

# **REVISION 1.20 ADDENDUM**

**for the**

# **NIAKWA DATA MANAGER**

1st Edition - October 1993  
COPYRIGHT © 1993 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 (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.

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

The Revision 1.20 Addendum for the Niakwa Data Manager (NDM) is designed as an addition to the NDM Programmer's Guide. This Revision 1.20 Addendum discusses the features and enhancements available on all Revision 1.20 NDM platforms. Refer to the appropriate NDM platform specific addendum for operating system specific features. The NPL platforms supported by NDM currently include:

Hardware	Operating System	Operating Environments	ISAM Supported
IBM and 100 % Compatibles	MS-DOS 3.1 or greater	Novell Netware 2.11 or greater. MS-Windows 3.0 or greater. Phar Lap 386/DOS Extender 5.X Or greater.	Btrieve 5.10 or greater
Approved Intel based 80386 or greater	Intel Unix	Altos Unix AT&T Unix Interactive Unix SCO Unix	C-ISAM 4.00 or greater for the specific operating environment in use.
IBM RS/6000	AIX Version 3.XX or greater	N/A	C-ISAM 4.00 or greater.

---



Hardware	Operating System	Operating Environments	ISAM Supported
IBM AS/400 Server	OS/400 V2R2M0 or greater		Native AS/400
IBM 80286 and 100% Compatible or greater Clients	MS-DOS 5.0 or greater	MS-Windows 3.1 or greater.	

Refer to the NPL and NDM Platform-Specific Addendum and Supplements for additional information.

## TABLE OF CONTENTS

### PREFACE

### TABLE OF CONTENTS

151-

### NDM ENHANCEMENTS

1.1 Overview.....	1-1
1.2 Bug Fixes.....	1-2
1.3 Security.....	1-2
1.3.1 NDM Security Eliminated.....	1-2
1.3.2 NDM RunTime Disk.....	1-2
1.3.3 User Limit.....	1-2
1.3.4 Serial Number.....	1-2
1.4 Release IV Support.....	1-3
1.5 New NDM Functions.....	1-3
1.5.1 NDM_CREATE_CONV_TABLE_FOR_KEY_FROM_CAT.....	1-4
1.5.2 NDM_CREATE_CONV_TABLE_FROM_CAT.....	1-7
1.5.3 NDM_OPEN_FILE_FROM_CATALOG.....	1-9
1.5.4 NDM_SET_MAX_KEY_SEGS.....	1-11
1.5.5 NDM_SET_READ_BLOCKING.....	1-12

1.6 New Return Messages.....	1-14
151-Toolbox feature not enabled for this ISAM.....	1-14
152-Mismatched field types in conversion table.....	1-14
153-Toolbox file limits not activated.....	1-14

### NDM UTILITY ENHANCEMENTS

2.1 Overview.....	2-1
2.2 Updated Utilities.....	2-1
2.2.1 Access Catalog File.....	2-2
2.2.2 Access Key Description File.....	2-2
2.2.3 Export Data Dictionary to IQ Utility.....	2-2
Features.....	2-3
General Use.....	2-3
Converting an NDM Data Dictionary To IQ 3.0.....	2-3
2.2.4 Set NDMUTIL Options.....	2-5
2.2.5 Display Program Information.....	2-6

---

# CHAPTER 1

## NDM ENHANCEMENTS

### 1.1 Overview

The following chapter discusses the enhancements made with Revision 1.20.

Section 1.2 discusses NDM Revision 1.00 bug fixes.

Section 1.3 discusses Revision 1.20 security issues.

Section 1.4 discusses NPL Release IV support.

Section 1.5 discusses the new NDM API functions.

Section 1.6 discusses the new NDM return codes.

---

## 1.2 Bug Fixes

Most known NDM 1.00 bugs have been corrected in Revision 1.20. Refer to the NDM 1.20 Release Notes for any outstanding problems.

## 1.3 Security

The following section discusses the changes made to the NDM security in Revision 1.20.

### 1.3.1 NDM Security Eliminated

All security checks have been eliminated from NDM 1.20.

### 1.3.2 NDM RunTime Disk

The NDM RunTime disk has been eliminated, due to the fact that the NDM security checks being eliminated. Files, not related to the security check, that had been previously on the RunTime disk are now located on the Development Software. Refer to the Platform-Specific Addendum for a list of these files. Developers can install any of the NDM files necessary for the developer's application at the end-user site.

**NOTE: NDM is sold on a per-programmer and per-platform basis, and developers must be licensed for a NDM Pak on a specific platform before they can distribute NDM. Updates for developers who already have a NDM Starter Kit are available. Contact the Niakwa Sales Department for details.**

### 1.3.3 User Limit

The user limit check is now only enforced on the NPL RunTime. No user check is performed by NDM 1.20.

### 1.3.4 Serial Number

The serial number for NDM diskettes has been eliminated.

## 1.4 Release IV Support

NDM 1.20 supports calling all NDM functions by number or function name.

For example, the '31320 NDM\_INITIALIZE call can be called by either of the following methods:

```
GOSUB '31320 (10,10,20,1024,R)
```

or

```
GOSUB 'NDM_INITIALIZE (10,10,20,1024,R)
```

**NOTE: NDM 1.20 is still compatible with Revision 3.20 of NPL.**

The NDM Utilities (NDMUTIL) have been enhanced for mouse support under MS-DOS based platforms.

## 1.5 New NDM Functions

The following are new NDM API functions calls introduced in Revision 1.20 of NDM. The first three functions are implemented to reduce the number of calls needed to perform a specific operation. The last two are new ISAM specific additions to increase performance and functionality.

```
31042 NDM_CREATE_CONV_TABLE_FOR_KEY_FROM_CAT  
31045 NDM_CREATE_CONV_TABLE_FROM_CAT  
31350 NDM_OPEN_FILE_FROM_CATALOG  
31405 NDM_SET_MAX_KEY_SEGS  
31415 NDM_SET_READ_BLOCKING
```

The following section describes each of these new functions in detail.

### 1.5.1 NDM\_CREATE\_CONV\_TABLE\_FOR\_KEY\_FROM\_CAT

```
'31042NDM_CREATE_CONV_TABLE_FOR_KEY_FROM_CAT
      ( CATALOG_HANDLE$, FILE_TITLE$, INDEX_NUMBER,
        CONVERSION_TABLE$, KEY_IN_RECORD$, RETURN_CODE
      )
```

#### Input Parameters

CATALOG\_HANDLE\$ - the handle to the open catalog file containing the entry for the file whose conversion table is to be created

FILE\_TITLE\$ - a 32-character mnemonic name for the file, which is matched against the "FILE TITLE" field of the catalog file.

INDEX\_NUMBER - the number to match the "INDEX NUMBER" field of the key description record.

KEY\_IN\_RECORD\$ - contains "Y" if the keys to be converted by this conversion table are in their correct places in the NPL record or "N" if they are contiguous at the start of the record.

#### Output Parameters:

CONVERSION\_TABLE\$ - a two-byte handle to the newly created conversion table.

#### Description:

This function creates a conversion table describing the key segments of a particular index of a file. This table can then be used to convert keys before performing key operations such as **31360 NDM\_READ\_BY\_KEY** .

Calling this routine is equivalent to the following calls:

```
31150 NDM_GET_CATALOG_ENTRY
31340 NDM_OPEN_FILE (Data Description File)
31340 NDM_OPEN_FILE (Key Description File)
31040 NDM_CREATE_CONV_TABLE_FOR_KEY
31020 NDM_CLOSE_FILE (Data Description File)
31020 NDM_CLOSE_FILE (Key Description Files)
```

If any of the above calls return a non-zero RETURN\_CODE, execution terminates and the error code is returned in the RETURN\_CODE for the respective call.

The key description file associated with the specified FILE\_TITLE is searched for entries with the specified INDEX\_NUMBER. For each of these, the entry in the data description file (associated with the specified FILE\_TITLE) with a FIELD\_NAME matching the key description file's entry supplies the conversion information for that field.

If the keys to be used are placed into their normal positions in an NPL record, then a "Y" should be passed in the KEY\_IN\_RECORD\$. If an "N" is passed in this value, keys must be placed contiguously into the NPL record starting at byte 1, in the order of the segment number.

This routine must be called only once, at the start of a program, for each conversion table required.

**Pre-Requisites:**

- The catalog file must be open.

**Currency Changes:**

The current record and current index of catalog file are changed by this call.

**Toolbox Features:**

None

**Possible Errors:**

10Invalid file name.

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 access.

115Index number if out of range.

**See Also:**

31020NDM\_CLOSE\_FILE

31040NDM\_CREATE\_CONV\_TABLE\_FOR\_KEY

31150NDM\_GET\_CATALOG\_ENTRY

31340NDM\_OPEN\_FILE

### 1.5.2 NDM\_CREATE\_CONV\_TABLE\_FROM\_CAT

```

31045 NDM_CREATE_CONV_TABLE_FROM_CAT
      (CATALOG_HANDLE$, FILE_TITLE$, FIELD_NAMES$,
      CONVERSION_TABLE_NUMBER, CONVERSION_TABLE$,
      RETURN_CODE)

```

#### Input Parameters:

CATALOG\_HANDLE\$ - the handle to the open catalog file containing the entry for the file whose conversion table is to be created.

FILE\_TITLE\$ - a 32-character mnemonic name for the file, which is matched against the "FILE TITLE" field of the catalog file.

FIELD\_NAMES\$ - a list of field names, as stored in the data description file associated with the specified FILE\_TITLE, to be used in the conversion table. Multiple names are separated by commas. If this parameter is blank, all fields with specified conversion table number are used.

CONVERSION\_TABLE\_NUMBER - the number of the conversion table to create. This value is matched against the "conversion table number" field of the data description records. If names are specified in FIELD\_NAMES\$, this value is ignored.

#### Output Parameters:

CONVERSION\_TABLE\$ - a two-byte handle to the newly created conversion table.

#### Description:

This function reads the conversion information in the data description file associated with the specified FILE\_TITLE\$ into an internal conversion table and returns a handle to it. Only records in the data description file whose "conversion table number" field matches the CONVERSION\_TABLE\_NUMBER parameter or whose names are specified in FIELD\_NAMES\$ (if any), are stored in the conversion table. The resulting CONVERSION\_TABLE\$ handle can then be used in subsequent 31030 NDM\_CONVERT calls.

Calling this routine is equivalent to the following calls:

```

31150 NDM_GET_CATALOG_ENTRY
31340 NDM_OPEN_FILE (Data Description File)
31050 NDM_CREATE_CONVERSION_TABLE
31020 NDM_CLOSE_FILE (Data Description File)

```

If any of the above calls returns a non-zero RETURN\_CODE, execution is terminated and the code is returned in the RETURN\_CODE for this call.

This routine needs to be called for each conversion table required.

**Pre-Requisites:**

- The catalog file must be open.

**Currency Changes:**

The current record and current index of catalog file are changed by this call.

**Toolbox Features:**

None.

**Possible Errors:**

10 Invalid file name.

40 Lost position. The current record of data description file was deleted or its key was modified.

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

43 The data description file is in use.

48 Permission error. Network privileges disallow file access.

**See Also:**

31020NDM\_CLOSE\_FILE

31040NDM\_CREATE\_CONVERSION\_TABLE

31150NDM\_GET\_CATALOG\_ENTRY

31340NDM\_OPEN\_FILE

### 1.5.3 NDM\_OPEN\_FILE\_FROM\_CATALOG

```
'31350 NDM_OPEN_FILE_FROM_CATALOG
      (CATALOG_HANDLE$, FILE_TITLE$, MODE$, INDEX_NUM-
      BER, FILE_HANDLE$, RETURN_CODE)
```

#### Input Parameters:

CATALOG\_HANDLE\$ - the handle to the open catalog file containing the entry for the file to be created.

FILE\_TITLE\$ - a 32-character mnemonic name for the file, which is matched against the "FILE TITLE" field of the catalog file.

MODE\$ - one character:

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

INDEX\_NUMBER - the value used to set the current index for the opened file.

#### Output Parameters:

FILE\_HANDLE\$ - a handle identifying the file that was opened by this call. The size required of a handle may be determined by calling **31230 NDM\_GET\_HANDLE\_SIZE**.

#### Description:

This function opens the data file indicated by the given file title.

A file lock, specified if MODE\$="X", is a "no wait" lock. If the lock request fails, this routine returns immediately with an appropriate code in RETURN\_CODE.

#### Pre-Requisites:

- The catalog file must be open.

#### Currency Changes:

The current record and current index of the catalog file are changed. The current index of the newly opened file is set to the value passed in INDEX\_NUMBER if this corresponds to a valid index (it might not if the "key types" toolbox feature is disabled). The current record is set to "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:**

06 Key value not found in file

11 The file was not found.

24 The file was not created by this native ISAM.

43 The file is in use.

48 Permission error. Network privileges disallow file access.

122 Invalid file type. Unsupported native ISAM specific features are in use.

**See Also:**

31340 NDM\_OPEN\_FILE

### 1.5.4 NDM\_SET\_MAX\_KEY\_SEGS

```
' 31405 NDM_SET_MAX_KEY_SEGS (MAX_KEY_SEGS, RETURN_CODE)
```

**Input Parameters:**

MAX\_KEY\_SEGS - The maximum number of key segments allowed for any file to be opened or created.

**Output Parameters:**

None.

**Description:**

This function sets the maximum number of key segments that any open file may contain. The MAX\_KEY\_SEGS value does not take effect until the next call to 31320 NDM\_INITIALIZE.

**NOTE:** This call may be issued before the 31320 NDM\_INITIALIZE.

**Pre-Requisites:**

None.

**Currency Changes:**

None.

**Toolbox Features:**

This function may only be called if the "file limits" toolbox feature (feature #3) is enabled.

**Possible Errors:**

153 Toolbox file limits not activated.

**See Also:**

31320 NDM\_INITIALIZE  
31420 NDM\_SET\_TOOLBOX\_STATUS

### 1.5.5 NDM\_SET\_READ\_BLOCKING

```
'31415NDM_SET_READ_BLOCKING
      (FILE_HANDLE$, BLOCKING_FACTOR, RETURN_CODE)
```

#### Input Parameters:

FILE\_HANDLE\$ - the handle to the open file to set the blocking factor on.

BLOCKING\_FACTOR - the blocking factor to use for reading several records at a time.

#### Output Parameters:

None.

#### Description:

This function sets up an internal buffer to store a specified number of records. The first time a NDM read call is issued the number of records specified by the BLOCKING\_FACTOR are read and stored in the buffer. Additional NDM read requests attempt to find the requested record in the buffer. If the requested record is not in the buffer, another read of the specified number of records occurs.

**NOTE: No buffering takes occurs when the LOCK\$ parameter on a 31360 NDM\_READ\_RECORD\_BY\_KEY or a 31370 NDM\_READ\_RECORD\_BY\_POSITION is set to "L" (lock the record).**

Depending upon the ISAM in use, large performance gains can be realized by buffering the reads if access to the file is sequential using

**31370 NDM\_READ\_RECORD\_BY\_POSITION** .

If access to the file is by key, using **31360 NDM\_READ\_RECORD\_BY\_KEY** , performance can be increased when using buffering, if the SEARCH\_TYPE\$ parameter is equal to ">", "<", ">=", "<=". However, if too large of a BLOCKING\_FACTOR is used performance may decrease due to the large amount of data that may be read but not used.

**NOTE: Buffering is not used when the SEARCH\_TYPE\$ equals "=".**

The following formula is used to determine the amount of memory (in bytes) that is allocated by NDM when buffering is in use:

MEMORY ALLOCATION = ((RECORD LENGTH + 4 ) \* BLOCKING\_FACTOR) + 8

For example if:

RECORD LENGTH = 150 (in bytes)  
BLOCKING\_FACTOR = 20 (number of records)

Then the amount for memory allocated for buffering is:

$((150 + 4) * 20) + 8 = 3088$  bytes

The higher the blocking factor used, the better the performance gain, but at a cost of memory usage.

Best performance gain can be obtained when the size selected for BLOCKING\_FACTOR allows for the amount for memory allocated for buffering (see formula above) to be as close as possible to the communication block size in use without being greater.

Depending on the operating system in use, increasing the size of the communication block size can gain additional performance. Refer to the native operating system manuals (Novell Netware, MS-DOS, UNIX, etc.) for information on increasing the communication block size.

**Pre-Requisites:**

- The FILE\_HANDLES\$ file must be open.

**Currency Changes:**

None.

**Toolbox Features:**

None.

**Possible Errors:**

None.

**See Also:**

31360 NDM\_READ\_BY\_KEY  
31370 NDM\_READ\_BY\_POSITION

## 1.6 New Return Messages

The following error messages have been added since NDM 1.00:

### **151 - Toolbox feature not enabled for this ISAM**

**Explanation:** A feature that must be enabled with  
31420 NDM\_SET\_TOOLBOX\_STATUS was used.

**Help:** Verify the status of the toolbox with  
**31370 NDM\_GET\_TOOLBOX\_STATUS** and then enable the desired  
feature using **31420 NDM\_SET\_TOOLBOX\_STATUS**.

### **152 - Mismatched field types in conversion table**

**Explanation:** The field class of the NPL and native field types must be the same for all  
corresponding fields in a data description file (i.e., the user can only con-  
vert alpha (A) to alpha, numeric (N) to numeric, and date (D) to date.

**Help:** Check each field in the Data Description field to make sure that the field  
classes for the NPL and Native types match.

### **153 - Toolbox file limits not activated**

**Explanation:** An application tried to extend the "file limits" beyond the minimum  
value. Refer to Section 5.6.3 in the NPL Programmer's Guide for details.

**Help:** Enable feature #3 using 31420 NDM\_SET\_TOOLBOX\_STATUS.

---

## CHAPTER 2

# NDM UTILITY ENHANCEMENTS

### 2.1 Overview

The following section discusses the changes made to the NDM Utilities.

Section 2.2 discusses the utilities that have been updated.

### 2.2 Updated Utilities

The following NDM Utilities have been updated:

- Access Catalog File
  - Access Key Description File
-

- Export Data Dictionary to IQ Utility
- Set NDMUTIL Options
- Display Program Information

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

### **2.2.1 Access Catalog File**

Additional fields have been added to the options displayed on the View Catalog Main Screen. These additional fields are specific to the IBM AS/400 implementation of NDM. Entries can be made into these fields on any NDM supported platform, to allow the information to be available when porting an NDM application to the IBM AS/400. Refer to the NDM IBM AS/400 Platform Specific Addendum for additional information.

### **2.2.2 Access Key Description File**

A new field, Index File Name, has been added to the View Key Description File Screen of this Utility. This entry is used only in conjunction with the index name entries and not the field entries. Entries can be made for this field on all NDM supported platforms, but the entry in this field is only used if the NDM application is ported to the IBM AS/400. This field is specific to the IBM AS/400 implementation of NDM. Refer to the IBM AS/400 Supplement for NDM for additional information.

### **2.2.3 Export Data Dictionary to IQ Utility**

The Export Data Dictionary to IQ Utility has been updated to allow conversion to IQ 3.0 data dictionary files in addition to IQ 2.67 (Niakwa version 1.0) data dictionary files. By using this utility, the developer can immediately export an NDM data file to IQ.

**NOTE:**The IQ development package (either version 2.67 or version 3.0) must be installed on the system where this utility is executed.

The directory where IQ is installed must be on the current path.

There must be sufficient memory available to execute an NPL \$SHELL.

### Features

The features of this utility are the same as documented in Chapter 4 of the NDM Programmer's Guide with the following additions:

- Prompts the user for the version of IQ (2.67 or 3.0) to which the selected data file should be converted.

### General Use

The use of the utility is the same as documented in Chapter 4 of the NDM Programmer's Guide with the following exceptions:

After selecting this utility from the NDM Utilities Main Menu, the user is prompted to select the version of IQ that the data dictionary file should be converted to (i.e., IQ 2.67 (Niakwa version 1.0) and IQ 3.0).

**NOTE:**If IQ version 2.67 is selected, the utility operates as documented in Chapter 4 of the NDM Programmer's Guide.

If IQ version 3.0 is selected, refer to the description below:

### Converting an NDM Data Dictionary To IQ 3.0

IQ 3.0 uses a specially encoded file (the IQ 3.0 DD file) to store descriptions of the file that IQ uses. This file cannot be directly accessed by programs other than IQ 3.0. IQ provides the following two programs to allow access to the IQ 3.0 data dictionary file:

DDUNLOAD	Converts IQ 3.0 DD file to an IQ 3.0 DD text file
DDLLOAD	Converts IQ 3.0 DD text file to an IQ 3.0 DD file

The NDM Convert DD to IQ utility performs the following three steps in order to export an NDM DD file to IQ 3.0:

1. Executes DDUNLOAD to convert the IQ 3.0 DD file into a text file.
2. Export the NDM file by modifying the text file.
3. Executes DDLOAD to convert the text file into an IQ DD file.

**NOTE: DDUNLOAD and DDLOAD are executed by NDMUTIL through a NPL \$SHELL command and should be located in a directory on the user's search path.**

**Under the MS-DOS implementation of NDM there is not enough memory available to execute these programs (DDUNLOAD and DDLOAD) under \$SHELL. It is therefore necessary under the MS-DOS implementation of NDM, for the user to execute DDUNLOAD before and DDLOAD after executing the NDM Utilities (NDMUTIL). Refer to the MS-DOS Platform Specific Addendum for NDM 1.2 for details.**

Once the user has selected the choice of converting a NDM data dictionary to IQ 3.0 follow the steps below:

1. Use the arrow keys to select a data file, whose data dictionary needs to be converted, and press Execute to start the conversion of Cancel to quit.
2. After selecting the data file, a selection box displays. In the selection box enter, the name of the IQ data dictionary file (the default name is DDMASTER.DAT). To select the default press Execute. Otherwise, enter the full path name of IQ data dictionary name to be updated or created

**NOTE: Under the MS-DOS implementation of NDM 1.2 the user is prompted for the name of the data dictionary text file. This only occurs under MS-DOS. Refer to the MS-DOS Platform Specific Addendum for details.**

**If the full path name (providing the location of the IQ directory) is not used, it is necessary to copy the DDMASTER.DAT file to the IQ directory before it can be used by IQ 3.0**

3. Press Execute.

If the NDM data dictionary file is already defined in the IQ data dictionary file, a selection box appears, asking if the user wants to replace the existing IQ definition. Select OK to replace or cancel to retain the existing information.

**NOTE:** Under IQ 3.0, it is no longer a straight forward process for this utility to create IQ categories containing more than one file ( a category in IQ is a grouping to the files that is used to produce the IQ report). In order to have more than one file in a category, it is necessary to specify the relationships between the data fields in the various files. These relationships are called links in IQ. Since this information is not stored in NDM, it is impossible for NDMUTIL to create the IQ category. Instead, only the NDM file information is exported and the user must use the IQ 3.0 DDMAINT program (included with the IQ 3.0 Development software) to create the IQ category and specify the links between the various file contained within the category. This is true even for a single file category (although no links will need to be created). Refer to the IQ 3.0 documentation for instructions on creating categories and links.

#### 2.2.4 Set NDMUTIL Options

This Set Options Screen of this utility has been updated with two additional fields. These fields are:

READ BLOCKING FACTOR

Sets the blocking factor that the NDM Utilities use to read several records at a time. The value used for this field is used in the **31415 NDM\_SET\_READ\_BLOCKING** call. Refer to Section 1.5 for more information on this call.

MAXIMUM KEY SEGMENTS

Sets the maximum number of key segments that any open file used in the NDM Utilities may have. The value used for this field is used in the **31405 NDM\_SET\_MAX\_KEY\_SEGS** call. Refer to Section 1.5 for more information on this call.

**NOTE:**The above features may not work on all supported NDM platforms. Refer to the appropriate NDM Platform-Specific Addendum for details.

### **2.2.5 Display Program Information**

The number of authorized users and NDM serial number fields have been eliminated from this utility.

Refer to the NDM 1.20 Revision Release Notes for the correct version number of the NDMUTIL version and API version fields of this utility.