RESOURCE AND PATIENT MANAGEMENT SYSTEM

# BMXNet20

## ADO.NET Data Access and Connectivity Utilities for RPMS

## (BMX)

## Technical Manual

Version 2.0
March 2007

Office of Information Technology (OIT)
Division of Information Resource Management
Albuquerque, New Mexico

# Preface

The purpose of this manual is to provide technical information about the *BMXNet ADO.NET Data Access and Connectivity Utilities for RPMS* (BMX) package. The BMX package is designed to support connectivity and data exchange between .NET applications and RPMS.

# Change History

The following changes have been made to this document since the last publication.

| Date of Change | Location of Revision | Revision |
|---|---|---|
| 11/06/06 | 3.1 | Added routine BMXPO; associated updates |
| 11/06/06 | 4.1, 4.2 | Added file BMX GUI REPORTS; associated updates |
| 03/07/07<br>3/14/07 | 2.1 | Changed system requirements from “Cache 5.1” to 5.0.21 or higher<br>Requirement: Windows 2000, 2003, XP with Service Packs installed |
| 03/07/07 | 6.4 | Added Client Components information |
| | | |

# TABLE OF CONTENTS

# 1.0 Introduction

BMXNet20 represents a rewrite of BMXNet to support Microsoft’s .*Net* framework 2.0. This manual provides IHS site managers with a technical description of the BMXNet routines, files, menus, cross references, globals, and other necessary information required to effectively manage the system. BMX Logging is introduced in Version 2.0 along with minor bug fixes (see the *BMXNet ADO.NET Data Access and Connectivity Utilities for RPMS* User Manual for additional information). File access to the BMX MONITOR file has been changed to the site manager access code.

All routines, files, options, and keys are namespaced starting with the letters “BMX.” The file number range for this package is 90093.1 – 90093.99.

# 2.0 Implementation and Maintenance

See the Installation Instructions section in the *Installation Guide* for maintenance and implementation information.

## 2.1 System Requirements

1. VA FileMan version 22 or higher
2. VA Kernel version 8.0 or higher
3. Cache version 5.0.21 or higher
4. Windows 2000, 2003, XP with Service Packs installed

# 3.0 Routines

## 3.1 Routines with Descriptions

The following table details routines and their associated descriptions. There are a total of 60 routines.

| Routine | Description |
|---|---|
| BMXADE1 | IHS/OIT/HMW - BMXNet ADO.NET PROVIDER ; |
| BMXADE2 | IHS/OIT/HMW - BMXNet ADO.NET PROVIDER ; |
| BMXADO | IHS/CIHA/GIS - RPC CALL: GENERATE AN ADO SCHEMA STRING AND DATA SET |
| BMXADO2 | IHS/CIHA/GIS - BMX ADO RECORDSET UTILS |
| BMXADOF | IHS/CIHA/GIS - RPC CALL FOR EXTENDED FUNCTIONALITY OF BMXNet UTILITIES |
| BMXADOF1 | IHS/CIHA/GIS - RPC CALL FOR EXTENDED FUNCTIONALITY OF BMXNet UTILITIES |
| BMXADOF2 | IHS/CIHA/GIS - RPC CALL FOR EXTENDED FUNCTIONALITY OF BMXNet UTILITIES |
| BMXADOFD | IHS/CIHA/GIS - RPC CALL FOR EXTENDED FUNCTIONALITY OF BMXNet UTILITIES |
| BMXADOFS | IHS/CIHA/GIS - RPC CALL FOR EXTENDED FUNCTIONALITY OF BMXNet UTILITIES |
| BMXADOI | IHS/CIHA/GIS - RPC CALL: GENERATE DATA FOR AN ADO DATASET |
| BMXADOL | CIHA/GIS - LOG FILER PARAMETERS: FILE, DAS, DATA |
| BMXADOS | IHS/CIHA/GIS - UPDATE THE BMX ADO SCHEMA FILE |
| BMXADOS1 | IHS/CIHA/GIS - UPDATE THE BMX ADO SCHEMA FILE GUI VERSION |
| BMXADOV | IHS/CIHA/GIS - RPC CALL: GENERATE DATA FOR AN ADO DATASET |
| BMXADOV1 | IHS/CIHA/GIS - RPC CALL: GENERATE DATA FOR AN ADO DATASET |
| BMXADOV2 | IHS/CIHA/GIS - RPC CALL: GENERATE DATA FOR AN ADO DATASET |
| BMXADOVJ | IHS/CIHA/GIS - RPC CALL: GENERATE DATA FOR AN ADO DATASET |
| BMXADOX | IHS/CIHA/GIS - RPC CALL: GENERATE AN ADO SCHEMA STRING AND DATA SET |
| BMXADOX1 | IHS/CIHA/GIS - RPC CALL: GENERATE AN ADO SCHEMA STRING AND DATA SET |
| BMXADOX2 | IHS/CIHA/GIS - RPC CALL: GENERATE AN ADO SCHEMA STRING AND DATA SET |
| BMXADOXX | IHS/CIHA/GIS - RPC CALL: GENERATE AN ADO SCHEMA STRING AND DATA SET |
| BMXADOXY | IHS/CIHA/GIS - RPC CALL: GENERATE AN ADO SCHEMA STRING AND DATA SET |
| BMXE01 | IHS/OIT/FJE - ENVIRONMENT CHECK FOR BMX 2.0 ; |

| Routine | Description |
|---|---|
| BMXFIND | IHS/OIT/HMW - BMX GENERIC FIND |
| BMXG | IHS/OIT/HMW - UTIL: GET DATA |
| BMXGETS | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXMBRK | IHS/OIT/HMW - BMXNet MONITOR |
| BMXMBRK2 | IHS/OIT/HMW - BMXNet MONITOR |
| BMXMEVN | IHS/OIT/HMW - BMXNet MONITOR |
| BMXMON | IHS/OIT/HMW - BMXNet MONITOR |
| BMXMSEC | IHS/OIT/HMW - BMXNet MONITOR |
| BMXNTEG | INTEGRITY CHECKER;OCT 31, 2006 |
| BMXPO | IHS/CMI/MAW - Populate appcontext with all namespaced RPCs |
| BMXPRS | IHS/OIT/HMW - BMX WINDOWS UTILS |
| BMXRPC | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXRPC1 | IHS/OIT/HMW - UTIL: REMOTE PROCEDURE CALLS |
| BMXRPC2 | IHS/OIT/HMW - FIELD LIST |
| BMXRPC3 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXRPC4 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXRPC5 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXRPC6 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXRPC7 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXRPC8 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXRPC9 | IHS/OIT/HMW - RPC CALL FOR EXTENDED BROKER FUNCTIONALITY |
| BMXSQL | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXSQL1 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXSQL2 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXSQL3 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXSQL4 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXSQL5 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXSQL6 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXSQL7 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXSQL91 | IHS/OIT/HMW - BMX REMOTE PROCEDURE CALLS |
| BMXTABLE | IHS/OIT/HMW - BMX RETURN ENTIRE TABLE |
| BMXTRS | IHS/OIT/HMW - UPPERCASE-LOWERCASE |
| BMXUTL1 | IHS/OIT/HMW - UTIL: PATIENT DEMOGRAPHICS |
| BMXUTL2 | IHS/OIT/HMW - UTIL: PATIENT INFO |
| BMXUTL5 | IHS/OIT/HMW - DATE FORMAT |
| BMXUTL6 | IHS/OIT/HMW - BMXNET INSTALLATION CALLS |
| BMXUTL7 | IHS/OIT/HMW - BMXNET INSTALLATION CALLS |

Figure 3-1: Table of routines and descriptions

# 4.0 Files and Tables

The following tables include information regarding file lists and file access.

## 4.1 File List

The following table details file numbers and names.

| File Number | File Name |
|---|---|
| 90093.1 | BMX USER |
| 90093.2 | BMX APPLICATION |
| 90093.5 | BMXNET MONITOR |
| 90093.9 | BMX GUI REPORT |
| 90093.98 | BMX ADO LOG |
| 90093.99 | BMX ADO SCHEMA |

Figure 4-1: Table of file list information

## 4.2 File Access

The following table lists file access information.

| FILENAME | FILE (#) | GL | RD | WR | LYG | DD | DEL |
|---|---|---|---|---|---|---|---|
| BMX USER | 90093.1 | ^BMXUSER( | @ | @ | @ | @ | @ |
| BMX APPLICATION | 90093.2 | ^BMXAPPL( | @ | @ | @ | @ | @ |
| BMXNET MONITOR | 90093.5 | ^BMXMON( | # | # | # | @ | # |
| BMX GUI REPORT | 90093.9 | ^BMXGUIR( | @ | @ | @ | @ | @ |
| BMX ADO LOG | 90093.98 | ^BMXADOL( | @ | @ | @ | @ | @ |
| BMX ADO SCHEMA | 90093.99 | ^BMXADO( | @ | @ | @ | @ | @ |

Figure 4-2: Table of file access information

## 4.3 Cross References

The BMX files all use the standard “B” cross reference as defined by FileMan. One additional cross reference is used in the BMX GUI REPORT file. The PACKAGE field has a MUMPS cross reference to identify the package specifics of the report. The “AUSRP” cross reference identifies the cross reference as: ^BMXGUIR(“AUSRP”, User Pointer, Package Pointer, Inverse Start Date/Time, Record IEN). The following example shows how cross references may appear on your computer screen.

```
                                FILE: BMX USER
                               GLOBAL: ^BMXUSER(
                                FILE #: 90093.1

FIELD #      FIELD NAME                       SUBSCRIPT              PIECE TYPE
===============================================================================

.01          WINIDENT                         D0,0                      1    F
                X: B
                    1)   S ^BMXUSER("B",$E(X,1,30),DA)=""
.02          USER                               "                       2    P
.03          V ENCRYPTED                        "                       3    F


                              FILE: BMX APPLICATION
                                GLOBAL: ^BMXAPPL(
                                FILE #: 90093.2

FIELD #      FIELD NAME                       SUBSCRIPT              PIECE TYPE
===============================================================================

.01          MAJOR VERSION                    D0,0                      1    F
                X: B
                    1)   S ^BMXAPPL("B",$E(X,1,30),DA)=""
.02          MINOR VERSION                      "                       2    F
.03          BUILD                              "                       3    D


                              FILE: BMXNET MONITOR
                                GLOBAL: ^BMXMON(
                                FILE #: 90093.5

FIELD #      FIELD NAME                       SUBSCRIPT              PIECE TYPE
===============================================================================

.01          PORT                             D0,0                      1    N
                X: B
                    1)   S ^BMXMON("B",$E(X,1,30),DA)=""
.02          ENABLED                            "                       2    S
.03          INTEGRATED SECURITY                "                       3    S
.04          SESSION NAMESPACE                  "                       4    F
```

This example (Figure 4-3) continues on the following page.

```
                              FILE: BMX GUI REPORT
                               GLOBAL: ^BMXGUIR(
                                FILE #: 90093.9

FIELD #      FIELD NAME                       SUBSCRIPT              PIECE TYPE
===============================================================================

.01          NAME                             D0,0                      1    F
                X: B
                    1)   S ^BMXGUIR("B",$E(X,1,30),DA)=""
.02          USER WHO REQUESTED REPORT          "                       2    P
.03          START TIME                         "                       3    D
.04          END TIME                           "                       4    D
.05          OUTPUT TYPE                        "                       5    S
.06          OPTION NAME                        "                       6    F
.07          REPORT STATUS                      "                       7    S
.08          PACKAGE                            "                       8    P
                X: AUSRP^MUMPS
                    1)   S ^BMXGUIR("AUSRP",$P(^BMXGUIR(DA,0),"^",2),X,(9999999.
9999-$P(^BMXGUIR(DA,0),"^",3)),DA)=""
                    2)   K ^BMXGUIR("AUSRP",$P(^BMXGUIR(DA,0),"^",2),X,(9999999.
9999-$P(^BMXGUIR(DA,0),"^",3)),DA)

1100         OUTPUT  (90093.911)
  .01          OUTPUT                         D0,11,D1,0                1    W



                               FILE: BMX ADO LOG
                               GLOBAL: ^BMXADOL(
                                FILE #: 90093.98

FIELD #      FIELD NAME                       SUBSCRIPT              PIECE TYPE
===============================================================================

.01          TRANSACTION TIMESTAMP            D0,0                      1    D
                X: B
                    1)   S ^BMXADOL("B",$E(X,1,30),DA)=""
.02          FILE NUMBER                        "                       2    F
.03          DAS                                "                       3    F

1            DATA  (90093.981)
  .01          DATA                           D0,1,D1,0                 1    W
```

This example (Figure 4-3) continues on the following page.

```
                              FILE: BMX ADO SCHEMA
                               GLOBAL: ^BMXADO(
                               FILE #: 90093.99

FIELD #      FIELD NAME                       SUBSCRIPT             PIECE TYPE
==============================================================================

.01          SCHEMA NAME                      D0,0                     1    F
                X: B
                    1)   S ^BMXADO("B",$E(X,1,30),DA)=""
.02          FILE OR SUBFILE NUMBER             "                      2    N
.03          DATASET IS READ ONLY               "                      3    S

1            FIELD NUMBER  (90093.991)
  .01          FIELD NUMBER                   D0,1,D1,0                1    F
                X: B
                    1)   S ^BMXADO(DA(1),1,"B",$E(X,1,30),DA)=""
  .02          DATA TYPE                        "                      2    S
  .03          FIELD LENGTH                     "                      3    F
  .04          COLUMN HEADER                    "                      4    F
  .05          READ ONLY                        "                      5    S
  .06          KEY FIELD                        "                      6    S
  .07          NULL ALLOWED                     "                      7    S
  .08          IEN AUTOMATICALLY INCLUDED       "                      8    S
  .09          ALWAYS GET INTERNAL VALUE        "                      9    S
  1            AUTO IDENTIFIER EXTR FUNCT     D0,1,D1,1             E1,240   F
  2            SPECIAL UPDATE EP              D0,1,D1,2             E1,245   F
  3            EXTR FUNCT FOR TRIGGERED VALU  D0,1,D1,3             E1,245   F
               E

2            VIEW  (90093.992)
  .01          VIEW                           D0,2,D1,0                1    F
                X: B
                    1)   S ^BMXADO(DA(1),2,"B",$E(X,1,30),DA)=""
  1            ENTRY POINT                    D0,2,D1,1             E1,240   F

  2            PARAMETER  (90093.9922)
    .01          PARAMETER                    D0,2,D1,2,D2,0           1    F
                X: B
                    1)   S ^BMXADO(DA(2),2,DA(1),2,"B",$E(X,1,30),DA)=""
    .02          BRIEF DESCRIPTION              "                      2    F

  3            DESCRIPTION  (90093.9923)
    .01          DESCRIPTION                  D0,2,D1,3,D2,0           1    W
```

Figure 4-3: Sample of cross reference screens

## 4.4 Table File

Please see the following example screens of a Table File.

```
                                FILE: BMX USER
                              GLOBAL: ^BMXUSER(
                               FILE #: 90093.1

FIELD #      FIELD NAME                        SUBSCRIPT              PIECE TYPE
===============================================================================

.01          WINIDENT                          D0,0                     1    F
.02          USER                                "                      2    P
.03          V ENCRYPTED                         "                      3    F

                             FILE: BMX APPLICATION
                              GLOBAL: ^BMXAPPL(
                               FILE #: 90093.2

FIELD #      FIELD NAME                        SUBSCRIPT              PIECE TYPE
===============================================================================

.01          MAJOR VERSION                     D0,0                     1    F
.02          MINOR VERSION                       "                      2    F
.03          BUILD                               "                      3    D

                              FILE: BMXNET MONITOR
                               GLOBAL: ^BMXMON(
                               FILE #: 90093.5

FIELD #      FIELD NAME                        SUBSCRIPT              PIECE TYPE
===============================================================================

.01          PORT                              D0,0                     1    N
.02          ENABLED                             "                      2    S
.03          INTEGRATED SECURITY                 "                      3    S
.04          SESSION NAMESPACE                   "                      4    F

                              FILE: BMX ADO SCHEMA
                               GLOBAL: ^BMXADO(
                               FILE #: 90093.99

FIELD #      FIELD NAME                        SUBSCRIPT              PIECE TYPE
===============================================================================

.01          SCHEMA NAME                       D0,0                     1    F
.02          FILE OR SUBFILE NUMBER              "                      2    N
.03          DATASET IS READ ONLY                "                      3    S

1            FIELD NUMBER  (90093.991)
  .01          FIELD NUMBER                    D0,1,D1,0                1    F
  .02          DATA TYPE                         "                      2    S
  .03          FIELD LENGTH                      "                      3    F
  .04          COLUMN HEADER                     "                      4    F
  .05          READ ONLY                         "                      5    S
  .06          KEY FIELD                         "                      6    S
  .07          NULL ALLOWED                     "                       7    S
  .08          IEN AUTOMATICALLY INCLUDED       "                       8    S
  .09          ALWAYS GET INTERNAL VALUE        "                       9    S
  1            AUTO IDENTIFIER EXTR FUNCT     D0,1,D1,1             E1,240   F
  2            SPECIAL UPDATE EP              D0,1,D1,2             E1,245   F
  3            EXTR FUNCT FOR TRIGGERED VALU  D0,1,D1,3             E1,245   F
               E

2            VIEW  (90093.992)
  .01          VIEW                           D0,2,D1,0                 1    F
  1            ENTRY POINT                    D0,2,D1,1             E1,240   F

  2            PARAMETER  (90093.9922)
    .01          PARAMETER                    D0,2,D1,2,D2,0            1    F
    .02          BRIEF DESCRIPTION              "                       2    F

  3            DESCRIPTION  (90093.9923)
    .01          DESCRIPTION                  D0,2,D1,3,D2,0            1    W
  2            SPECIAL UPDATE EP              D0,1,D1,2             E1,245   F
```

Figure 4-4: Sample of table file screens

## 4.5 Client Assembly Information

One assembly is furnished with this application. This assembly is found on the Windows Client computer. The attributes can be found by highlighting the assembly *BMXNet20.dll* and right clicking on the file. The following information represents the released version of the assembly and can be found under the Version tab of the Properties menu.

| **BMXNet20.dll Version and Assembly Information** | |
|---|---|
| Assembly Version | 2.0.2459.21970 |
| Comments | ADO.NET 2.0 Data Provider for RPMS |
| Company | Indian Health Service (IHS) |
| File Version | 2.0.0.0 |
| Internal Name | BMXNet20.dll |
| Language | Language Neutral |
| Original File Name | (Intentionally left blank) |
| Product Version | 2.0.0.0 |

Figure 4-5: Table of assembly information

# 5.0 Internal Relations

There are no internal relations in the BMXNet package.

# 6.0 External Relations

## 6.1 External Calls

There are no external calls in the BMXNet package.

## 6.2 Published Entry Points

There are no published entry points in BMXNet.

## 6.3 Exported Options

Please see the following table for option names and descriptions.

| Option Name | Description |
|---|---|
| BMX MONITOR EDIT | Blank Community/City Report |
| BMX MONITOR START | Use this option to start or restart all BMXNet monitors in the BMXNET MONITOR file.<br>This option should be scheduled as a STARTUP type option in TaskMan. Do not use this option to start a specific monitor. To do this, in programmer mode, do STRT^BMXMON(PORT). See the product documentation for instructions on how to start session monitors in a particular namespace and on how to enable or disable Windows Integrated Security. |
| BMX MONITOR STOP | Use this option to stop all BMXNet monitors. |
| BMXMENU | Menu contains options in the BMX namespace |
| BMXRPC | All BMXNet users must have access to this option. |

Figure 6-1: Table of exported options information

## 6.4 Client Components

Only BMXNet20.dll is exported in the package for the client. BMXNet20.dll is a dotNet 2.0 object that is used with the development of client applications. The *dll* has been *dotfuscated* and source code is protected. The object is written in C# using Visual Studio 2005 IDE. There are no additional dependencies. There are no third party controls used to create BMXNet20.dll.

**Visual Studio 2005 IDE**

- Microsoft Visual Studio 2005 Version 8.0.50727.762 (SP.050727-7600)
- Microsoft .NET Framework Version 2.0.50727
- Installed Edition: Enterprise
- Microsoft Visual C# 2005 77642-113-3000004-41007
- Microsoft Visual C# 2005
- Microsoft Visual Studio Tools for Office 77642-113-3000004-41007
- Microsoft Visual Studio Tools for the Microsoft Office System
- Visual Studio 2005 Team Edition for Developers 77642-113-3000004-41007
- Microsoft Visual Studio 2005 Team Edition for Software Developers
- Crystal Reports AAC60-G0CSA4B-V7000AY
- Crystal Reports for Visual Studio 2005
- Enterprise Library Configuration Editor 3.0 Jan 2007 CTP
- Microsoft patterns & practices Enterprise Library Configuration Editor
- InstallShield 12 Express Editor
- InstallShield 12 Express Editor.
    - For more information see the Macrovision website at <http://www.macrovision.com>. Copyright 2005 Macrovision Corporation.
- Microsoft Visual Studio 2005 Team Edition for Software Developers - ENU Service Pack 1 (KB926601)
- This service pack is for Microsoft Visual Studio 2005 Team Edition for Software Developers - ENU.
    - http://support.microsoft.com/kb/926601
- CodeKeep 2.0 (VS 2005)
    - http://www.codekeep.net. Copyright (c) 2006 Arcware, Inc.
- VSNETcodePrint2005
    - http://www.StarPrint limited.com Copyright (c) 2005 StarPrint limited Inc.

**BMXNet20.dll**

- Machine Intel 386
- Characteristics
    - The file is a dynamic-link library
    - Line numbers stripped from file
    - Local symbols stripped from file
    - Size Of Code 81,920 Bytes
- Operating System 32-bit Windows
- File Type: Dynamic Link Library

# 7.0 Security Keys

Security key information includes the key name and description as shown in the following table.

| Key Name | Description |
|---|---|
| BMXZMENU | All BMXNet managers must have access to this option. |

Figure 7-1: Table of security key information

# 8.0 Archiving and Purging

No archiving or purging is necessary with BMXNet.

# 9.0 Generating Online Documentation

This section describes a few methods to generate BMXNet system technical documentation. Online BMXNet software technical documentation, in addition to that which is located in the help prompts throughout the BMXNet package, can be generated through the use of several Kernel options. These include, but are not limited to, the following:

- %INDEX
- Menu Management
- Inquire Option
- Print Option File
- VA FileMan
- Data Dictionary Utilities
- List File Attributes

Typing a question mark **?** at the "`Select . . . Option`" prompt will provide online technical information.

- A single question mark (**?**) lists all options that can be accessed from the current option.
- Two question marks (**??**) list all options accessible from the current one, showing the formal name and lock for each.
- Three question marks (**???**) display a brief description for each option in a menu.
- An option name preceded by a question mark (**?OPTION**) shows extended help, if available, for that option.

For a more exhaustive option listing and further information about other utilities that supply online technical information, consult the DHCP Kernel Reference manual.

## 9.1 %INDEX

This option analyzes the structure of a routine to determine in part if the routine adheres to RPMS programming standards. The %INDEX output can include the following components:

- Compiled list of errors and warnings
- Routine listing
- Local variables
- Global variables
- Naked globals

- Label references
- External references

Running %INDEX for a specified set of routines, allows users to discover any deviations from RPMS programming standards that exist in the selected routines and to see how routines interact with one another (i.e., which routines call or are called by other routines).

To run %INDEX for the BMXNet package, type the BMX namespace at the Routine(s)?> prompt.

## 9.2 Inquire Option

This menu management option provides the following information about a specified option:

- Option name
- Menu text
- Option description
- Type of option
- Lock (if any)

In addition, all items on the menu are listed for each menu option. To secure information about BMXNet options, specify the BMX namespace.

## 9.3 Print Option File

This utility generates a listing of options from the Option file (#19). Users can choose to print all of the entries in this file or specify a single option or range of options. For a list of BMXNet options, refer to the Exported Options section of this manual.

## 9.4 List File Attributes

This VA FileMan option allows users to generate documentation pertaining to files and file structure. Using the standard format of this option yields the following data dictionary information for a specified file:

- File name and description
- Identifiers
- Cross-references
- Files pointed to by the file specified

- Files that point to the file specified
- Input, print, and sort templates

In addition, the following applicable data is supplied for each field in the file:

- Field name, number, title, and description
- Global location
- Help prompt
- Cross-references
- Input transform
- Date last edited
- Notes

Using the Global Map format of this option generates an output that lists the following information:

- All cross-references for the file selected
- Global location of each field in the file
- Input, print, and sort templates

For a comprehensive listing of BMXNet package files, please refer to the Files section of this manual.

# 10.0 SAC Requirements and Exemptions

Applicable standards are listed in the following table:

| Standard Number | Description |
|---|---|
| 2.2.3.1.4 | SET DUZ |
| 2.2.3.2.5.1 | Use of ^TMP($J |
| 2.2.3.3.2 | Intrinsic Variables direct use is prohibited |
| 2.2.4.3 | CLOSE, direct use prohibited |
| 2.2.4.4 | HALT command |
| 2.2.4.5 | JOB, direct use prohibited |
| 2.2.4.9 | OPEN, is prohibited |

Figure 10-1: Table of standards and descriptions

**Reasons for Exemption**

BMXNet is a system level utility which mediates connections between .*NET* applications and the RPMS environment. As such, BMXNet requires the ability to directly *open* and *close* TCP/IP socket connections, access intrinsic variables, manipulate DUZ and its descendants, and directly invoke the JOB and HALT commands. The ^*TMP(* is used without $J subscript to temporarily store data to support asynchronous RPC calls which span jobs.

# 11.0 Glossary

| Term | Definition |
|---|---|
| ADO.NET | Set of classes that expose data access services to the .NET programmer. |
| API | Application Program Interface. Callable entry points that enable software to communicate with other software. |
| Application Context | The Application Context is a read/write part of application programs which contains rules dictating how the application interacts with other applications and users. |
| BMXNet | A set of software utilities designed to connect to RPMS data by .NET applications. |
| column_alias | Specifies an alternative name to replace the column name in the query result set. For example, an alias such as “Quantity,” “Quantity to Date,” or “Qty” can be specified for a column named **quantity** |
| field_name | Specifies from which field (column) the FROM clause should return. The field_name is associated with the preceding table_name by using a dot (.) between the two. |
| FROM Keyword | Specifies that the system should return the values from the following table_list statement. |
| INDEX Keyword | Identifies a specific FileMan cross-reference to use when retrieving data. Always use the SHOWPLAN keyword in conjunction with the INDEX keyword to ensure that you achieve the intended result. |
| Index Row | Part of the SHOWPLAN results that includes the M code created by BMXNet that will execute on the RPMS server and iterate through the FileMan file. |
| Iterate | To say or perform again; repeat. |
| join_type | Specifies a join using nonstandard syntax and the WHERE clause. The =* operator is used to specify a One-to-many (OTM) join. Use the OTM join to express relationship between tables A and B such that a record in table A can be referenced by a FileMan pointer field in one or more records in table B. |
| Log on | Connect to a network |
| M | A programming language that originated from the medical sector but is currently used in a variety of database applications due to its retrieval capabilities. |
| M Routines | A collection of command lines, all associated with a single name that can be stored and retrieved as a unit. |

| Term | Definition |
|---|---|
| MAXRECORDS Keyword | Specifies the maximum number of records to return |
| Operator | A symbol that specifies which operation the systems should perform relative to the indicated operator arguments. |
| Operator Arguments | A value or expression dictating the information upon which the related operator acts. |
| Overloads | The creation of more than one procedure, instance constructor, or property in a class with the same name but different argument types. Overloading is especially useful when your object model dictates that you employ identical names for procedures that operate on different data types |
| Parameter | A value given to a variable until the related operation is completed. Parameters are treated by the system as constants. Parameters are often used to customize a program for a particular purpose |
| Port | Software that links one computer with another using TCP/IP address and port numbers assigned by network administrators. |
| primary_table_name | The file from which the system should first retrieve data. |
| related_table_name | The file from which the system should receive data that matches the primary table. |
| Remote Procedure Call (RPC) | A technique used to constructing distributed, client-server based applications. RPC extends the capabilities of local procedure calling, so that the called procedure need not exist in the same address space as the calling procedure. |
| Screen Rows | Part of the SHOWPLAN results that includes the M code that BMXNet will execute to filter the results. The “^” character in the query plan is replaced by the “~” character. |
| search_condition | Restricts the rows returned in the result set by using predicates. There is no limit to the number of predicates that can be included in a search condition. |
| SELECT Keyword | Specifies that the system should return the values by the following select_list statement. |
| select_list | Specifies which fields (columns) the system should select for the returned set. The select_list a series of expressions separated by commas. |
| Server | A computer that hosts RPMS applications. |

| Term | Definition |
|---|---|
| SHOWPLAN Keyword | Returns the query plan, including the M code that will be executed to retrieve the records. |
| table_name | Specifies from which files (tables) the FROM clause should return fields. |
| table_source | Specifies from which files (tables) the FROM clause should return fields. |
| WHERE Keyword | Specifies a search condition to restrict the rows returned. |

# 12.0 Contact Information

If you have any questions or comments regarding this distribution, please contact the OIT Help Desk by:

| | |
|---|---|
| **Phone**: | (505) 248-4371 or |
| | (888) 830-7280 |
| **Fax**: | (505) 248-4199 |
| **Web**: | http://www.ihs.gov/GeneralWeb/HelpCenter/Helpdesk/index.cfm |
| **Email**: | ITSCHelp@mail.ihs.gov |