox feature, refer to Chapter 5 of the NDM Programmer’s Guide.

NOTE: The current status of each toolbox feature can be determined by calling another API function, 31310 NDM_GET_TOOLBOX_STATUS.

Below is a list of the toolbox features Niakwa is supporting under the the IBM AS/400 implementation of NDM.

4.10.1 Key Types

NDM key segments can normally be string or integer. When the key types toolbox feature is enabled, other types of fields are allowed as key segments. Refer to Section 4.7.1 for a list of this fields.

NOTE: Even when the key types toolbox feature is enable, field type 9, NUMERIC (BTRV) and field type 12, BFLOAT, cannot be used as key segments.

4.10.2 File Limits

Using this toolbox feature allows NDM files to use the IBM AS/400 limits defined in Section 4.8.

4.10.3 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.

On the IBM AS/400, transaction processing requires the presence of the following system objects before transaction processing can take place.

NOTE: The objects shown below must be created before transaction processing can occur. They should be set up using the IBM AS/400 commands shown below before executing the NDM application. No changes to the NDM application itself (if ported from another supported NDM platform) are required.

JOURNALS	An object that can have one or two journal receivers assigned to it.
JOURNAL RECEIVERS	Objects that store the journalled data.

The journal receiver must be created first. This is accomplished using the IBM AS/400 "CRTJRNRCV" command. Once the journal has been created, a JOURNAL object must be created, using the IBM AS/400 "CRTJRN" command.

For example, to create a journal receiver JRNRCV and a journal JRN associated with it, enter the following from the AS/400 command line:

```
CRTJRNRCV JRNRCV(JRNRCV)
CRTJRN  JRN(JRN)  JRNRCV(JRNRCV)
```

NOTE: **If the file is not located in the default IBM AS/400 library, it is necessary to specify the library name as part of the data file name.**

Once the journal has been created, physical files (NDM data files) can be selected for journalling by using the IBM AS/400 "STRJRNPF" (start journalling physical file) command.

For example, to enable journalling to take place in an NDM application, for the NDM data file CUSTOMER, enter the following for the IBM AS/400 command line:

```
STRJRNF FILE(CUSTOMER) JRN(JRN)
```

NOTE: **If the file is not located in the default IBM AS/400 library, it is necessary to specify the library name as part of the data file name.**

Several options are available for the STRJRNPF command. Some of these options allow the user to specify that changes to the file should be journalled in a before/after format, or allow the selection to omit open and close events from the journal. Refer to the AS/400 documentation for details on other options available for the STRJRNPF command.

4.10.4 Create/Delete Index

This feature allows access to the **31080 NDM_CREATE_INDEX** and **31110 NDM_DELETE_INDEX** NDM API calls. These are used to create temporary indices on existing data files. Refer to Section 4.2 for details.

4.11 IBM AS/400 Specific API Calls

Several new APIs have been added to the NDM specification to support features of the IBM AS/400 and to increase performance in a cooperative processing architecture. The new calls are listed below:

31055 NDM_CREATE_DEFINED_FILE
31135 NDM_ESTABLISH_CONNECTION
31235 NDM_GET_INDEX_FILE_NAME
31402 NDM_SET_INDEX_FILE_NAME

The following sections provide a detailed listing of the IBM AS/400-specific API calls.

4.11.1 NDM_CREATE_DEFINED_FILE

```
'31055  NDM_CREATE_DEFINED_FILE
        (DD_HANDLE$, KEYDESC_HANDLE$,FILE_NAME$, MODE$,IN-
        DEX_NUMBER, FILE_HANDLE$, RETURN_CODE)
```

Input Parameters

DD_HANDLE$ - a handle to an open data description file that describes the fields referred to in the KEYDESC_HANDLE$ file.

KEYDESC_HANDLE$ - a handle to an open key description file containing the desired key information.

FILE_NAME$ - the name of the file to be created. This must not include an extension (unless the "filename extensions" toolbox feature is enabled).

MODE$ - one character:

S	open in shared mode
X	open in exclusive mode
R	open in read only mode.

INDEX_NUMBER - the value to which the current index of the new file should be set.

Output Parameters

FILE_HANDLE$ - a handle to the newly created file, opened in the specified mode.

Description

This function creates a new data file with the specified name and with the characteristics defined by the supplied data description file and key description file. This function does not create a catalog file entry for the new file - this must be done in a separate call. This call enables the application to create a file using the data description without having to create a catalog record. An example of where this is useful is creating a user CATALOG file.

Calling this routine is equivalent to the following calls:

31090 NDM_CREATE_KEY_TABLE
31290 NDM_GET_RECORD_LENGTH_FROM_DD
31060 NDM_CREATE_FILE

As with **31060 NDM_CREATE_FILE**, the current record of the newly created file is "undefined" and passing a value of zero in the INDEX_NUMBER parameter means the current index of the newly created file is also undefined.

The index numbers supplied in the key description are used by NDM only to determine the boundaries between different keys. This means that if the indices are not numbered consecutively, they are renumbered by NDM.

For example, if the key description file (which is usually created from the data description file created by the NDM Utilities) specifies index numbers 1, 1, 3, 3, 3, 5, 0 for its segments, the file created has a two-segment index as index number 1, a three-segment index as index number 2, and a single-segment index as index number 3. This may cause confusion when running the NDM Utilities if the expected index numbers were 1, 3, 5.

Prerequisites

- The data description file and key description file must be open.

Currency Changes

The current record and current index of both the data description and key description files can be changed by this call. The current index of the newly created file is set to the value passed in INDEX_NUMBER (unless it's zero). The current record of the newly created file is "undefined".

Toolbox Features

If the "key types" feature is enabled (feature #2), then the indices described in the key description file parameter may contain native ISAM specific key types.

If the "file limits" feature is enabled (feature #3), then the NDM limits are replaced by the native ISAM's limits (i.e., depending on the native ISAM, a key longer than 120 bytes may be defined).

Possible Errors

10 Invalid file name.

37 File already exists.

40 Lost position. The current record of either the data description file or the key description file has been deleted or its key has been modified.

42 A required record in either the key description file or the data description file is in use.

43 Either the key description file or the data description file is in use.

48 Permission error. Network privileges disallow file creation.

115 Index number is out of range.

See Also:

31020	NDM_CLOSE_FILE
31060	NDM_CREATE_FILE
31090	NDM_CREATE_KEY_TABLE
31290	NDM_GET_RECORD_LENGTH_FROM_DD
31340	NDM_OPEN_FILE

4.11.2 NDM_ESTABLISH_CONNECTION

```
'31135 NDM_ESTABLISH_CONNECTION
       (UNAME$, RETURN_CODE)
```

Input Parameters

UNAME$ - a variable containing the $UNAME system variable.

Output Parameters

None.

Description

This establishes the connection between the IBM AS/400 server that was started by NPL and the NDM external library. The NDM application must issue this call before any other NDM calls are made.

NOTE: This call must be issued prior to the 31320 NDM_INITIALIZE call.

Pre-Requisites

None .

Currency Changes

None.

Toolbox Features

None.

Possible Errors

154 General Communications Error

See Also:

31320 NDM_INITIALIZE

4.11.3 NDM_GET_INDEX_FILE_NAME

```
’31235 NDM_GET_INDEX_FILE_NAME
       (FILE_HANDLE$, INDEX_NUMBER, FILE_NAME$,
       RETURN_CODE)
```

Input Parameters

FILE_HANDLE$ - the handle of the open file to be queried.

INDEX_NUMBER - the number of the index we are querying.

Output Parameters

FILE_NAME$ - the name of the index file. The maximum length for this parameter is 10 characters.

Description

This function gets the name of the file that contains the key records for the specified index.

Prerequisites

- The FILE_HANDLE$ file must be open.

Currency Changes

None.

Toolbox Features

None.

Possible Errors:

None.

See Also:

31402 NDM_SET_INDEX_FILE_NAME

4.11.4 NDM_SET_INDEX_FILE_NAME

```
'31402 NDM_SET_INDEX_FILE_NAME
       (FILE_HANDLE$, INDEX_NUMBER, FILE_NAME$, RE-
       TURN_CODE)
```

Input Parameters

FILE_HANDLE$ **-** the handle of the open file to be queried.

INDEX_NUMBER **-** the number of the index with which a specific file is associated. Must be between 2 and the number of indexes + 1.

FILE_NAME$ **-** the name of the index file to be set.

Output Parameters

None.

Description

This function can be used either to change an existing index for the file or to pre-set the name of the next index to be created for that file using **31080 NDM_CREATE_INDEX**.

The first case is used to set an index number to point to a specific IBM AS/400 logical file associated with the physical file. The index number may be any of the existing indices except index 1 or it may be the next index number available for the file. The function looks for a logical file with the specified name, and if it finds it, sets up the handle to associate the index to the specified file.

The second case is used to create a new index and give it a specific name. The index number must be the next one available for the file. The index name specified is used by the next call to **31080 NDM_CREATE_INDEX**. Only one index name may be pre-set for each file at a time. If multiple indices are to be created in this way, then **31080 NDM_CREATE_INDEX** must be called after each call to **31402 NDM_SET_INDEX_FILE_NAME**.

Prerequisites

- The FILE_HANDLE$ file must be open.

Currency Changes

None.

Toolbox Features

None.

Possible Errors

115 Index Number out of range

See Also

31235 NDM_GET_INDEX_FILE_NAME

4.12 Detailed Listing of IBM AS/400 Error Codes

The error codes used in the IBM AS/400 implementation of NDM are divided into three categories:

- PC support router API errors. 1-99.
- AS/400 server errors. 101-999.
- AS/400 file I/O errors 3000-4000.

Additional NDM error codes specific to the IBM AS/400 implementation of NDM are listed in Section 4.12.4. Original NDM error codes are listed in Chapter 11 of the NDM Programmer's Guide. NPL system error codes and messages are listed in Appendix B of the NPL Programmer's Guide. MS-DOS error messages are listed in Appendix B of the MS-DOS Supplement for NPL.

4.12.1 PC Support Router API Errors

These errors are returned by the PC support router API library routines.

0	No error - the function completed successfully
1	Deallocation normal
2	Program error - no truncation
3	Program error - truncation
4	Program error - purging
5	Resource failure, retry
6	Resource failure, no retry
7	Unsuccessful
8	APPC busy
9	Parameter check - invalid verb
10	Parameter check - invalid conversion id
11	Parameter check - buffer crosses segment
12	Parameter check - TP name length
13	Parameter check - invalid conversion type
14	Parameter check - bad synch level on allocation
15	Parameter check - bad return control
16	Parameter check - pip too long

17	Parameter check - bad partner name
18	Parameter check - confirm not allowed
19	Parameter check - bad Deallocation type
20	Parameter check - prepare to receive type
21	Parameter check - bad fill type
22	Parameter check - receive max length too big
23	State Check - confirm issued not in confirm stat
24	State Check - receive immediate and not in receive
25	State Check - request to send in bad state
26	State Check - send issued in bad state
27	State Check - send error in bad state
28	Allocation error - no retry
29	Allocation error - retry
30	Allocation error - program not available, no retry
31	Allocation error - tpn not recognized
32	Allocation error - program not available retry
33	Allocation error - security not valid
34	Allocation error - conversion type mis-match
35	Allocation error - pip not allowed
36	Allocation error - pip not correct
37	Allocation error - synch level not supported
38	DeAllocation abend program
39	Router not installed
40	Insufficient memory
41	Undefined
42	Too many conversations
99	ICF message too long

4.12.2 IBM AS/400 Server Errors

The following errors are defined by and returned from the AS/400 server:

101	System file not found
102	Too many segments
103	Invalid lock option
104	Maximum number of handles exceeded
105	Key buffer to small
106	Invalid direction
107	Invalid ISAM call
109	Invalid seek type
110	File deletion failed
112	Error trying to get version number
115	Invalid position handle
118	Index number not found or invalid
119	Could not open index file
122	Bad key information
123	Invalid system call
128	Transaction processing not enabled
131	Out of memory
132	Error accessing serial number file
133	Serial number registration error
134	Error accessing active user file
135	User count exceeded
136	Error accessing ID file
137	Error accessing terminal ID file
140	Conversion table not found
141	Bad conversion type
142	Bad conversion length
143	Invalid classes for conversion

4.12.3 AS/400 File I/O Errors

The following errors are encountered by the IBM AS/400 server when making record I/O calls for accessing IBM AS/400 database files.

Errors Set by the IBM AS/400 I/O Routines

3004	File has not been opened
3005	File not opened for read
3009	File not opened for write
3014	Invalid file name specified
3015	Invalid file open mode specified
3030	Command invoked by system failed
3041	File not opened for update
3042	File not opened for delete
3044	Bad key length
3101	I/O exception non-recoverable error
3102	I/O exception recoverable error
4098	Unable to allocate object

Errors Set by the IBM AS/400 Operating System

NOTE: The number in parenthesis indicates the IBM AS/400 error message code associated with the errors. The last four digits of the code are in hexadecimal.

16641	File not found	(CPF4101)
16680	File locked to another process	(CPF4128)
20481	End of file encountered	(CPF5001)
20486	Record not found in file	(CPF5006)
20487	Record deleted	(CPF5007)
20504	Maximum file size reached	(CPF5018)
20517	Attempted to read past end of file	(CPF5025)
20518	Duplicate keys not allowed	(CPF5026)
20519	Record locked by another task	(CPF5027)
20520	Record key changed	(CPF5028)
20521	Data mapping error	(CPF5029)
20528	Partial damage to file member	(CPF5030)
20529	Maximum number of locks exceeded	(CPF5031)
20530	Record already locked to this job	(CPF5032)
20531	Select/omit error	(CPF5033)
20532	Duplicate key on access path	(CPF5034)
20544	Record not read	(CPF5040)

20601	Commitment control limit reached	(CPF5079)
22547	File already exists	(CPF5813)
29442	File not created	(CPF7302)

4.12.4 NDM API Errors

The following error codes are returned by the NDM API that have been added for the IBM AS/400 implementation of NDM. Refer to Chapter 11 of the NDM Programmer's Guide and the NDM 1.2 Addendum for a complete list of all NDM error codes.

IBM AS/400 Specific Error Codes:

154 - General communications error.

Explanation : This indicates that an error occurred in the Inter-systems Communications Function (ICF) connecting PC-Support with the server process. This could occur if the server process is aborted by the operating system, or if an error occurs in the communications line.

Help: Check the job log for more information as to why the communications failure occurred. It will be necessary to restart NPL and NDM.

155 - Error trying to get version number.

Explanation: This error suggests a failure in the NDMMRI program or the operating system. The most likely cause is that another program wit the name NDMMRI occurred on the library list of the server process prior to the one shipped by Niakwa.

Help: Verify that the NDMMRI program is valid

156 - Index file was not found.

Explanation: This error occurs if the logical or physical file which contained the relevant index was inaccessible (the file has been deleted or is locked by another process).

Help: Ensure that the file is present and available and retry the operation. The file name can be retrieved with **31235 NDM_GET_INDEX_FILE_NAME**. It is recommended to close and reopen the file.

157 - Invalid system call.

Explanation: The server encountered a system error while attempting to access information about a file (such as the number of indices, the list of fields, or the key sequences). This may indicate that a deletion of one of the indices of the file by another user while the file was in use occurred.

Help: Ensure that the file is present and available. It is recommended to close and re-open the file.

158 - Invalid ISAM call.

Explanation: Either a software or an ICF communications failure has occurred, because the sever function requested by the quick library is no valid. This could occur is the versions of NPL and NDM installed on the server and PC-client are not the same.

Help: Make sure that the versions of NPL and NDM installed on the server and the PC-client are the same. Contact Niakwa for more information.

159 - File is full.

Explanation: The file in use contains the maximum number or records that were specified when it was created.

Help: It may be possible to create space in the file by removing deleted records with the IBM AS/400 RGZPFM (Reorganize Physical File Member) command. It may also be possible to increase the size of the file wit the IBM AS/400 CHGPF (Change Physical File) command. This command must be followed wit the RGZPFM command for the change to be applied to the existing file member.

4.13 NDM Utilities under the IBM AS/400

The following NDM Utilities have been updated for the IBM AS/400 implementation of NDM.

- Access Catalog File
- Access Key Description File
- Convert Native File to NPL Format
- Convert NPL Format Files to Native Files
- Set NDMUTIL Options

The following sections discuss the changes made to these utilities. Refer to Section 4.12 of the NDM Programmer's Guide and Chapter 2 of the NDM 1.20 Addendum for a complete discussion of the NDM Utilities.

4.13.1 Access Catalog File

The following fields, which define file creation parameters, have been added to the options displayed on the View Catalog Main Screen:

INITIAL RECORDS
INCREMENT SIZE
MAX INCREMENTS
MAX INCREMENTS
FILE WAIT TIME
RECORD WAIT TIME

Refer to Section 4.6 for an explanation on the use and the definitions of these fields.

4.13.2 Access Key Description File

The following field, which defines key definition parameters, has been added to the options displayed on the View Key Description File Screen:

INDEX FILE NAME

Refer to Section 4.6 for an explanation of the use and the definition of this field.

4.13.3 Convert Native File to NPL Format

This utility has been updated to operate correctly with the IBM AS/400 implementation of NDM. The utility operates as documented in Section 4.13 of the NDM Programmer's Guide.

NOTE: **When converting NDM applications from other supported NDM platforms (including MS-DOS based) to the IBM AS/400 it is first necessary to use this utility to covert the native ISAM data to NPL diskimage format and then use the Convert NPL Format Files to Native Files utility to port the data to the IBM AS/400 native format.**

4.13.4 Convert NPL Format Files to Native Files

This utility has been updated to operate correctly with the IBM AS/400 implementation of NDM. The utility operates as documented in Section 4.13 of the NDM Programmer's Guide.

NOTE: **When converting NDM applications for other supported NDM platforms (including MS-DOS based) to the IBM AS/400 it is first necessary to use the Convert Native File to NPL Format utility and then use this utility to port the data to the IBM AS/400 native format.**

4.13.5 Set NDMUTIL Options

The Set Options Screen has been updated with two additional fields. These fields are:

READ BLOCKING FACTOR
MAXIMUM KEY SEGMENTS

Refer to the Section 2.2.4 of the NDM Revision 1.2 Addendum for NDM for explanations on these fields.

CHAPTER 5

END-USER CONSIDERATIONS

5.1 Overview

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

Section 5.2 discusses the minimum file requirements for the client system and the IBM AS/400 server.

Section 5.3 discusses the minimum file requirements for the NDM Error Description File.

Section 5.4 discusses providing the NDM Utilities to end-users.

Section 5.5 discusses the data dictionary files.

Section 5.6 discusses the catalog files.

Section 5.7 discusses the field type file.

5.2 Minimum NDM Software Requirements

There are specific files that must be present at the end-user site for both the Client and IBM AS/400 portion of the NDM software for the IBM AS/400. The following lists the files that are necessary:

Minimum NDM File Requirements for the Workstation (Client) System

The following are the the files necessary on the end-user's PC-client workstation:

NDMA400.QLB	NDM for the IBM AS/400 quick library.

Refer to Chapter 2 of the IBM AS/400 Supplement and Chapter 2 of this Addendum for details on the NPL files required at an end-user site..

Minimum File Requirements for the IBM AS/400 (Server)

There are several conditions that require the developer to provide additional NDM files from the NDM and NPL IBM AS/400 server tape to the end-user's system. The following sections discuss these conditions and which NPL IBM AS/400 server tape software must be provided to the end-user.

5.3 NDM Error Description File

If the application uses the NDM API function call **31210 NDM_GET_ERROR_DESCRIPTION**, the following file must be provided to the end-user on the IBM AS/400 server:

NOTE: **Each NDM file listed below consists of the actual file (as shown) and up to three (3) additional files ending with the characters "01", "02" or "03". These additional files (not shown) are logical files storing the indices for the listed file.**

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

5.4 Providing the NDM Utilities to End Users

If the developer wishes to install the NDM Utilities on the end user's IBM AS/400 system, the following files must be provided:

NOTE: Each NDM file listed below consists of the actual file (as shown) and up to three (3) additional files ending with the characters "01", "02", or "03". These additional files (not shown) are logical files storing the indices for the listed file.

Files Required on the IBM AS/400 (Server)

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.

Files Required on the Workstation (Client)

NDMUTIL.BS2	The NDM utilities diskimage.
NDMUTIL.BAT	Program to start the NDM utilities.
NDMUTIL.OBJ	Boot program used when starting the NDM utilities.

5.5 Data Dictionary Files

If the application's data dictionary files are accessed under program control, the following files must be installed on IBM AS/400.

NOTE: **Each NDM file listed below consists of the actual file (as shown) and up to three (3) additional files ending with the characters "01", "02" or "03". These additional files (not shown) are logical files storing the indices for the listed file.**

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.

NOTE: **If the data dictionary files are not accessed directly, the end-user does not require these files.**

5.6 Catalog File

If the application builds catalogs under program control, the following files are required on the IBM AS/400:

NOTE: Each NDM file listed below consists of the actual file (as shown) and up to three (3) additional files ending with the characters "01", "02" or "03". These additional files (not shown) are logical files storing the indices for the listed file.

NDMCATD	Data description file for the catalog file.
NDMCATK	Key description file for the catalog file.

5.7 Field Type File

If the application uses **31200 NDM_GET_DEFAULT_FIELD_TYPE**, the following file is necessary on the IBM AS/400:

NOTE: Each NDM file listed below consists of the actual file (as shown) and up to three (3) additional files ending with the characters "01", "02" or "03". These additional files (not shown) are logical files storing the indices for the listed file.

NDMFTF	Field type file.