MAX DB2/UTIL

MAX DB2/UTIL V3.4.0

User Reference Manual

MAX SOFTWARE™
Denver, Colorado
MAXimizing your performance

CA, INC.

Contact Information

Corporate Headquarters

CA, Inc.
One CA Plaza
Islandia, NY 11749
USA

www.ca.com

Technical Support

For online technical assistance and a complete list of locations, primary service hours, and telephone numbers, contact Technical Support at http://ca.com/support

Web Site

http://ca.com/support

Disclaimer

Disclaimer of Warranties and Limitation of Liabilities

The staff of MAX Software has taken due care in preparing this manual; however, nothing contained herein modifies or alters in any way the standard terms and conditions of the MAX Software, Inc. standard software evaluation agreement; purchase agreement; lease agreement; or rental agreement by which this software was acquired; nor increases in any way MAX Software's liability to the customer. In no event shall MAX Software be liable for incidental or consequential damages in connection with or arising from the use of this manual or any program contained herein.

Release 3.4.0 (November 2005)

Copyright (c) MAX Software, Inc. 1993 - 2005.

All Rights Reserved Licensed Material Unauthorized Duplication Prohibited. This manual contains confidential material protected by license agreements.

MVS, DB2, IMS, REXX, TSO/E, and ISPF are software products of International Business Machines Corporation.

MAX MVS/UTIL (a complete set of data file manipulation tools with the following 3 components: MAX Data/Util, MAX/PDF and MAX/Batch); MAX IMS/UTIL (a complete set of IMS database manipulation tools with the following 2 components: MAX/IMS Online and MAX/IMS Batch); MAX DB2/UTIL (a complete set of DB2 database manipulation tools); and MAX/REXX (an interface between REXX and VSAM, SAM, PDS and DB2 data) are trademarks of MAX Software, Inc.

TABLE OF CONTENTS

LIST OF FIGURES

REVISIONS

Release 3.4.0: November 2005

- Support for new DB2 V8 function.

Release 3.3.0: March 2004

- Relational Unload/Reload.
- Unloaded and formatted table data can be created in various EBCDIC, ASCII and Unicode code pages.
- **CODEPAGE** option of the batch SQL processor's **SELECT INTO** command.

Release 3.2.0: March 2003

- New batch SQL processor.
- Unload and reload of table data.
- Export of SQL data to XML or other formats using Data Transformation.
- On-line submission of RUNSTATS utility job.
- Support for DB2 version 8 long table and column names.
- CUT and PASTE rows may be stored in a sequential data set or UNIX file in addition to the ISPF "CUT Buffer".
- SQL program statement generation for COBOL, PL/I, C, C++, assembler or MAX/REXX.
- Uppercase/mixed case editing by column.
- PROFILE option to limit the number of PASTE rows.
- PROFILE option to override the DB2 PLAN name.
- Data privacy functions **SCRAMBLE**, **TRANSLATE** and **UNSCRAMBLE**.

Release 3.1.0: June 2002

- Initial release.

PREFACE

This book provides a guide and reference about the various functions of MAX DB2/UTIL. Use this book to learn and use the MAX DB2/UTIL product.

It describes:

- An introduction to the product.
- Guideline information on data functions.
- Command syntax and descriptions.
- Command operand syntax along with a description of the Operands.
- Numerous examples.
- Return code information.

Who Should Read This Book

This book is for programmers, database administrators, system programmers, or other technical persons who perform data manipulation of DB2 databases.

Database manipulation includes DB2 table query selection, browse, modification, and printing. Users are expected to have knowledge of DB2.

Notational Conventions

The following notational conventions are used in this manual:

- Uppercase commands and their operand(s) should be entered as shown but need not be in uppercase.
- Operand(s) shown in lower case are variables and a value should be substituted for them.
- Operand(s) shown in brackets [ ] are optional, with choices indicated by a vertical bar |. One or none may be chosen; the defaults are underscored.
- Operand(s) shown in braces { } are alternatives; one must be chosen.
- An ellipsis (...) indicates that the parameter shown may be repeated to specify additional items of the same category.

How This Book is Organized

This book contains the following chapters:

Chapter 1: Introduction to MAX DB2/UTIL
describes the need for MAX DB2/UTIL and its overall uses.

Chapter 2: How to Use MAX DB2/Util
provides an overview of how the online product functions.

Chapter 3: SQL Command Utility
describes the MAX DB2/UTIL batch SQL utility used to process SQL commands, load DB2 tables and unload SQL data in a variety of formats.

Chapter 4: Data Privacy Functions
describes how to use MAX DB2/UTIL to ensure data privacy while allowing access to live data.

Chapter 5: Data Set Name List Functions
provides an overview of how to access DB2 tables with MAX DB2/UTIL from DSNL entries.

CHAPTER 1: INTRODUCTION TO MAX DB2/UTIL

Database Maintainability

DB2 is the premier relational database management system for the IBM computing environment. As a relational database, DB2 provides ease of data access not found in other database architectures. DB2 is usually the preferred choice for managing traditional mainframe application data and newer web-oriented application data.

With DB2 playing such a key role in many companies' computing environment, the presence of cost-effective tools that facilitate the edit and test of databases is essential. A set of easy to use application tools for the efficient support, change and test lifecycles of DB2 databases is exactly what MAX DB2/UTIL has been designed to provide.

Namely:

- Tools for the online manipulation of DB2 databases.
- ISPF-like commands that programmers of all skill levels can learn.
- Powerful features to bring meaning to the analysis and manipulation of DB2 database contents.
- Easy access to all DB2 data types.
- Special attention to DB2 database integrity requirements.

Table Editor

MAX DB2/UTIL provides online Browse and Edit facilities so that DB2 tables can be browsed and edited with the same ease that programmers browse and edit text, VSAM and large sequential type data files.

The MAX DB2/UTIL Browse and Edit facilities have the same look and feel as MAX Data/Util, MAX IMS/UTIL, and ISPF text browse and edit facilities.

DB2 tables differ considerably from text and VSAM files in that:

- DB2 data within tables is contained within predefined columns with predefined data types. Special catalog tables containing metadata are automatically maintained by DB2 to describe customer-defined data. No copybooks are required for data typing.
- DB2 data has no inherent sequence. Any required ordering must be requested when the data is accessed.
- DB2 data values can take on the special NULL value, which is distinct from all other possible values.
- DB2 data is managed through separate DB2 subsystems with special storage organizations and processes to enhance performance and to control access.
- DB2 provides recovery by logging all changes to DB2 data and backing out changes as needed. DB2 allows concurrent access by providing an automatic locking protocol.
- DB2 can automatically enforce data integrity through predefined rules, relationships and triggered actions.

Despite these differences, programmers still want to process DB2 data with the same ease as they process their text and VSAM files now. The MAX DB2/UTIL Browse and Edit facilities were specially designed to accommodate these differences and shield the user from DB2 complexities.

MAX DB2/UTIL has two options for browsing and editing DB2 tables:

- A Formatted Editor with the table columns displayed vertically for a single result row at a time.
- A Horizontal Editor with the table columns presented horizontally for data from multiple result rows displayed below the corresponding column names.

The MAX DB2/UTIL table Browse and Edit facilities include the following features that programmers are already familiar with having used ISPF text, MAX Data/Util and MAX IMS/UTIL products.

- PF (program function) key scrolling.
- **FIND** and **CHANGE** commands.
- Hex mode displays.
- Line commands for deleting, inserting and repeating.
- **CUT** and **PASTE** commands.

The following functions were added to accommodate the special needs of DB2 tables:

- Processing all DB2 data types, including null values, large objects and automatically generated values.
- Invoking Browse and Edit interchangeably from any place in a result set without losing position.
- Scrolling forward and backward through a result set.
- Logging for backing out changes of an Edit session.

The Formatted Browse and Edit facilities offer all of the same commands and capabilities as the Horizontal Browse and Edit facilities with the exception that one result row is presented at a time allowing the overtyping of column contents. Result rows can generally be scrolled in a forward or backward direction and positioned with the LEFT and RIGHT scroll keys.

Formatted and Horizontal features include:

- Data is displayed and edited using definitions from the DB2 catalog without needing to prepare copybooks.
- Data from character columns defined as containing text are displayed in character, and data from non-text character columns is displayed in hexadecimal format. The display format can be different for each column.
- A data field that does not fit on one row is wrapped to the next row (Formatted).
- Column and data values are presented in the sequence defined by the user and limited subsets of the columns and rows may be specified by the user.

DB2 Specifications

MAX DB2/UTIL provides support for the following DB2 facilities:

- All DB2 V6, V7 and V8 data types.
- BLOB, CLOB and DBCLOB values up to 4 megabytes.
- Tables with and without primary keys.
- Referential integrity.
- CHECK constraints.
- Triggers.
- Standard existing security packages.
- Table names up to 128 characters in length and column names up to 30 characters.

Launching MAX DB2/UTIL

There are several ways to execute MAX DB2/UTIL after logging on to ISPF:

Option `D`

If your ISPF Primary Option Menu has been modified, enter '`D`' at the option prompt to display the MAX DB2/UTIL Main Menu.

```
TSO %MAX DB2
```

Enter the MAX DB2 command from any command line to display the MAX DB2/UTIL Main Menu.

```
TSO %MAX DSNL
```

Enter the MAX DSNL command from any command line to display the currently established DSNL.

"*Chapter 3: SQL Command Utility*" on page 71 describes the construction and use of Data Set Name Lists (DSNL) to invoke MAX DB2/UTIL with stored parameter on specific DB2 tables.

CHAPTER 2: HOW TO USE MAX DB2/UTIL

MAX DB2/UTIL operates like ISPF, using the same PF keys, online tutorials, split Panel capabilities, and Panel navigation procedures.

The conventions used in MAX DB2/UTIL are similar to ISPF conventions.

In addition, MAX DB2/UTIL is supported by extensive online tutorial panels that may be accessed at any time by pressing the HELP PF (program function) key.

The main point of entry to MAX DB2/UTIL is the Specify a Table Name panel. The following is a sample of the Specify a Table Name panel.

```
MAX DB2/UTIL --------------- SPECIFY A TABLE NAME --------------- MAX DB2/UTIL
COMMAND ===> ____________________________________________________________

Select one of the following. Then press Enter.
___  0 Profile parameters          ? List all MAX DB2/UTIL options
     1 Browse table  (default)       Load table, Unload data,
     2 Edit table                    Relational unload
     3 DB2 utilities (RUNSTATS)

Specify table or view:
OWNER ID . . . . . . %_______          (Wild cards (% or _) may be used)
TABLE/VIEW NAME. . . EMP%_____________________________________________________
                     ____________________________________________________________
                     ________________  (Wild cards (% or _) may be used)
DB2 SUBSYSTEM. . . . DSN1              (Not currently connected to DB2)

INITIAL DISPLAY. . . FORMATTED_        (Formatted, Horizontal)
CRITERIA DSN . . . . ____________________________________________________________
CURRENT SQLID. . . . ________           (Default is MX10006)


        (c) Copyright MAX SOFTWARE Inc. 2002-2003. All rights reserved.
```

Figure 1: Specify a Table Name panel

From this panel, the DB2 subsystem ID and Table/View name are entered to identify the DB2 table to be processed.

In preparation for using MAX DB2/UTIL, the following presents some general usage information.

Primary Commands

Enter primary commands at the COMMAND ===> prompt located in the upper left corner of a panel. Primary command abbreviations will be indicated within parentheses in this manual.

Utilize the following guidelines when entering primary commands:

- Enter a blank to separate operands, **do not use the CURSOR keys**.
- Insert or expand operands using the system insert mode.
- Enter multiple commands in the COMMAND field by entering a semicolon between each command. This process is known as "command stacking."

Line Commands

Enter executable line commands in the COMMAND area to the left of the line under the ROW column in Horizontal mode, and the POS column in Formatted mode. Single-character line commands operate on individual lines.

DB2 Table Name Specification

The Browse and Edit Table functions all require specification of a DB2 Table/View name to process.

A DB2 table or view name consists of two parts: the owner ID and the name. You can specify the full owner ID and name of the table or view, or you can let MAX DB2/UTIL search for names matching a pattern. A percent sign (%) matches zero or more characters and an underscore (_) matches any single character. If only a percent sign is entered into a field, all names will match. The name of the DB2 subsystem containing the table or view must also be entered.

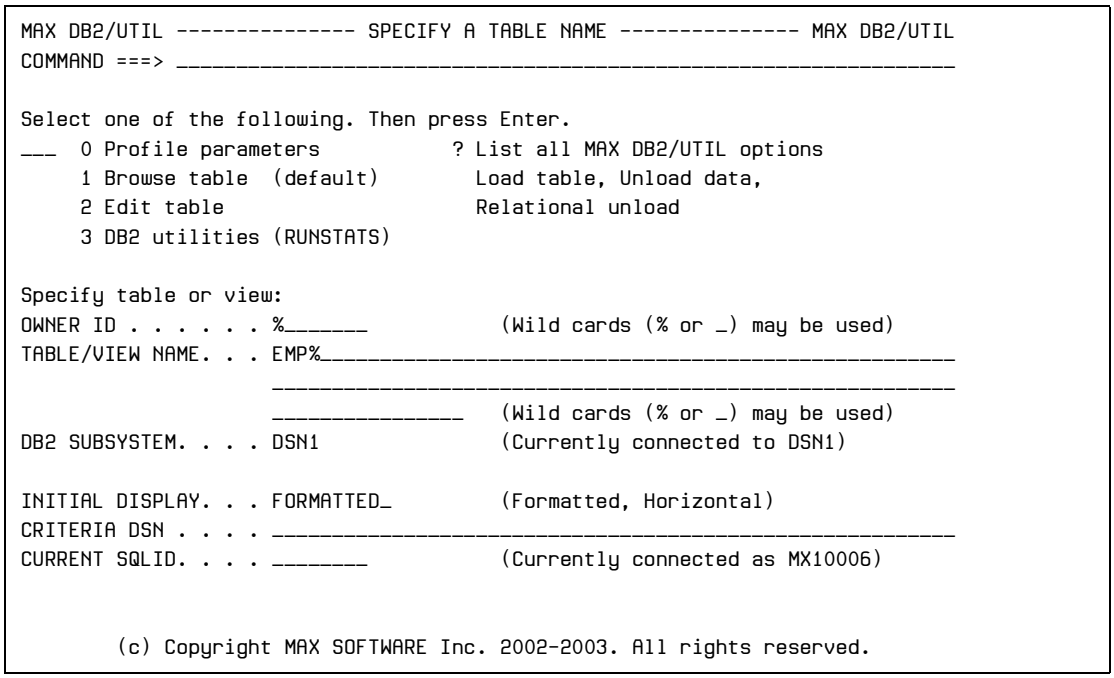

Figure 2: Specify DB2 Table Name panel

INITIAL DISPLAY: The **BROWSE** and **EDIT** functions can display table or view rows in either Formatted (single row) format, or Horizontal (multiple row) format. This field sets the initial display format. The display format can be changed once in the edit or browse functions.

CRITERIA DSN: If a data set name is entered in this field, MAX DB2/UTIL will automatically load previously saved SELECT/WHERE/ORDER criteria from the data set. If the data set is a partitioned data set (PDS), a member name may also be entered. If a member name is not entered for a PDS, then a list of members will be displayed. If a data set name is not entered, you can still load saved SELECT/WHERE/ORDER criteria before editing or browsing the table or view.

CURRENT SQLID: By default, MAX DB2/UTIL will connect to the DB2 subsystem with your TSO user ID as the CURRENT SQLID. This field allows you to override the CURRENT SQLID with a different name.

Once MAX DB2/UTIL is connected to a DB2 subsystem, the subsystem name and CURRENT SQLID will be displayed to the right of these entry fields. You can change these values to have MAX DB2/UTIL re-connect to a different subsystem or CURRENT SQLID.

Table Selection List

If wild card characters are used for the owner ID and/or table/view name, MAX DB2/UTIL will search the named DB2 subsystem for all tables or views matching the pattern. If the exact name is entered or only one table or view matches the pattern, MAX DB2/UTIL will start the requested function on the table or view. If more than one name matches the pattern, a list of all matching names will be displayed allowing you to select the table or view.

Note: The percent sign (`%`) and underscore (`_`) are the normal DB2 wild card characters. MAX DB2/UTIL will automatically translate the asterisk (`*`) to a percent sign and a question mark (`?`) to an underscore to aid anyone preferring to use those characters.

```
 MAX DB2/UTIL --------------- TABLE SELECTION LIST ------------ Row 1 to 4 of 4
COMMAND ===>                                                  SCROLL ===> PAGE
B - Browse   E - Edit   L - Load   U - Unload   / - List actions   S = B
  Owner    Name               Type  Label
  MX10006  EMP_CURRENT        View  MAX DB2/UTIL sample view
  MX10006  EMP_FORMER         Table MAX DB2/UTIL sample table: deleted rows
  MX10006  EMP_PHOTO_RESUME   Table MAX DB2/UTIL sample table with LOB columns
  MX10006  EMP_PRIVATE        View  MAX DB2/UTIL sample data privacy view
******************************** Bottom of data ********************************
```

Figure 3: Table Selection List panel

The list of table names matching the entry criteria is displayed for selection. Entry of an ‘`S`’ will select that table for processing with the main menu entered function, while ‘`/`’ will cause a list of available actions to be displayed, allowing you to select one.

Column and Row Selection

Once a table or view has been selected for edit or browse, a list of the columns will be displayed. From this list, you can select which columns and rows will be displayed and in what order. The current column selection list will be redisplayed once you leave the editor or browser giving you the chance to re-enter the editor or browser with different criteria. You can also **SAVE** and later **COPY** selection criteria.

```
MAX DB2/UTIL --------------- COLUMN SELECTION LIST --------------- Row 1 of 11
COMMAND ===> _____________________________________________ SCROLL ===> PAGE
Use GO to edit table MX10006.EMP_PHOTO_RESUME
Other commands are: CHECK,SQL,UNLOAD,RESET,ALL,NONE,SAVE,COPY,END and CANCEL.

                      Default WHERE connector. . . . . AND  (And or Or)
Col Row    Data
Seq Ord A/D Fmt Column Name                    Type
1__ 1__  A   _  EMPNO                          INTEGER NOT NULL
                Where: ____________________________________________________
2__ ___  A   _  FIRSTNAME                      VARCHAR(12) NOT NULL
                Where: ____________________________________________________
3__ ___  A   _  MIDINIT                        CHAR(1)
                Where: ____________________________________________________
4__ ___  A   _  LASTNAME                       VARCHAR(15) NOT NULL
                Where: ____________________________________________________
5__ ___  A   _  GENDER                         CHAR(1) NOT NULL
                Where: ____________________________________________________
6__ ___  A   _  JOB                            CHAR(8) NOT NULL
                Where: ____________________________________________________
7__ ___  A   _  HIREDATE                       DATE NOT NULL
                Where: ____________________________________________________
8__ ___  A   _  SALARY                         DECIMAL(9,2) NOT NULL
                Where: ____________________________________________________
9__ ___  _   _  RESUME                         CLOB(5120)
                Where: ____________________________________________________
10_ ___  _   X  BMP_PHOTO                      BLOB(102400)
                Where: ____________________________________________________
11_ ___  A   _  EMP_ROWID                      ROWID NOT NULL
                Where: ____________________________________________________
******************************** Bottom of data ********************************
```

Figure 4: Column Selection List panel

Column Selection

The columns of the selected table or view are listed in the order they were defined. Initially, all columns are selected in that same order. A number in the `Col Seq` column indicates that the corresponding table column is selected and the sequence in which the column will appear in the editor or browser. If the `Col Seq` column is blank, the corresponding table column will not be displayed by the editor or browser. You can use the **RESET** or **ALL** command to select every table column or the **NONE** command to deselect every column.

A column must be selected to be used as a sort column (see Row Ordering). A column does not need to be selected to be used in the row selection criteria (see Row Selection). If there are duplicate numbers in the `Col Seq` column, those table columns will be selected in the order in which they were defined in the table or view.

Row Selection

A subset of the table or view rows can be selected by entering row selection criteria. It is particularly important to use the row selection criteria when editing or browsing large tables. It is much faster to select only the rows you want in the Column Selection List than it is to select all rows and then use the **FIND** command to locate the rows you are interested in.

Anything entered for `Where:` is used to build a SQL WHERE clause to select rows. You can enter a comparison operator below a column name to select rows based on the values of that column. If you are familiar with SQL, you can use the `Where:` area to enter a free-form WHERE clause.

The `Default WHERE connector` value is automatically added to all but the last `Where:` value unless the entered value ends with **AND** or **OR**.

If a `Where:` value starts with any of the following comparison operators or starts with the word **NOT** followed by one of these operators, the corresponding column name will be automatically prefixed to the `Where:` value: `=`, `<`, `>`, `!`, , `IN`, `IS`, `LIKE` and `BETWEEN`. Thus, you can specify a `Where:` value for a numeric column like this: '`>5`' which will select only rows where that column has a value greater than '`5`'.

If the `Where:` value does not start with one of the above comparison operators, the column name will not be prefixed to the `Where:` value, allow more free-form entry of the WHERE clause.

Row Ordering

The Row Ord and A/D columns are used to indicate the order in which the editor or browser will present selected rows. To sort the selected rows on a table column, select that column by entering a number under Row Ord. This will select that column for sorting. The column must also be selected by entering a number under Col Seq. If no columns are selected for sorting, it is unpredictable in what order the selected rows will be displayed.

The rows will be sorted by the column with the lowest number under Row Ord. If there are multiple rows with the same value in that column, those rows will be sorted by the column with the next highest number under Row Ord and so on. The A/D column indicates if the ordering is to be Ascending or Descending for that table column.

If a primary key is defined for the table, the initial Column Selection List will select the rows to be ordered in the same sequence as the primary key. DB2 columns defined as LONG VARCHAR, LONG VARGRAPHIC, BLOB, CLOB or DBCLOB cannot be used as sort columns.

Data Formatting

The Data Fmt column is used to indicate the display and edit formatting for character columns. The data format may be set to 'X', 'C', 'U' or may be blank. If set to 'X', the column data will be presented in hexadecimal. If set to 'C' the column data will be presented in character format and mixed-case input will be accepted. If set to 'U', the column data will also be shown in character format but all input will be translated to uppercase. If the Data Fmt column is left blank, the column data will be presented in character format unless the display mode is Formatted and the column value contains non-displayable characters in which case the data will be presented in hexadecimal.

The Data Fmt column is ignored for numeric, date/time and ROWID columns. By default, BLOB and FOR BIT DATA columns are set to 'X' format and all other column formats are set to blank. The format of character columns can be changed in the editor or browser using the **CHAR**, **HEX**, **UPPER** or **DATA** primary, line or column commands. The Data Fmt column has no effect on an unload request.

Column Selection Primary Commands

A number of primary commands are available in the Column Selection List. If you just want to leave the Column Selection List panel, you can use the **END** or **CANCEL** commands. Both will return to the previous screen, either the main menu or the Table Selection List panel. The difference is that the **CANCEL** command will skip any automatic saving of the criteria, whereas **END** will perform validation and automatic saving of the criteria if requested on the last **COPY** or **SAVE**.

RESET

Resets the selection criteria to the default for the table or view. All columns are selected in the order they were defined. The row ordering is reset and the row selection is cleared. Any pending autosave is cleared.

ALL

Is similar to **RESET**, but does not clear the row selection or autosave data set. All columns are selected.

NONE

Is the opposite of **ALL**. All columns are deselected. If you only want to select a few columns from a table with many columns, it is quicker to use the **NONE** command and then select the columns you want.

GO

Validates the column and row selection criteria. If no problems are found, the editor or browser is started.

SQL

Validates the column and row criteria. If no problems are found, the corresponding SQL statement is displayed. If a problem is discovered, the DB2 error message is shown. The **CHECK** command also works on the Column Selection List to perform the SQL function. Once the SQL statement is displayed, you can format it and the corresponding data definitions for COBOL, PL/I, C, C++, assembler or MAX/REXX and save the code in a file or data set. The saved code can be used to help write SQL programs.

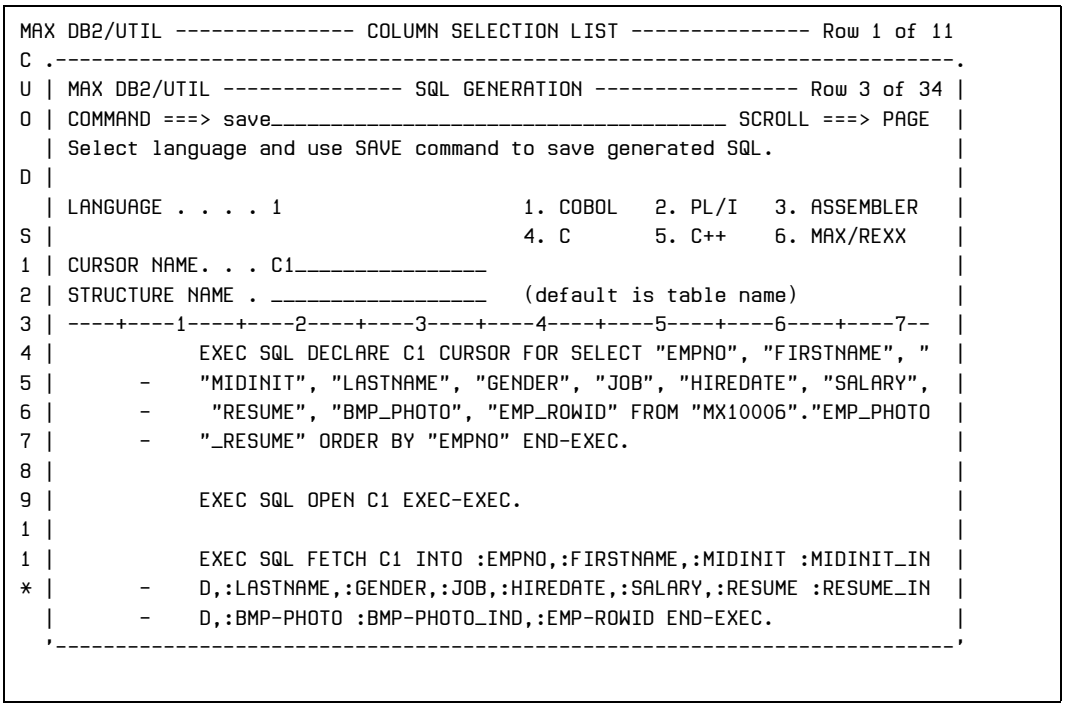

Figure 5: Copy Select/Where/Order Criteria

COPY

COPY reloads previously saved selection criteria from a sequential data set or PDS member. The saved criteria can be from a different table or view, but the selected column names and definitions must be compatible. Unselected columns do not matter.

A pop-up panel will be displayed allowing you to specify the data set and PDS member, and if any changes should be saved automatically upon leaving the Column Selection List. If you omit the PDS member name for the **COPY** command, you will be presented with a list of the members in the PDS.

```
MAX DB2/UTIL --------------- COLUMN SELECTION LIST --------------- Row 1 of 11
C .-------------------------------------------------------------------. GE
U | MAX - COPY SELECT/WHERE/ORDER CRITERIA                            |
O | COMMAND ===>                                                      |
  |                                                                   |
D | Specify "FROM" data set below.                                    |
  |                                                                   |
S | From partitioned or sequential data set:                          |
1 |    DATA SET NAME. . .                                             |
2 |    VOLUME SERIAL. . .            (If not cataloged)               |
3 |                                                                   |
4 | "FROM" data set options:                                          |
5 |    AUTOMATICALLY SAVE CHANGES . . . YES   (Yes, No)               |
6 |                                                                   |
7 |    Press ENTER to perform the copy request.                       |
8 |    Enter END command to cancel the copy request.                  |
9 '-------------------------------------------------------------------'
10          BMP_PHOTO          BL(100K)
11       A  EMP_ROWID          ROWID
******************************** Bottom of data ********************************
```

Figure 6: Copy Select/Where/Order Criteria

SAVE

SAVE saves the current selection criteria in a sequential data set or PDS member. Before saving the criteria, it is validated and any errors are reported. (See the **SQL** command description on page 12 for further information.) The output data set must already exist before the **SAVE** command is used.

A pop-up panel will be displayed allowing you to specify the data set and PDS member names.

```
MAX DB2/UTIL --------------- COLUMN SELECTION LIST --------------- Row 1 of 11
C .-------------------------------------------------------------------. GE
U | MAX - SAVE SELECT/WHERE/ORDER CRITERIA                            |
O | COMMAND ===>                                                      |
  |                                                                   |
D | Specify "SAVE TO" data set below.                                 |
  |                                                                   |
S | To partitioned or sequential data set:                            |
1 |    DATA SET NAME. . .                                             |
2 |    DESCRIPTION. . . .                                             |
3 |    VOLUME SERIAL. . .            (If not cataloged)               |
4 |                                                                   |
5 | "SAVE TO" data set options:                                       |
6 |    IF PARTITIONED, REPLACE LIKE-NAMED MEMBER. . .       (Yes, No) |
7 |    AUTOMATICALLY SAVE CHANGES . . . . . . . . . . NO    (Yes, No) |
8 |                                                                   |
9 |    Press ENTER to perform the save request.                       |
1 |    Enter END command to cancel the save request.                  |
1 '-------------------------------------------------------------------'
****************************** Bottom of data ********************************
```

Figure 7: Save Select/Where/Order Criteria

Menu Options

The following sections are detailed descriptions for each option displayed in the Specify a Table Name panel:

0. Profile Parameters
1. Browse Table
2. Edit Table
3. DB2 Utilities
4. Unload Data
5. Load Table
6. Relational Unload

0. Profile Parameters

Profile parameters may be entered into the DB2/UTIL Profile Options panel to configure various features of a MAX DB2/UTIL session. Parameters entered into this panel will remain in effect from session to session. Use the **RESET** command to reset all fields to their respective defaults.

```
MAX ---------------------- DB2/UTIL PROFILE OPTIONS ---------------------- MAX
COMMAND ===> _____________________________________________________________
Press ENTER to update profile, END to exit or use RESET command for defaults.
Press DOWN  to scroll forward, UP  to scroll backward.

Commit frequency (number of rows changed) . . 1__        (0 = Upon save)

Maximum character lines (FORMATTED) . . . . . 100        (1-999)

Maximum character column width (HORIZONTAL) . 100__      (1-99999)

Row fetch limit (prevent runaway queries) . . 1000___    (0 = No limit)

Row paste limit (prevent long PASTE). . . . . 100____    (0 = No limit)

DB2 page access warning level . . . . . 1000____   (0 = No warning)

DB2 plan name override. . . . . ________   (blank=default)

Terminal code page (ISPF override). . . . . DEFAULT    (DEFAULT,CP870,CP838)

CANCEL Log Data Set allocation unit name. . . SYSALLDA   (Example: SYSALLDA)
  Primary number megabytes. . . . . 10___      (1-99999)
  Secondary number megabytes. . . . . 50___      (0-99999)

Application relationship definition file for Relational Unload:
  Data Set Name . . . . . 'MAX.SQL.RELATE'_________________________________
```

Figure 8: DB2/UTIL Profile Options panel

Field Definitions for selected functions:

Commit frequency (number of rows changed): This determines how often MAX DB2/UTIL will commit your changes to DB2. Committing changes releases locks on updated rows, allowing others to access the table. However, once some changes are committed, MAX DB2/UTIL may not be able to back out changes if the **CANCEL** command is used. Setting the frequency to zero prevents MAX DB2/UTIL from committing your changes until you end the edit session, or you issue the **COMMIT** or **SAVE** commands and allows full reversal of changes upon a **CANCEL**.

Maximum character lines (FORMATTED): In Formatted Edit and Browse mode, this determines how many lines to display for each character value. If more than this number of lines is required for the value, the value will be marked as truncated and it will not be changeable without using the value editor.

Maximum character column width (HORIZONTAL): In Horizontal Edit and Browse mode, this determines how many characters of a character column to display. If more than this number of characters is required for the column, the column width will be truncated. The column heading will end with '>' and any truncated values will also end with '>'.

Row fetch limit (prevent runaway queries): Setting this to a non-zero value will cause MAX DB2/UTIL to warn you if the number of result rows fetched for a query exceeds the value. You will be given a chance to end the query or to continue with a new warning value.

Row paste limit (prevent long PASTE): Setting this to a non-zero value will cause MAX DB2/UTIL to warn you if the number of pasted rows exceeds the set value. You will be given a chance to end the **PASTE** operation or to continue with a new warning value.

DB2 page access warning level: Setting this to a non-zero value will cause MAX DB2/UTIL to warn you if it determines that a query will potentially access more than this number of pages (I/O operations) before the first row can be fetched. A non-zero value will also cause MAX DB2/UTIL to warn you if it cannot estimate the cost of a query because RUNSTATS has not been run.

DB2 plan name override: Setting this to a non-blank value will cause MAX DB2/UTIL to connect to DB2 with the specified plan name rather than the default plan name. Your DBA may have bound alternate plans to provide for different cursor stability options or authorization.

Terminal code page (ISPF override): Setting this to other than DEFAULT overrides the ISPF terminal setting for the set of displayable characters and the uppercase and lowercase letters.

CANCEL Log Data Set allocation unit name: In order for the **CANCEL** command to back out committed updates, all updates must be written to a temporary file called the "**CANCEL** Log Data Set". This data set must be large enough to hold all of the updates performed between **SAVE** commands. You can override the default allocation unit name and size of this data set. If you will be updating large object columns (BLOB, CLOB, or DBCLOB), you may need to increase the allocation size.

If you are editing a table and have not performed a **SAVE** command, the **CANCEL** Log Data Set is already allocated and changing the allocation definitions will not take effect until the next time the data set is allocated.

The **CANCEL** Log Data Set name always has a high level qualifier of you TSO user ID, a middle name of MAXDB2UT, and a last name of TEMPLOGn where the letter 'n' is replaced with the ISPF session number.

Application relationship definition file for Relational Unload:

> Data Set Name: Setting this to a non-blank value will allow MAX DB2/UTIL users to define application relationships between tables. These relationships are used by the Relational Unload function. These relationships may be defined to MAX DB2/UTIL and stored in the VSAM file named on the profile screen. If the field is left blank, only FOREIGN KEY relationships are used.

1. Browse Table

The MAX DB2/UTIL Browse Facility allows you to view DB2 table rows in two different display modes:

1. Browse, Formatted Display
2. Browse, Horizontal Display

The Browse Facility is controlled by primary commands that are used to:

- Find and display rows containing specified character strings.
- Display SQL SELECT statement used to obtain result set.
- Locate a specific table row by row number.
- Locate a column name within a table row.
- Scroll through table rows.
- Display column numbers.
- Display column data in hexadecimal format.
- Toggle between browse and edit.
- Toggle between display options.
- Print table row contents.
- **CUT** (copy) table rows to a temporary area for a subsequent **PASTE** (insert) into the same or other Edit session.

Browse, Formatted Display

The following is a sample panel showing the Browse, Formatted Display:

```
MAX FORMATTED BROWSE TABLE MA10002.EMP_PHOTO_RESUME             Line 1 of 135
COMMAND ===>                                                  SCROLL ===> PAGE
Display:  DH - Horizontal  ED - Edit                              Row 1 of 4
Read:     N  - Next        P  - Previous     L  - Locate by row
POS   COLUMN NAME *-TYPE--* N *-------------------CONTENTS--------------------*
      EMPNO       INTEGER     130
_____ FIRSTNAME   VC(7)       DOLORES
_____ MIDINIT     CHAR(1)   N M
_____ LASTNAME    VC(8)       QUINTANA
_____ GENDER      CHAR(1)     M
_____ JOB         CHAR(8)     ANALYST
      HIREDATE    DATE        28.07.1971
      SALARY      DEC(9,2)    +23800.00
00001 RESUME      CL(1201)  N     Resume:  Delores M. Quintana    Personal Info
00050                           rmation   Address:            1150 Eglinton Ave
00099                                              Mellonville, Idaho 83725  Pho
00148                           ne:              (208) 875-9933  Birthdate:
00197                               September 15, 1925  Sex:                Femal
00246                           e  Marital Status:     Married  Height:
00295                              5'2"  Weight:             120 lbs.    Departme
00344                           nt Information   Employee Number:    000130  Dept
00393                            Number:        C01  Manager:             Sally Kw
00442                           an  Position:           Analyst  Phone:
00491                               (208) 385-4578  Hire Date:          1971-07-2
00540                           8    Education   1965                Math and Eng
00589                           lish, B.A.                      Adelphi Universit
00638                           y   1960                Dental Technician
00687                                          Florida Institute of Technology
00736                           Work History   10/91 - present     Advisory Syste
00785                           ms Analyst                      Producing documen
00834                           tation tools for engineering
00883                            department.   12/85 - 9/91        Technical Writ
```

Figure 9: Formatted Browse panel

The Formatted Browse panel shows the values of a single row. If the single values of a row do not fit on the screen, you can scroll UP and DOWN to see the rest. In Browse mode, you can also scroll LEFT and RIGHT to view the previous and next rows.

The upper right corner of the screen will show the visible screen rows. The COMMAND area allows entry of primary commands, and the SCROLL area allows the default scroll amount to be changed.

The Display: line shows the available screen format commands. The Read: line shows the row positioning commands.

The POS field allows entry of line commands for character type values. If the value requires more than one line to display, the POS field of the second and subsequent lines will contain the position of that portion of the value.

The COLUMN NAME field shows the name of the column, and the *-TYPE--* field shows the actual data type. The data type names may be abbreviated to fit. You may see DEC or DC for DECIMAL, GRPH for GRAPHIC, VC for VARCHAR, VG for VARGRAPHIC, BL for BLOB, CL for CLOB, or DB for DBCLOB. The length may also be shortened to K (kilobytes) or M (megabytes).

The N field is present if any selected columns were defined without the **NOT NULL** clause, so they may contain a null value. If the column value is **NULL**, the letter Y will be displayed in the N field and the CONTENTS field will be blank. If the column value is **NOT NULL**, the letter N will be displayed in the N field and the value will be displayed in the CONTENTS field.

The CONTENTS field shows the value of the column for the current row. A numeric value is shown for INTEGER, SMALLINT, DECIMAL, FLOAT, or DOUBLE columns. Date/time columns (DATE, TIME and TIMESTAMP) are shown as character values. ROWID column values are shown in hexadecimal format. The remaining data types (CHAR, GRAPHIC, VARCHAR, VARGRAPHIC, BLOB, CLOB, and DBCLOB) are shown in either character or hexadecimal formats.

If a character field contains undisplayable characters and the column is being displayed in character mode, the undisplayable characters are shown as a dot. If a character field is displayed in hexadecimal format, an indicator line (-------- Hexadecimal --------) will be displayed above the value. You can enter the **HEX** or **CHAR** commands in the POS field to switch the column display between hexadecimal and character mode.

If a character value requires more lines than the limit set in the DB2/UTIL Profile Options panel, an indicator line (-------- Truncated --------) will be displayed above the first value line.

You can display the entire value of any character column by invoking the value browser by entering the 'B' line command in the POS field for the value.

Browse, Horizontal Display

The Horizontal Display allows you to view the table result set while presenting data from multiple rows on one panel.

```
MAX HORIZONTAL BROWSE TABLE MA10002.EMP_PHOTO_RESUME            1 to 72 of 486
COMMAND ===>                                                  SCROLL ===> PAGE
Display:  DF - Formatted   ED - Edit                         L - Locate by row
Type:  INTEGER      VARCHAR(12)    CHAR(1)  VARCHAR(15)      CHAR(1)  CHAR(8)
ROW--- EMPNO------  FIRSTNAME---   MIDINIT  LASTNAME-------  GENDER-  JOB-----
000001         130  DOLORES         M        QUINTANA         M        ANALYST
000002         140  HEATHER         A        NICHOLLS         F        ANALYST
000003         150  BRUCE          <null>    ADAMSON          M        DESIGNER
000004         190  JAMES           H        WALKER           M        DESIGNER
****************************** Bottom of data ********************************
```

Figure 10: Horizontal Browse panel

The Horizontal Browse panel displays as many available rows as there is room. The screen may be scrolled UP and DOWN to display other rows in the result set. If the number and width of the selected columns is wider than the screen, you can also scroll LEFT and RIGHT to view the data.

The upper right corner of the screen will show the visible screen columns. The COMMAND area allows entry of primary commands and the SCROLL area allows the default scroll amount to be changed. The third line shows the available screen format commands.

The top row of the column header normally shows the defined data type. If the **COLS** command has been entered for the column, the column scale will be displayed instead of the data type. For character type columns, the headings in this row may be overtyped with column commands.

The second row of the column header shows the column names.

The remaining lines of the screen show as many result rows as will fit on the screen. At the left side of each line is the relative row number of the result set row. These numbers are assigned as each row is retrieved or added and will not change if rows are deleted. The **LOCATE** command may be used to position the screen by result row number.

Character (CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, BLOB, CLOB, and DBCLOB) values are displayed in either character or hexadecimal format. You can switch a column between character and hexadecimal by entering either **CHAR** or **HEX** in the column heading or you can switch all columns by using the **CHAR** or **HEX** primary commands.

Numeric (INTEGER, SMALLINT, DECIMAL, FLOAT, and DOUBLE) values are displayed as numbers, date/time (DATE, TIME and TIMESTAMP) values are displayed as character strings, and ROWID values are displayed in hexadecimal. Integer and decimal values are right justified. All other values are left justified.

If a character column is longer than the limit set in the DB2/UTIL Profile Options panel, the values may be truncated. Truncation is indicated by the '>' character at the right end of the column heading or value.

In Horizontal Browse mode, the column headings of character columns containing the data type can be overtyped with a command to switch the display format for that column between character and hexadecimal. The following column commands are supported:

CHAR can be used to switch the column format from hexadecimal to character

HEX can be used to switch the column format from character to hexadecimal

These commands are equivalent to the line commands by the same name in Formatted Browse mode. Once the **CHAR** or **HEX** command is entered in either display mode for a column, the format of the column will be used for the remainder of the Browse session until changed.

Browse, Primary Commands

Command	Description	Formatted	Horizontal
CHAR	Display character.	YES	YES
COLS	Specify columns.	NO	YES
COUNT	Count rest of result set.	YES	YES
CUT	Copy row(s).	YES	YES
DATA	Data-dependent display.	YES	YES
DF	Display formatted mode.	YES	YES
DH	Display horizontal mode.	YES	YES
EDIT	Edit table.	YES	YES
END	End session.	YES	YES
FIND	Find data.	YES	YES
HEX	Display hexadecimal.	YES	YES
LOCATE	Locate row by row-number.	YES	YES
LOCATEF	Locate column name.	YES	YES
NEXT	Scroll forward.	YES	NO
PREV	Scroll backward.	YES	NO
PROFILE	Display profile parameters.	YES	YES
PRT	Print row(s).	YES	YES
REFRESH	Retrieve result set.	YES	YES
RESET	Reset display.	YES	YES
SQL	Display SQL SELECT statement.	YES	YES

CHAR

CHAR [*|column]

The **CHAR** command sets the display format for one or more character columns to character format. This command is available in all display modes. If an asterisk is specified for the column name, this is equivalent to entering the **CHAR** command into every line in Formatted mode or every column in Horizontal mode. If a column name is specified, only that column's display format will be changed. If no column name is specified, a list of the character column names will be displayed, allowing you to set the display format for each.

COLS

COLS [ON|OFF]

The **COLS** command is available in Horizontal mode to switch the column heading between the data type and a column scale. If neither **ON** nor **OFF** is specified, the heading is switched to the other format.

COUNT

COUNT

This command causes the rest of the result set to be retrieved and subsequently displays the number of rows returned by the query.

CUT

CUT [rows] [R]

The **CUT** command copies one or more rows to the "**CUT** buffer, data set or UNIX file." These rows can be inserted later into the same or another table using the **PASTE** command. The **CUT** command is available in any display mode except New Row Entry. The **PASTE** command is only available in Edit mode.

The '**rows**' and '**R**' parameters are optional. If '**rows**' is not specified, only one row will be cut. In Formatted mode, rows are copied starting with the currently visible row. In Horizontal mode, rows are copied starting with the row at the top of the screen. If '**R**' is specified, the copied rows will replace any previously copied rows in the **CUT** buffer, data set or UNIX file. If '**R**' is not specified, the copied rows will be added to any previously copied rows as long as the previous rows are from a table or view with selected columns of the same name and data type.

DATA

DATA [*|column]

The **DATA** command sets the display format for one or more character columns to data-dependent format. This command is available in all display modes. If an asterisk is specified for the column name, this is equivalent to entering the **DATA** command into every line in Formatted mode or every column in Horizontal mode. If a column name is specified, only that column's display format will be changed. If no column name is specified, a list of the character column names will be displayed, allowing you to set the display format for each column.

DF

DF

In Horizontal mode, this command changes the display to Formatted. The Edit/Browse mode is not changed by this command.

DH

DH

In Formatted mode, this command changes the display to Horizontal. The Edit/Browse mode is not changed by this command.

EDIT

EDIT
ED

This command changes the display to **EDIT**. The Formatted/Horizontal mode is not changed by this command. You cannot change the display of a view from Browse to Edit.

END

END

The **END** command ends the Browse session.

FIND

FIND [value [column [start [end]]]] [**NEXT**|**PREV**|**FIRST**|**LAST**]
>RFIND [**NEXT**|**PREV**|**FIRST**|**LAST**]

The **FIND** command locates an occurrence of a value within the result set. The **FIND** command can be used in both Browse and Edit modes and in both Horizontal and Formatted modes, although the display will switch to Formatted mode when a value is found. The **FIND** command can be abbreviated to the letter **F**.

The **>RFIND** command repeats a previous **FIND** action. **>RFIND** is usually invoked by the PF5 key. (The leading '**>**' is required by ISPF since **RFIND** is normally disabled outside of the PDF editor.)

If the find value is not specified, the **FIND** and **>RFIND** commands both will repeat the **FIND** action of a previous **FIND** or **CHANGE**. Both of these versions of the **FIND** command can be qualified with any of the key words: **NEXT**, **PREV**, **FIRST** or **LAST** to indicate the direction of the search.

NEXT The **NEXT** key word starts the search from the cursor location, finding the next occurrence of the value. For a **FIND** command with a value, the default search direction is **NEXT**.

PREV The **PREV** key word also starts the search from the cursor location but locates the previous occurrence of the value from the cursor.

FIRST The **FIRST** key word starts the search from the start of the value, finding the first occurrence.

LAST The **LAST** key word starts the search from the end of the value finding the last occurrence. If a **FIND** with **FIRST** or **LAST** is repeated without a direction, the default direction is **NEXT** or **PREV** respectively.

The value to be found may be preceded by the word **NOT** which indicates that the value is considered to be found when it does not match.

The value may be specified in several ways:

NULL This searches for a null value, whereas **NOT NULL** searches for values that are not null. Only columns not defined as **NOT NULL** are searched.

operator number This performs comparisons on values in numeric columns.

string This performs character comparisons. Only non-numeric columns are searched and null values are not compared. **NOT** string will match any value that does not contain the string.

" " or ' ' This performs searches varying length columns for values of a matching length.

The syntax for searching for a numeric value is:

FIND **[NOT] operator value [column] [NEXT|PREV|FIRST|LAST]**

Numeric columns (INTEGER, SMALLINT, DECIMAL, FLOAT and DOUBLE) are searched using a comparison operator and numeric value. The operator must be one of: =, <, >, <>, >= or <=. The **NOT** key word reverses the comparison, thus **NOT** =0.0 and <> 0 both search for a number that is not zero. Blanks are optional between the operator and the number. The number may include a sign, decimal point and digits and an exponent. Null values are not compared.

Note: Without an operator, a character search of non-numeric columns will be performed. For example, **F** 0 will not search numeric columns.

Character values may be specified in several ways:

Tttt **T'tttt'** **T"tttt"**	These perform a case-insensitive search for the characters.
'cccc' **"cccc"** **C'cccc'** **C"cccc"**	These perform a case-sensitive search for the characters.
X'xxxx' **X"xxxx"**	These search for a hexadecimal value. An even number of hexadecimal digits must be specified.

The characters must be enclosed in quotes if any of the characters are quotes (" or ') or blanks, if the string starts with a comparison operator (=, >, <), or if they are a key word (**NOT**, **NULL**, **NEXT**, **PREV**, **FIRST**, **LAST**). Quoted strings cannot contain the enclosing quote character. The **NOT** key word causes a match for values that do not contain the characters.

The empty string (" " or ' ') has a special meaning. It is used to search for strings with a specific length, as described below.

To search varying length columns (VARCHAR, VARGRAPHIC, ROWID, BLOB, CLOB, and DBCLOB) for values with a specific length or range of lengths, specify the search value as the empty string ("" or ' '). Without the column name, all selected varying length columns will be searched for zero-length values.

Note: A zero-length string is not the same as a null value.

To search a varying length column for values with a specific length, specify the column name and start position. This will search for values in that column with the specified length. To search a varying length column for values with a range of lengths, specify the column name and both the start and end positions.

Using the empty string as a search value will skip numeric columns, fixed length columns (DATE, TIME, TIMESTAMP, CHAR and GRAPHIC) and null values. The **NOT** key word will match values that do not have the specified length.

You can limit the search for a value to a specific database column by following the search value with a column name. The named column must be one of the selected columns. The data type of the column must match the search value. The column must not be defined as **NOT NULL** to search for **NULL** or **NOT NULL**, the column must be numeric for a numeric comparison, the column must be non-numeric for a character search and the column must be varying for a length comparison.

For character searches, you can further limit the search to a specific position within the value or a subset of the value. If only start is specified, the comparison is performed on the portion of values from the starting position for a length of the search value. If both start and end are specified, the range of characters between the starting and ending positions of the column values are searched. No padding is performed, so values shorter than the search range will not match. If the characters are found in more than one place in the column value, the cursor will be placed at the first (or last) found position within the value editor or browser. You can then repeat the **FIND** as needed.

HEX

HEX **[*|column|ON|OFF]**

The **HEX** command sets the display format for one or more character columns to hexadecimal format. This command is available in all display modes. If an asterisk is specified for the column name, this is equivalent to entering the **HEX** command into every line in Formatted mode or every column in Horizontal mode. If a column name is specified, only that column's display format will be changed. If no column name is specified, a list of the character column names will be displayed, allowing you to set the display format for each. '**HEX ON**' is equivalent to '**HEX ***' and '**HEX OFF**' is equivalent to '**CHAR ***'.

LOCATE

LOCATE **row-number**

The **LOCATE** command moves the display to the requested row. In Formatted mode, that row is displayed. In Horizontal mode, that row is displayed at the tope of the screen. The '**row-number**' is required. The **LOCATE** command may be abbreviated as the letter **L**.

LOCATEF

LOCATEF **[column-name]**

The **LOCATEF** command moves the display to the requested column (field). The '**column-name**' must be the name of the column in the view or table that was selected for viewing in the Column Selection List. If the column name is not specified in the command, a pop-up screen will provide a list of column names for you to select from. The column name may also be specified as a partial name ending with an asterisk, which will select the first matching column name. The **LOCATEF** command may be abbreviated as **LF**.

NEXT

NEXT
N

Scrolls forward ‘1’ row.

PREV

PREV
P

Scrolls backward ‘1’ row.

PROFILE

PROFILE

The **PROFILE** command may be used in any display mode. The **PROFILE** command displays the current MAX DB2/UTIL option values, allowing you to change them. When you leave the Profile panel, the current panel will be rebuilt using any changes that you have made.

Note: Some of the option changes will not take effect immediately. For instance, if the log data set is already allocated, changing the allocation unit will have no effect until the data set are reallocated.

PRT

PRT
PR

The **PRT** command prints one or more result rows. The **PRT** command is available in any display mode except New Row Entry.

A pop-up panel will be displayed in response to the **PRT** command, allowing you to set the starting row number and number of rows to be printed. In this same panel, you can also set the print destination.

REFRESH

REFRESH

This command causes the result set to be refreshed just as if the **END** command was used, followed by the **GO** command.

RESET

RESET

This primary command resets any pending line commands.

SQL

SQL

This command displays the SQL SELECT statement generated from the Column Selection List and used to obtain the result set. Once the SQL statement is displayed, you can format it to the corresponding data definitions for COBOL, PL/I, C, C++, assembler or MAX/REXX and save the code in a file or data set. The saved code can be used to help write SQL programs.

Browse, Line Commands

FORMATTED

For character data type columns, you can enter the following line commands in the entry area on the left side of the first line of the column value:

CHAR
HEX

These commands switch the display for the column value to character or hexadecimal mode. By default, `FOR BIT DATA` and `BLOB` columns, and any character values containing undisplayable characters, are displayed in hexadecimal mode and all other values are displayed in character mode. Once a column has been set to hexadecimal or character mode, values in that column will be displayed in that mode until the mode is changed.

B
S
/

These commands cause the value to be displayed in the value browser. The value browse screen is useful for viewing long values, such as `VARCHAR`, `BLOB`, and `CLOB` data.

HORIZONTAL

B
S
/

These commands switch the display to Formatted Browse mode on the selected row.

P
PR
PRT
PRINT

This command is the same as entering the **PRT** primary command but sets the default starting row to be printed to the selected row.

2. Edit Table

The MAX DB2/UTIL Edit facility provides DB2 table editing functions using two different display modes:

1. Edit, Formatted Display
2. Edit, Horizontal Display

The Edit facility is controlled by primary commands that are used to:

- Insert, delete, repeat, and change table rows.
- Reset the display.
- Find and display table columns with specified character strings.
- Display SQL SELECT statement used to obtain result set.
- Locate a specific table row by row number.
- Locate a column name within a table row.
- Scroll through table rows.
- Display column numbers.
- Display data in hexadecimal format.
- Toggle between Browse and Edit.
- Toggle between display options.
- Print table row contents.
- **CUT** (copy) table rows to a temporary area for a subsequent **PASTE** (insert) into the same or other Edit session.
- **PASTE** (insert) segments from a temporary area that were saved with a previous **CUT** (copy) command.

Edit, Formatted Display

The following is a sample panel showing the Formatted Edit panel.

```
MAX FORMATTED EDIT TABLE MA10002.EMP_PHOTO_RESUME              Line 1 of 135
COMMAND ===>                                                 SCROLL ===> PAGE
Display:  DH - Horizontal  BR - Browse                              Row 1
Read:     N  - Next        P  - Previous     L  - Locate
Actions:  DR - Delete Row  AR - Add Row      NR - New Row
POS   COLUMN NAME *-TYPE--* N *-------------------CONTENTS--------------------*
      EMPNO       INTEGER     130
_____ FIRSTNAME   VC(7)       DOLORES
_____ MIDINIT     CHAR(1)   N M
_____ LASTNAME    VC(8)       QUINTANA
_____ GENDER      CHAR(1)     M
_____ JOB         CHAR(8)     ANALYST
      HIREDATE    DATE        28.07.1971
      SALARY      DEC(9,2)    +23800.00
00001 RESUME      CL(1201)  N     Resume:  Delores M. Quintana    Personal Info
00050                         rmation   Address:            1150 Eglinton Ave
00099                                          Mellonville, Idaho 83725  Pho
00148                         ne:              (208) 875-9933  Birthdate:
00197                             September 15, 1925  Sex:                Femal
00246                         e  Marital Status:     Married  Height:
00295                            5'2"  Weight:             120 lbs.    Departme
00344                         nt Information   Employee Number:    000130  Dept
00393                          Number:        C01  Manager:            Sally Kw
00442                         an  Position:           Analyst  Phone:
00491                             (208) 385-4578  Hire Date:          1971-07-2
00540                         8    Education   1965                Math and Eng
00589                         lish, B.A.                      Adelphi Universit
00638                         y   1960                Dental Technician
00687                                        Florida Institute of Technology
00736                         Work History   10/91 - present     Advisory Syste
```

Figure 11: Formatted Edit panel

The Formatted Edit and Browse screens are almost identical, except that in Edit mode, additional commands are available and you can overtype many of the displayed values. The New Row Entry screen is also very similar to the Formatted Edit screen, except many of the commands are not allowed.

The Formatted screen shows the values of a single row. If the values of a row do not fit on one screen, you can scroll UP and DOWN to see the rest. In Edit and Browse modes, you can also scroll LEFT and RIGHT to view the previous and next rows. In New Row Entry mode or in Edit mode, if any values have been changed, you cannot scroll to other rows until you have completed your changes using **UR** and **AR**, or you have discarded the changes using **RESET**. You can use the **END** command to leave New Row Entry mode and return to the Formatted Edit mode.

The top of the Formatted Browse screen is similar, except for the `Actions:` line. The top of the New Row Entry screen does not include the `Display:` and `Read:` lines. Once a change has been made to a column value in Edit mode, the `Display:` line will say `(frozen)` and the `Read:` line will be blank.

The upper right corner of the screen will show the visible screen rows. The COMMAND area allows entry of primary commands and the SCROLL area allows the default scroll amount to be changed.

The Display: line shows the available screen format commands. The Read: line shows the row positioning commands. The Actions: line shows the commands available for updating the database.

The POS field allows entry of line commands for character type values. If the value requires more than one line to display, the POS field of the second and subsequent lines will contain the position of that portion of the value.

The COLUMN NAME field show the name of the column and the *-TYPE--* field shows the actual data type. The data type names may be abbreviated to fit. You may see DEC or DC for DECIMAL, GRPH for GRAPHIC, VC for VARCHAR, VG for VARGRAPHIC, BL for BLOB, CL for CLOB or DB for DBCLOB. The length may also be shortened to K (kilobytes) or M (megabytes).

The N field is present if any selected columns were defined without the **NOT NULL** clause, so it may contain a null value. If the column value is **NULL**, the letter Y will be displayed in the N field and the CONTENTS field will be blank. If the column value is **NOT NULL**, the letter N will be displayed in the N field and the value will be displayed in the CONTENTS field. In Edit or New Row Entry mode, you can overtype the N field with a Y or N to indicate that the column value is null or not null.

The CONTENTS field shows the value of the column for the current row. A numeric value is shown for INTEGER, SMALLINT, DECIMAL, FLOAT or DOUBLE columns. Date/time columns (DATE, TIME and TIMESTAMP) are shown as character values. ROWID column values are shown in hexadecimal. The remaining data types (CHAR, GRAPHIC, VARCHAR, VARGRAPHIC, BLOB, CLOB and DBCLOB) are shown in either character or hexadecimal format.

In Edit and New Row Entry mode, you can normally overtype the contents of the field to change the value of that column of the current row. You cannot change the value of ROWID or UNIQUE columns. When you overtype the values of numeric or date/time fields, the new value will be checked to be valid for the data type, but any constraints are not checked until a database update command (**AR** or **UR**) is entered.

If a character field contains undisplayable characters and the column is being displayed in character mode, the undisplayable characters are shown as a dot and the value may not be overtyped. If a character field is displayed in hexadecimal, an indicator line (-------- Hexadecimal --------) will be displayed above the value and in Edit or New Row mode, you can overtype the hexadecimal digits to change the value.

If a character value requires more lines than the limit set in the DB2/UTIL Profile Options panel, an indicator line (-------- Truncated --------) will be displayed above the first value line. If the value is truncated, you cannot change the value by overtyping it.

You can display the entire value of any character column by invoking the value browser or editor by entering the **B** or **E** line command in the POS field for the value. The value editor also allows you to change the length of varying character strings. The **UC** and **LC** commands may be used in the POS field to translate characters to uppercase or lowercase.

You can also change the values, including the **NULL**/**NOT NULL** state, by using the **CHANGE** command.

Edit, Horizontal Display

The following is a sample panel showing the Horizontal Edit panel.

```
MAX HORIZONTAL EDIT TABLE MA10002.EMP_PHOTO_RESUME              1 to 72 of 486
COMMAND ===>                                                  SCROLL ===> PAGE
Display:  DF - Formatted   BR - BRowse                       L - Locate by row
Type:  INTEGER      VARCHAR(12)    CHAR(1)  VARCHAR(15)      CHAR(1)  CHAR(8)
ROW--- EMPNO------  FIRSTNAME---   MIDINIT  LASTNAME-------  GENDER-  JOB-----
000001         130  DOLORES         M        QUINTANA        M        ANALYST
000002         140  HEATHER         A        NICHOLLS        F        ANALYST
000003         150  BRUCE          <null>    ADAMSON         M        DESIGNER
000004         190  JAMES           H        WALKER          M        DESIGNER
******************************** Bottom of data ********************************
```

Figure 12: Horizontal Edit panel

The Horizontal Edit and Browse screens are identical, except for the valid commands. Each screen displays as many available rows as there is room. The screen may be scrolled UP and DOWN to display other rows in the result set. If the number and width of the selected columns is wider than the screen, you can also scroll LEFT and RIGHT to view the data.

The upper right corner of the screen will show the visible screen columns. The COMMAND area allows entry of primary commands and the SCROLL area allows the default scroll amount to be changed. The third line shows the available screen format commands.

The top row of the column header normally shows the defined data type. If the **COLS** command has been entered for the column, the column scale will be displayed instead of the data type. For character type columns, the headings in this row may be overtyped with column commands.

The second row of the column header shows the column names.

The remaining lines of the screen show as many result rows as will fit on the screen. At the left side of each line is the relative row number of the result set row. These numbers are assigned as each row is retrieved or added and will not change if rows are deleted. The **LOCATE** command may be used to position the screen by result row number.

Character (CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, BLOB, CLOB and DBCLOB) values are displayed in either character or hexadecimal format. You can switch a column between character and hexadecimal format by entering either **CHAR** or **HEX** in the column heading, or you can switch all columns by using the **CHAR** or **HEX** primary commands.

Numeric (INTEGER, SMALLINT, DECIMAL, FLOAT and DOUBLE) values are displayed as numbers, date/time (DATE, TIME and TIMESTAMP) values are displayed as character strings and ROWID values are displayed in hexadecimal. Integer and decimal values are right justified. All other values are left justified.

If a character column is longer than the limit set in the DB2/UTIL Profile Options panel, the values may be truncated. Truncation is indicated by the '>' character at the right end of the column heading or value.

In Horizontal Edit and Browse mode, the column headings of character columns containing the data type can be overtyped with a command to switch the display format for that column between character and hexadecimal. The following column commands are supported:

CHAR can be used to switch the column format from hexadecimal to character

HEX can be used to switch the column format from character to hexadecimal

These commands are equivalent to the line commands by the same name in Formatted Edit and Browse mode. Once the **CHAR** or **HEX** command is entered in either display mode for a column, the format of the column will be used for the remainder of the Edit or Browse session until changed.

Edit, Primary Commands

Command	Description	Formatted	Horizontal
AR	Add row.	YES	NO
BROWSE	Browse tables.	YES	YES
CANCEL	Cancel action.	YES	YES
CHANGE	Change rows.	YES	YES
CHAR	Display character.	YES	YES
COLS	Specify columns.	NO	YES
COMMIT	Commit changes.	YES	YES
COUNT	Count rest of result set.	YES	YES
CUT	Copy row(s).	YES	YES
DATA	Data-dependent display.	YES	YES
DF	Display formatted mode.	YES	YES
DH	Display horizontal mode.	YES	YES
DR	Delete row(s).	YES	NO
END	End session.	YES	YES
FIND	Find data.	YES	YES
HEX	Display hexadecimal.	YES	YES
LOCATE	Locate row by row-number.	YES	YES
LOCATEF	Locate column name.	YES	YES
NEW	Add new row.	YES	YES
NEXT	Scroll forward.	YES	NO

Command	Description	Formatted	Horizontal
PASTE	Insert **CUT** row(s).	YES	YES
PREV	Scroll backward.	YES	NO
PROFILE	Display profile parameters.	YES	YES
PRT	Print row(s).	YES	YES
REFRESH	Retrieve result set.	YES	YES
RESET	Reset error condition.	YES	YES
SAVE	Commit changes.	YES	YES
SQL	Display SQL SELECT statement.	YES	YES
UPPER	Uppercase character mode.	YES	YES
UR	Update row.	YES	NO

AR

AR
ADD

This command adds the currently displayed row to the table. This command may be used from the Formatted Edit screen or the New Row Entry screen. This is the only valid update command on the New Row Entry screen.

BROWSE

BR
BROWSE

This command changes the display to Browse. The Formatted/Horizontal mode is not changed by this command.

CANCEL

CANCEL

The **CANCEL** command ends the Edit or Browse session and attempts to undo any changes that have not been saved.

CHANGE

```
CHANGE    [from-value to-value [column [start [end]]]] [NEXT|PREV|FIRST|LAST]
>RCHANGE  [NEXT|PREV|FIRST|LAST]
```

The **CHANGE** command changes an occurrence of a value in the result set to a different value. The **CHANGE** command can be used both Horizontal and Formatted Edit mode, although the display will switch to Formatted mode when the '**from-value**' is found. The **CHANGE** command can be abbreviated to the letter **C**.

The **>RCHANGE** command repeats a previous **CHANGE** action. **>RCHANGE** is usually invoked by the PF6 key. (The leading ">" is required by ISPF since **RCHANGE** is normally disabled outside of the PDF editor.)

If the values are not specified, the **CHANGE** and **>RCHANGE** commands both will repeat a previous **CHANGE** action. Both of these versions of the **CHANGE** command can be qualified with any of the key words: **NEXT**, **PREV**, **FIRST** and **LAST** to indicate the direction of the search for the '**from-value**'.

NEXT	The **NEXT** key word starts the search from the cursor location, changing the next occurrence of the value. For a **CHANGE** command with values, the default search direction is **NEXT**.
PREV	The **PREV** key word also starts the search from the cursor location but locates the previous occurrence of the value from the cursor.
FIRST	The **FIRST** key word starts the search from the start of the result set, finding the first occurrence.
LAST	The **LAST** key word starts the search from the end of the result set finding the last occurrence. If a **CHANGE** with **FIRST** or **LAST** is repeated without a direction, the default direction is **NEXT** or **PREV** respectively.

The '**from-value**' may be preceded by the word **NOT** which indicates that the value is considered to be found when it does not match.

The '**from-value**' may be specified in several ways:

NULL	This searches for a null value, whereas **NOT NULL** searches for values that are not null. Only columns not defined as **NOT NULL** are searched.
operator number	This performs comparisons on values in numeric columns.
string	This performs character comparisons. Only non-numeric columns are searched and null values are not compared. **NOT** string will match any value that does not contain the string.
" " or **' '**	This performs searches varying length columns for values of a matching length.

The '**to-value**' is like the '**from-value**' except that the **NOT** key word and the numeric comparison operator are not allowed. The '**to-value**' may be null to change a value to null. If the '**from-value**' is a numeric comparison, the '**to-value**' must also be numeric. Character '**to-values**' are used as-is and are not translated to uppercase.

Examples of CHANGE commands:

CHANGE NOT NULL " "	Changes **NOT NULL** values to the empty string.
CHANGE " " NULL	Changes zero-length strings to **NULL**.
CHANGE < 0 1	Changes negative numbers to "1".
CHANGE T ' ' " "	Removes blanks.

The syntax for changing a numeric value is:

CHANGE [NOT] operator from-value to-value [column] [NEXT|PREV|FIRST|LAST]

Numeric columns (INTEGER, SMALLINT, DECIMAL, FLOAT and DOUBLE) are searched using a comparison operator.

Note: Without an operator, a character search of non-numeric columns will be performed. For example, **F** 0 will not search numeric columns.

Character from-values may be specified in several ways:

Tttt **T'tttt'** **T"tttt"**	These perform a case-insensitive search for the characters.
'cccc' **"cccc"** **C'cccc'** **C"cccc"**	These perform a case-sensitive search for the characters.
X'xxxx' **X"xxxx"**	These search for a hexadecimal value. An even number of hexadecimal digits must be specified.

The characters must be enclosed in quotes if any of the characters are quotes (**"** or **'**) or blanks, or if the string starts with a comparison operator (=, >, <) or is a key word (**NOT**, **NULL**, **NEXT**, **PREV**, **FIRST**, **LAST**). Quoted strings cannot contain the enclosing quote character. The **NOT** key word causes a match for values that do not contain the characters.

The empty '**from-value**' string (**" "** or **' '**) has a special meaning. It is used to match strings with a specific length, as described below.

To change varying length columns (VARCHAR, VARGRAPHIC, ROWID, BLOB, CLOB and DBCLOB) from values with a specific length or range of lengths, specify the '**from-value**' as the empty string (**" "** or **' '**). Without the column name, all selected varying length columns will be searched for zero-length values.

Note: A zero-length string is not the same as a null value.

To search a varying length column for values with a specific length, specify the column name and start position. This will search for values in that column with the specified length. To search a varying length column for values with a range of lengths, specify the column name and both the start and end positions.

Using the empty string as a '**from-value**' will skip numeric columns, fixed length columns (DATE, TIME, TIMESTAMP, CHAR and GRAPHIC) and **NULL** values. The **NOT** key word will match values that do not have the specified length.

You can limit the search for a value to a specific database column by following the '**to-value**' with a column name. The named column must be one of the selected columns. The data type of the column must match the '**from-value**'. The column must not be defined as **NOT NULL** to search for null or not null, the column must be numeric for a numeric comparison, the column must be non-numeric for a character search and the column must be varying for a length comparison.

For character searches, you can further limit the search to a specific position within the value or a subset of the value. If only start is specified, the comparison is performed on the portion of values from the starting position for a length of the search value. If both start and end are specified, the range of characters between the starting and ending positions of the column values are searched. No padding is performed, so values shorter than the search range will not match. If the characters are found in more than one place in the column value, the cursor will be placed at the first (or last) found position within the value editor. You can then repeat the **CHANGE** as needed.

CHAR

CHAR [*|column]

The **CHAR** command sets the display format for one or more character columns to character format and the entry format to mixed-case. This command is available in all display modes. If an asterisk is specified for the column name, this is equivalent to entering the **CHAR** command into every line in Formatted mode or every column in Horizontal mode. If a column name is specified, only that column's display format will be changed. If no column name is specified, a list of the character column names will be displayed, allowing you to set the display format for each.

COLS

COLS [ON|OFF]

The **COLS** command is available in Horizontal mode to switch the column heading between the data type and a column scale. If neither **ON** nor **OFF** is specified, the heading is switched to the other format.

COMMIT

COMMIT

This command causes any uncommitted changes to be committed. This will cause database locks held by these changes to be released. These changes may still be undone by a later **CANCEL** command.

COUNT

COUNT

This command causes the rest of the result set to be retrieved and subsequently displays the number of rows returned by the query.

CUT

CUT **[rows] [R]**

The **CUT** command copies one or more rows to the "**CUT** buffer, data set or UNIX file". These rows can be inserted later into the same or another table using the **PASTE** command. The **CUT** command is available in any display mode except New Row Entry. The **PASTE** command is only available in Edit mode.

The **rows** and **R** parameters are optional. If rows is not specified, only one row will be cut. In Formatted mode, rows are copied starting with the currently visible row. In Horizontal mode, rows are copied starting with the row at the top of the screen. If '**R**' is specified, the copied rows will replace any previously copied rows in the **CUT** buffer, data set or UNIX file. If '**R**' is not specified, the copied rows will be added to any previously copied rows as long as the previous rows are from a table or view with selected columns of the same name and data type.

DATA

DATA **[* | column]**

The **DATA** command sets the display format for one or more character columns to data-dependent format. This command is available in all display modes. If an asterisk is specified for the column name, this is equivalent to entering the **DATA** command into every line in Formatted mode or every column in Horizontal mode. If a column name is specified, only that column's display format will be changed. If no column name is specified, a list of the character column names will be displayed, allowing you to set the display format for each.

DF

DF

In Horizontal mode, this command changes the display to Formatted. The Edit/Browse mode is not changed by this command.

DH

DH

In Formatted mode, this command changes the display to Horizontal. The Edit/Browse mode is not changed by this command.

DR

```
DR        [rows]
DEL       [rows]
DELETE    [rows]
```

This command deletes one or more rows from the table. If '**rows**' is not specified, one row will be deleted. Rows are deleted starting with the currently displayed row.

END

```
END
```

The **END** command ends the edit or browse session. For edit, any changes are committed and the **CANCEL** log will be discarded. On the New Row Entry screen, the **END** command will end any pending **NEW** or **PASTE** primary command, or **R, RR** line command, and return to the Formatted Edit screen.

FIND

```
FIND      [value [column [start [end]]]]  [NEXT|PREV|FIRST|LAST]
>RFIND    [NEXT|PREV|FIRST|LAST]
```

The **FIND** command locates an occurrence of a value within the result set. The **FIND** command can be used in both browse and edit modes and in both Horizontal and Formatted modes, although the display will switch to Formatted mode when a value is found. The **FIND** command can be abbreviated to the letter **F**.

The **>RFIND** command repeats a previous **FIND** action. **>RFIND** is usually invoked by the PF5 key. (The leading '**>**' is required by ISPF since **RFIND** is normally disabled outside of the PDF editor.)

If the find value is not specified, the **FIND** and **>RFIND** commands both will repeat the **FIND** action of a previous **FIND** or **CHANGE**. Both of these versions of the **FIND** command can be qualified with any of the key words: **NEXT**, **PREV**, **FIRST** or **LAST** to indicate the direction of the search.

NEXT The **NEXT** key word starts the search from the cursor location, finding the next occurrence of the value. For a **FIND** command with a value, the default search direction is **NEXT**.

PREV The **PREV** key word also starts the search from the cursor location but locates the previous occurrence of the value from the cursor.

FIRST The **FIRST** key word starts the search from the start of the value, finding the first occurrence.

LAST The **LAST** key word starts the search from the end of the value finding the last occurrence. If a **FIND** with **FIRST** or **LAST** is repeated without a direction, the default direction is **NEXT** or **PREV** respectively.

The value to be found may be preceded by the word **NOT** which indicates that the value is considered to be found when it does not match.

The value may be specified in several ways:

NULL	This searches for a null value, whereas **NOT NULL** searches for values that are not null. Only columns not defined as **NOT NULL** are searched.
operator number	This performs comparisons on values in numeric columns.
string	This performs character comparisons. Only non-numeric columns are searched and null values are not compared. **NOT** string will match any value that does not contain the string.
" " or **' '**	This performs searches varying length columns for values of a matching length.

The syntax for searching for a numeric value is:

FIND **[NOT] operator value [column] [NEXT|PREV|FIRST|LAST]**

Numeric columns (`INTEGER`, `SMALLINT`, `DECIMAL`, `FLOAT` and `DOUBLE`) are searched using a comparison operator and numeric value. The operator must be one of: =, <, >, <>, >= or <=. The **NOT** key word reverses the comparison, thus **NOT** `=0.0` and `<> 0` both search for a number that is not zero. Blanks are optional between the operator and the number. The number may include a sign, decimal point and digits and an exponent. Null values are not compared.

Note: Without an operator, a character search of non-numeric columns will be performed. For example, **F** `0` will not search numeric columns.

Character values may be specified in several ways:

Tttt **T'tttt'** **T"tttt"**	These perform a case-insensitive search for the characters.
'cccc' **"cccc"** **C'cccc'** **C"cccc"**	These perform a case-sensitive search for the characters.
X'xxxx' **X"xxxx"**	These search for a hexadecimal value. An even number of hexadecimal digits must be specified.

The characters must be enclosed in quotes if any of the characters are quotes (**"** or **'**) or blanks, if the string starts with a comparison operator (=, >, <), or if they are a key word (**NOT**, **NULL**, **NEXT**, **PREV**, **FIRST**, **LAST**). Quoted strings cannot contain the enclosing quote character. The **NOT** key word causes a match for values that do not contain the characters.

The empty string (**""** or **' '**) has a special meaning. It is used to search for strings with a specific length, as described below.

To search varying length columns (`VARCHAR`, `VARGRAPHIC`, `ROWID`, `BLOB`, `CLOB`, and `DBCLOB`) for values with a specific length or range of lengths, specify the search value as the empty string (**""** or **' '**). Without the column name, all selected varying length columns will be searched for zero-length values.

Note: A zero-length string is not the same as a null value.

To search a varying length column for values with a specific length, specify the column name and start position. This will search for values in that column with the specified length. To search a varying length column for values with a range of lengths, specify the column name and both the start and end positions.

Using the empty string as a search value will skip numeric columns, fixed length columns (`DATE`, `TIME`, `TIMESTAMP`, `CHAR` and `GRAPHIC`) and null values. The **NOT** key word will match values that do not have the specified length.

You can limit the search for a value to a specific database column by following the search value with a column name. The named column must be one of the selected columns. The data type of the column must match the search value. The column must not be defined as **NOT NULL** to search for 'null' or 'not null', the column must be numeric for a numeric comparison, the column must be non-numeric for a character search and the column must be varying for a length comparison.

For character searches, you can further limit the search to a specific position within the value or a subset of the value. If only start is specified, the comparison is performed on the portion of values from the starting position for a length of the search value. If both start and end are specified, the range of characters between the starting and ending positions of the column values are searched. No padding is performed, so values shorter than the search range will not match. If the characters are found in more than one place in the column value, the cursor will be placed at the first (or last) found position within the value editor or browser. You can then repeat the **FIND** as needed.

HEX

HEX [*|column|ON|OFF]

The **HEX** command sets the display format for one or more character columns to hexadecimal format. This command is available in all display modes. If an asterisk is specified for the column name, this is equivalent to entering the **HEX** command into every line in Formatted mode or every column in Horizontal mode. If a column name is specified, only that column's display format will be changed. If no column name is specified, a list of the character column names will be displayed, allowing you to set the display format for each. **HEX ON** is equivalent to '**HEX ***' and '**HEX OFF**' is equivalent to '**CHAR ***'.

LOCATE

LOCATE row-number

The **LOCATE** command moves the display to the requested row. In Formatted mode, that row is displayed. In Horizontal mode, that row is displayed at the top of the screen. The '**row-number**' is required. The **LOCATE** command may be abbreviated as the letter **L**.

LOCATEF

LOCATEF [column-name]

The **LOCATEF** command moves the display to the requested column (field). The '**column-name**' must be the name of the column in the view or table that was selected for viewing in the Column Selection List. If the column name is not specified in the command, a pop-up screen will provide a list of column names for you to select from. The 'select from list' method must be used when browsing a View. The column name may also be specified as a partial name ending with an asterisk, which will select the first matching column name. The **LOCATEF** command may be abbreviated as **LF**.

NEW

NEW
NR

The **NEW** or **NR** command causes the New Row Entry screen to be displayed. You can enter the column values for the new row, and then add it to the database using the **ADD** command.

NEXT

NEXT
N

Scrolls forward one row.

PASTE

PASTE [D|K]

The **PASTE** command inserts rows from a data set, UNIX file or the **CUT** buffer. These rows must have been previously **CUT** from a table with compatible column names and data types. The optional '**K**' or '**D**' parameter may be specified to cause the **PASTE** command to keep (**K**) or delete (**D**) the data set, UNIX file or **CUT** buffer.

PREV

PREV [P]

Scrolls backward one row.

PROFILE

PROFILE

The **PROFILE** command may be used in any display mode. The **PROFILE** command displays the current MAX DB2/UTIL option values, allowing you to change them. When you leave the DB2/UTIL Profile Options panel, the current panel will be rebuilt using any changes that you have made.

Note: Some of the option changes will not take effect immediately. For instance, if the log data set is already allocated, changing the allocation unit will have no effect until the data set needs to be reallocated.

PRT

PRT
PR

The **PRT** command prints one or more result rows. The **PRT** command is available in any display mode except New Row Entry.

A pop-up panel will be displayed in response to the **PRT** command, allowing you to set the starting row number and number of rows to be printed. In this same panel, you can also set the print destination.

REFRESH

REFRESH

This command causes the result set to be refreshed just as if the **END** command was used, followed by the **GO** command. However, the **CANCEL** log is not discarded, allowing the **CANCEL** command to be used to undo any unsaved changes.

RESET

RESET

This primary command resets any pending line commands. In Formatted Edit mode, it also causes any changes to the current row contents to be discarded.

SAVE

SAVE

This command causes any uncommitted changes to be committed. This will cause database locks held by these changes to be released. The **CANCEL** log will be discarded and the changes will not be undone by a later **CANCEL** command.

SQL

SQL

This command displays the SQL SELECT statement generated from the Column Selection List and used to obtain the result set. Once the SQL statement is displayed, you can format it and the corresponding data definitions for COBOL, PL/I, C, C++, assembler or MAX/REXX and save the code in a file or data set. The saved code can be used to help write SQL programs

UPPER

UPPER [*|column]

The **UPPER** command sets the display format for one or more character columns to character and the entry format to uppercase. This command is available in all display modes. If an asterisk is specified for the column name, this is equivalent to entering the **UPPER** command into every line in Formatted mode or every column in Horizontal mode. If a column name is specified, only that column's display format will be changed. If no column name is specified, a list of the character column names will be displayed, allowing you to set the display format for each.

UR

UR
UPD
UPDATE

This command updates the current row in the table. You must change the contents of one or more column values. These changed values will be used to update the row. Once values have been changed on the edit screen, you must update the row, add the row or discard the changes before you can display another row or leave the editor. Use the **RESET** command to discard the changes.

Edit, Line Commands, Formatted

CHAR, DATA, HEX, UPPER

These commands switch the display and entry formats for the column value to character, data-dependent or hexadecimal. The **DATA** command set the display to data-dependent mode in which, any character values containing undisplayable characters are displayed in hexadecimal and all other values are displayed in character mode. The **CHAR** and **HEX** commands set the display format for a column to be always either character or hexadecimal. The **UPPER** command also sets the display to character mode, but also causes any updates to be translated automatically to uppercase.

B, E, S, /

These commands cause the value to be displayed in the value browser or editor. '**B**' causes the value browser screen to be displayed and '**E**', '**S**' and '/' all cause the value edit screen to be displayed. The value edit screen is useful for viewing or changing long values, such as `VARCHAR`, `BLOB` and `CLOB` data.

RESET

The **RESET** command discards any changes made to the data value, restoring the value to the contents from the result set.

LC

This command translates all uppercase characters in the value to lowercase. The set of characters translated depends on the terminal type. This command is not valid in browse mode.

UC

This command translates all lowercase characters in the value to uppercase. The set of characters translated depends on the terminal type. This command is not valid in Browse mode.

Edit, Line Commands, Horizontal

E, B, S, /

These commands switch the display to Formatted Edit or Browse mode on the selected row. '**S**' and '/' switch to Formatted without switching the Edit/Browse mode.

P, PR, PRT, PRINT

This command is the same as entering the **PRT** primary command but sets the default starting row to be printed to the selected row.

D, DD

This command deletes one or more rows from the table. The **DD** command must be paired with another **DD** command to delete all rows from the first selected row to the second selected row. These commands are invalid in Browse mode.

I

This command causes the New Row Entry screen to be displayed, allowing a new row to be inserted into the table. The '**I**' line command is equivalent to entering the **NEW** or **NR** primary command. This command is invalid in Browse mode.

R, RR

This command duplicates (repeats) one or more rows in the table. The **RR** command must be paired with another **RR** command to duplicate all rows between the first and second selected row. This command is invalid in Browse mode.

Browse and Edit Table Values

MAX DB2/UTIL provides browsing and editing functions for individual character column values. The value editor or browser can be invoked on a value by entering an '**E**', '**B**', '**S**' or '/' line command. This is done in the prefix area, on the first line of the value, in the Formatted Edit or Formatted Browse screen. MAX DB2/UTIL provides four different display modes for a value:

1. Edit, Single Line mode
2. Edit, Fixed Width mode
3. Browse, Single Line mode
4. Browse, Fixed Width mode

The Edit and Browse versions of both the Single Line Display mode and the Fixed Width Display mode are identical with one exception. In Browse mode, the ability to overtype the value, and commands that change the value, are not available.

The Value Browse and Edit Facility is controlled by primary commands that are used to:

- Scroll through the data.
- Display data in hexadecimal or character format.
- Change the length of a value.
- Convert the value to uppercase or lowercase.
- Find and change a specific character string within a value.

Edit, Single Line Display

The following is a sample panel showing the Edit, Single Line Display.

```
MAX Edit of column BMP_PHOTO                                    Column 1 to 80
 COMMAND ===> _____________________________________________ SCROLL ===> PAGE
 Data type. : BLOB(43690)          NEW LENGTH ===> _________ WIDTH ===> 00000
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
 ------------------------------------------------------------------------------
.(.............K...I...........w.....................................
44AA0000003000200D000C000000000007A001000100000000000000000000000000000000000
2DAA000000064008000200090001080000046003B003B000000000000000656064A07A707AA0B770A7
 ------------------------------------------------------------------------------
 ******************************** Bottom of data ********************************
```

Figure 13: Single Line Display

The Single Line display shows the value as one continuous line across the screen. If the value is too long to fit on the screen, you can scroll LEFT and RIGHT. Single Line Display mode is indicated by `WIDTH ===> 00000`. You can change the display mode to Fixed Width mode by entering a non-zero number in the `WIDTH` screen field.

In both Single Line Edit mode and Fixed Width Edit mode of a VARCHAR, VARGRAPHIC, BLOB, CLOB or DBCLOB column, you can change the length of the value by entering a new length in the NEW LENGTH field. This field is not displayed for CHAR and GRAPHIC columns since those column types are defined with a fixed length. The length of varying length columns can be changed to be anywhere from zero to the maximum defined column length. If the value length is increased, the value will be padded with blanks if the display is in character mode or with X'00' if the display is in hexadecimal mode.

Edit, Fixed Width Display

The following is a sample panel showing the Edit, Fixed Width Display.

```
MAX Edit of column RESUME                                    Line 1 to 16 of 16
 COMMAND ===> _____________________________________________ SCROLL ===> PAGE
 Data type. : CLOB(1206)             NEW LENGTH ===> _________ WIDTH ===> 00080
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----
    Resume:  Heather A. Nicholls    Personal Information   Address:            8
44 Don Mills Ave                      Mellonville, Idaho 83734  Phone:
    (208) 610-2310  Birthdate:          January 19, 1946  Sex:                Fe
male  Marital Status:     Single  Height:             5'8"  Weight:
130 lbs.    Department Information   Employee Number:    000140  Dept Number:
     C01  Manager:            Sally Kwan  Position:           Analyst  Phone:
           (208) 385-1793  Hire Date:          1976-12-15    Education   1972
             Computer Engineering, Ph.D.                      University of Wash
ington   1969               Music and Physics, B.A.                      Vassar
 College    Work History   2/83 - present      Architect, OCR Development
              Designing the architecture of OCR products.   12/76 - 1/83
 Text Programmer                      Optical character recognition (OCR) progra
mming in                      PL/I.   9/72 - 11/76        Punch Card Quality Ana
lyst                     Checking punch cards met quality specifications.    In
terests   o   Model railroading  o   Interior decorating  o   Embroidery  o   Kn
itting
 ******************************** Bottom of data ********************************
```

Figure 14: Fixed Width Display

The Fixed Width Display shows the value as one or more lines, each with a specified width. Fixed Width Display mode is indicated by a WIDTH field that contains a number other than zero. You can change the display mode to Single Line mode by entering all zeros in the WIDTH screen field.

The value will be divided into lines of the WIDTH field value and each of the lines will be displayed on the screen. Only the last line may shorter than the specified width. If there are more value lines than can be displayed on the screen, you can scroll up and down through the lines. Likewise, the screen can be scrolled LEFT and RIGHT if the specified width is wider than the screen.

Primary Commands

Command	Description	Browse	Edit
CANCEL	Cancel changes.	YES	YES
CHANGE	Change strings.	NO	YES
CHAR	Display character.	YES	YES
END	End session.	YES	YES
FIND	Find string.	YES	YES
HEX	Display hexadecimal.	YES	YES
LC	Convert to lowercase.	NO	YES
UC	Convert to uppercase.	NO	YES
UPPER	Uppercase character mode.	YES	YES

CANCEL

CANCEL

The **CANCEL** command ends the edit or browse session for the value and returns to the Formatted Edit or Browse session on the row. Any changes made during the canceled Edit session are discarded and the value is restored to the value it had upon entry to the edit session.

Note: The **CANCEL** command does not reset the column's value to the original value in the database. **CANCEL** only resets the column's value to what it had been before entering the value editor

CHANGE

CHANGE [from-value to-value [start [end]] [NEXT|PREV|FIRST|LAST]
>RCHANGE [NEXT|PREV|FIRST|LAST]

The **CHANGE** command changes an occurrence of one or more characters within the displayed value to a different set of characters. The **>RCHANGE** command repeats a previous **CHANGE** action. **>RCHANGE** is usually invoked by the PF6 key. (The leading '**>**' is required by ISPF since **RCHANGE** is normally disabled outside of the PDF editor.) The **CHANGE** command can be abbreviated to the letter **C**.

If the values are not specified, the **CHANGE** and **>RCHANGE** commands both will repeat a previous **CHANGE** action. The only difference is that **CHANGE** will only repeat a previous action from the same edit screen. If there is no previous action in the same edit screen, **>RCHANGE** will attempt to repeat a **CHANGE** action from the table edit session. Both of these versions of the **CHANGE** command can be qualified with any of the following key words: **NEXT**, **PREV**, **FIRST** or **LAST** to indicate the direction of the search for the "**from-value**".

NEXT The **NEXT** key word starts the search from the cursor location, changing the next occurrence of the value. For a **CHANGE** command with values, the default search direction is **NEXT**.

PREV The **PREV** key word also starts the search from the cursor location but changes the previous occurrence of the value.

FIRST The **FIRST** key word starts the search from the start of the value, changing the first occurrence.

LAST The **LAST** key word starts the search from the end of the value changing the last occurrence. If a **CHANGE** with **FIRST** or **LAST** is repeated without a direction, the default direction is **NEXT** or **PREV** respectively.

The '**from-value**' may be specified in a number of ways:

Tttt
T'tttt'
T"tttt" These perform a case-insensitive search for the characters.

'cccc'
"cccc"
C'cccc'
C"cccc" These perform a case-sensitive search for the characters.

X'xxxx'
X"xxxx" These perform a search for the hexadecimal value.

The '**to-value**' may also be specified in any of these ways; however, there is no case sensitivity in the '**to-value**' string. The '**to-value**' is used exactly as specified. If the '**from-value**' and '**to-value**' are not the same length, then the effect depends on whether the value has a variable length. For a `CHAR` or `GRAPHIC` fixed length value, the value will be truncated or padded as needed after the change. For a `VARCHAR`, `VARGRAPHIC`, `BLOB`, `CLOB` or `DBCLOB`, the length of the value will be changed.

If the values are specified, then the start and end columns may also be specified. If only the start column is specified, then only that column is searched. If both the start and end columns are specified, then the characters within that range of columns are searched.

CHAR

CHAR

The **CHAR** command sets the display to character mode and the entry mode to mixed-case. Each line is displayed only as a character value. In the browser, any non-displayable value will be shown as a dot. In the editor, if there are any non-displayable characters, you will not be able to change the display to character mode.

END

END

The **END** command ends the Edit or Browse session for the value, and returns to the Formatted Edit or Browse session on the row.

Note: The **END** command does not save the changes to the database. You must still use the **UR** or **AR** command to save the changes once you are done changing the row.

FIND

FIND **[value to-value [start [end]] [NEXT|PREV|FIRST|LAST]**
>RFIND **[NEXT|PREV|FIRST|LAST]**

The **FIND** command locates an occurrence of one or more characters within the displayed value. The **FIND** command can be used in both browse and edit mode. The **>RFIND** command repeats a previous **FIND** action. **>RFIND** is usually invoked by the PF5 key. (The leading '**>**' is required by ISPF since **RFIND** is normally disabled outside of the PDF editor.) The **FIND** command can be abbreviated to the letter **F**.

If the find value is not specified, the **FIND** and **>RFIND** commands both will repeat a previous **FIND** action. The only difference is that **FIND** will only repeat a previous action from the same browse or edit screen. If there is no previous action in the same Browse or Edit screen, **>RFIND** will attempt to repeat a **FIND** action from the table/view edit/browse. Both of these versions of the **FIND** command can be qualified with any of the keywords: **NEXT**, **PREV**, **FIRST** or **LAST** to indicate the direction of the search.

NEXT The **NEXT** key word starts the search from the cursor location, finding the next occurrence of the value. For a **FIND** command with a value, the default search direction is **NEXT**.

PREV The **PREV** key word also starts the search from the cursor location but locates the previous occurrence of the value from the cursor.

FIRST The **FIRST** key word starts the search from the start of the value, finding the first occurrence.

LAST The **LAST** key word starts the search from the end of the value finding the last occurrence. If a **FIND** with **FIRST** or **LAST** is repeated without a direction, the default direction is **NEXT** or **PREV** respectively.

The value to be found may be specified in a number of ways:

Tttt **T'tttt'** **T"tttt"**	These perform a case-insensitive search for the characters.
'cccc' **"cccc"** **C'cccc'** **C"cccc"**	These perform a case-sensitive search for the characters.
X'xxxx' **X"xxxx"**	These perform a search for the hexadecimal value.

If the search value is specified, then the start and end columns may also be specified. If only the start column is specified, then only that column is searched. If both the start and end columns are specified, then the characters within that range of columns are searched.

HEX

HEX **[ON|OFF]**

The **HEX** command sets the display to hexadecimal mode. Each line of the value is displayed in hexadecimal with the translated character value just above. In the editor, you can change the hexadecimal values but not the character translation. The **HEX** command may be followed by **ON** or **OFF**. '**HEX OFF**' is equivalent to **CHAR**.

LC

LC

This command translates all uppercase letters to lowercase. The set of characters translated depends on the terminal type.

UC

UC

This command translates all lowercase letters to uppercase. The set of characters translated depends on the terminal type.

UPPER

UPPER

This command sets the display to character mode and the entry mode to uppercase. Each line is displayed only as a character value. In the browser, any non-displayable value will be shown as a dot. In the editor, if there are any non-displayable characters, you will not be able to change the display to character mode.

3. DB2 Utilities

This option allows you to interactively build JCL to invoke the DB2 RUNSTATS utility for tablespaces or indexes. You can create the list of tablespaces or indexes from a list of names matching a tablespace or index pattern or from the tablespaces or indexes related to a list of tables matching a pattern.

```
MAX DB2/UTIL ----------------- DB2 UTILITY MENU ----------------- MAX DB2/UTIL
COMMAND ===> ____________________________________________________________________

Select one of the following. Then press Enter.
_    1. RUNSTATS TABLESPACE - select by TABLESPACE
     2. RUNSTATS TABLESPACE - select by TABLE
     3. RUNSTATS INDEX - select by INDEX
     4. RUNSTATS INDEX - select by TABLE

Specify table or index:
OWNER ID . . . . . . %_______          (Wild cards (% or _) may be used)
TABLE/INDEX NAME . . EMP%_____________________________________________________
                     _____________________________________________________________
                     _______________    (Wild cards (% or _) may be used)
or tablespace:
DATABASE . . . . . . %_______          (Wild cards (% or _) may be used)
TABLESPACE NAME. . . %_______          (Wild cards (% or _) may be used)
```

Figure 15: DB2 Utility Menu

For option 1. RUNSTATS TABLESPACE - select by TABLESPACE, you will need to enter a database name or pattern and a tablespace name or pattern. For the other utility options, you will need to enter an owner name or pattern and a table or index name or pattern.

For the OWNER ID, TABLE/INDEX NAME, DATABASE, or TABLESPACE NAME, you can either specify the name or let MAX DB2/UTIL search for matching names using a pattern. In a pattern, a percent sign (%) matches zero or more characters and an underscore (_) matches any single character. If only a percent sign is entered into a field, all names will match.

The percent sign (%) and underscore (_) are the normal DB2 wild card characters. MAX DB2/UTIL will automatically translate the asterisk (*) to a percent sign and a question mark (?) to an underscore to aid anyone preferring to use those characters.

```
MAX DB2/UTIL ------------- TABLESPACE SELECTION LIST -------------- Row 1 of 5
COMMAND ===> _____________________________________________ SCROLL ===> PAGE
Use GO to continue, ALL or NONE to select/deselect all tablespaces.
Line commands: S and / - RUNSTATS

  Database    Tablespace
S DSN8710     XEMP_PHO
S DSN8710     XEMPPROJ
S DSN8710     XEMPPROJ
S DSN8710     XEMP1
S DSN8710     XEMP2
******************************** Bottom of data ********************************
```

Figure 16: Tablespace Selection List

If the tablespace or index is selected directly by name and only one matching name is found, the utility JCL for that object will be generated immediately. If multiple matching tablespaces or indexes are found or if the tablespace or index was found indirectly by table name, a list of all matching objects will be displayed, allowing you to deselect any tablespaces or indexes that you do not want to process. Once you have selected the objects to be processed from the list, use the **GO** command to generate the JCL.

```
EDIT       MX10006.SPFTEMP1.CNTL                          Columns 00001 00072
Command ===>                                                Scroll ===> PAGE
****** ***************************** Top of Data ******************************
000001 //MX10006  JOB  (MAX),'ME',CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID
000002 //*
000003 //* DOC: THIS JOB WAS GENERATED BY MAX DB2/UTIL
000004 //*
000005 //STEP1 EXEC PROC=DSNUPROC,SYSTEM=DSN1
000006 //DSNUPROC.SYSPRINT DD SYSOUT=*
000007 //SYSIN DD  *
000008  RUNSTATS TABLESPACE DSN8710.XEMP_PHOTO_RESUME
000009  RUNSTATS TABLESPACE DSN8710.XEMPPROJACT1
000010  RUNSTATS TABLESPACE DSN8710.XEMPPROJACT2
000011  RUNSTATS TABLESPACE DSN8710.XEMP1
000012  RUNSTATS TABLESPACE DSN8710.XEMP2
000013 //
****** **************************** Bottom of Data ****************************
```

Figure 17: Generated RUNSTATS JCL

The DB2 utility JCL is generated into a temporary data set and an ISPF edit session will be started. You can edit the JCL as needed, save it to another data set using the **CREATE** command and submit the JCL for execution using the **SUBMIT** command. The **JOB** statement is copied from your ISPF list/log defaults.

4. Unload Data

MAX DB2/UTIL can unload data from DB2 tables and views in a variety of formats. To unload data, enter the owner ID and table name on the main menu and select the Unload function. If wild card characters are used for the owner ID or table/view name and more than one table or view matches the pattern, a list of matching tables and views will be displayed. You can then select the name from the Table Selection List using '**U**' or '/' to unload, '**L**' to load, '**B**' to browse or '**E**' to edit.

Once a table or view is selected for unloading, a list of columns is displayed, allowing you to select the columns and rows to be unloaded.

```
MAX DB2/UTIL ------- CHOOSE UNLOAD DESTINATION AND FORMAT ------- MAX DB2/UTIL
COMMAND ===> ____________________________________________________________________

Unload destination. . . 1   1. Sequential or partitioned data set
                            2. UNIX file
Data Set Name. . 'MX10006.MY.CUT'_____________________________________________
or UNIX file . . ____________________________________________________________
                ____________________________________________________________
                ____________________________________________________________
                ____________________________________________________________

Unload options:
  Replace like-named member or file . YES        ('/': Yes, blank: No)
  Maximum rows. . . . . . . . . . . . 6________  (0: no limit)
  Skip first rows . . . . . . . . . . 2________  (0-999999999)
  then unload every row . . . . . . . 7________  (1: unload all rows)
  Unload format . . . . . . . . . . . 1   0. CUT/PASTE    1. XML
                                          2. Comma sep.   3. Tab delimited
                                          4. Other. . . . ________
  Output code page override . . . . . _____      (1-65534, use ? for list,
                                                  ignored for CUT/PASTE)
Press ENTER to perform the unload request.
Enter END command to cancel the unload request.
```

Figure 18: Choose Unload Destination and Format panel

This panel is displayed after you have selected the columns and rows to be unloaded from a table or view. You must then select either a data set or UNIX file as the destination for the unloaded data. If you select a data set, you must enter the name of an existing sequential or partitioned (PDS or PDSE) data set. For a partitioned data set, you must also provide a member name. If you select a UNIX file as the destination, you must enter the path name of the file. For either a partitioned data set member or a UNIX file, you can specify if an existing member or file is to be replaced.

Once you have selected the destination, you can set the unload options.

Unload options:

Maximum rows: This field allows you to limit the total number of rows unloaded from the table or view. The value '0' specifies that the number of rows is unlimited. A value from 1 to 999999999 limits the number of rows to that value.

Skip first rows: This field causes MAX DB2/UTIL to skip the specified number of rows in the result set before unloading. The first row unloaded will the value specified here plus 1.

then unload every row: This field causes MAX DB2/UTIL to unload every Nth row. A value of '1' causes all rows to be unloaded. A value of '2' causes every second row to be unloaded, and so on.

Unload format: These fields allow you to specify the format of the unloaded data. Format codes 0 through 3 are for formats predefined by MAX Software. Format code 4 allows you to specify the name of a customized format defined in the MAXDFLTS module.

Format code '0' specifies that the data will be unloaded in MAX DB2/UTIL **CUT** and **PASTE** format. The resulting file can be loaded back into a table using either the **LOAD** function or the **PASTE** editor command.

Format codes '1', '2' and '3' specify that the data will be unloaded as text lines in eXtended Markup Language (XML), Comma Separated Variable (CSV) or Tab Delimited (TAB) format. These formats are useful for transferring DB2 data to another product or platform. Because the text lines can potentially be very long, the best choice for a destination for these formats is a UNIX file since data sets have limits for record sizes.

Output code page override: This field allows you to override the default code page for formatted output. It is ignored if the unload format is set to '0' (**CUT**/**PASTE**). For any other format, the number of any code page supported by MAX DB2/UTIL may be specified. Enter a '?' into this field and press ENTER to view a list of the supported code pages and to select one from the list. The default code page for the predefined formats is '37'.

```
EDIT       MX10006.SPFTEMP1.CNTL                          Columns 00001 00072
Command ===>                                                Scroll ===> PAGE
****** ***************************** Top of Data ******************************
000001 //MX10006  JOB  (MAX),'ME',CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID
000002 //*
000003 //* DOC: THIS JOB WAS GENERATED BY MAX DB2/UTIL
000004 //*
000005 //MAXSQL EXEC PGM=MAXSQL
000006 //STEPLIB  DD DSN=MXS.MXRXV320.LOADLIB,DISP=SHR
000007 //SYSTSPRT DD SYSOUT=*
000008 //SYSTSIN  DD *
000009 CONNECT DSN1;
000010 IF LASTCC<>0 THEN EXIT;
000011 SELECT "EMPNO", "FIRSTNAME", "MIDINIT", "LASTNAME", "GENDER", "JOB",
000012 "HIREDATE", "SALARY", "RESUME", "BMP_PHOTO", "EMP_ROWID"
000013 INTO DSN('MX10006.MY.CUT')
000014 FORMAT(XML)
000015 LIMIT(6)
000016 SKIP(2)
000017 EVERY(7)
000018 FROM "MX10006"."EMP_PHOTO_RESUME" ORDER BY "EMPNO";
000019 //
****** **************************** Bottom of Data ****************************
```

Figure 19: Generated Unload JCL

The unload utility JCL is generated into a temporary data set and an ISPF edit session will be started. You can edit the JCL as needed, save it to another data set using the **CREATE** command and submit the JCL for execution using the **SUBMIT** command. The JOB statement is copied from your ISPF list/log defaults.

5. Load Table

MAX DB2/UTIL can load DB2 tables. To load a table, enter the owner ID and table name on the main menu and select the Load function. If wild card characters are used for the owner and table name and more than one table or view matches the pattern, a list of matching tables and views will be displayed. You can then select a name from the Table Selection List using '**L**' or '/' to load, '**U**' to unload, '**B**' to browse or '**E**' to edit.

If you attempt to load a view, a list of the tables upon which the view is will be displayed, allowing you to load one of those tables.

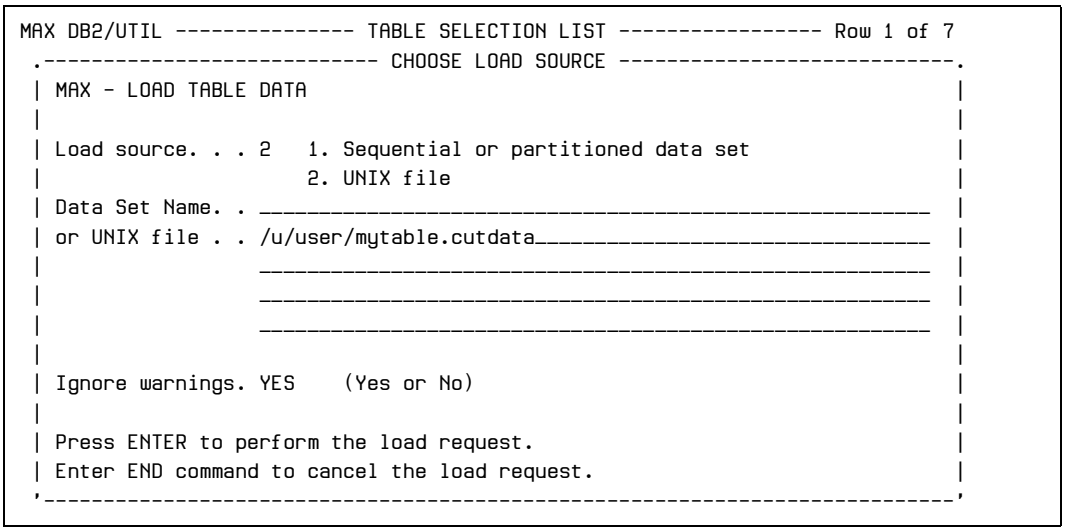

Figure 20: Choose Load Source panel

This panel is displayed after you have selected a table to be loaded. You must select either a data set or UNIX file as the source of the data to be loaded. If you select a data set name, you must enter the name of an existing sequential data set or an existing partitioned (PDS or PDSE) data set and member. If you select a UNIX file as the source, you must enter the path name of an existing file.

You can also indicate if SQL warnings generated during the load are to be ignored or not. If not ignored, any SQL warning or error will cause the load process to stop.

```
EDIT       MX10006.SPFTEMP1.CNTL                          Columns 00001 00072
Command ===>                                                Scroll ===> PAGE
****** ***************************** Top of Data ******************************
000001 //MX10006  JOB  (MAX),'ME',CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID
000002 //*
000003 //* DOC: THIS JOB WAS GENERATED BY MAX DB2/UTIL
000004 //*
000005 //MAXSQL EXEC PGM=MAXSQL
000006 //STEPLIB  DD DSN=MXS.MXRXV320.LOADLIB,DISP=SHR
000007 //SYSTSPRT DD SYSOUT=*
000008 //SYSTSIN  DD *
000009 CONNECT DSN1;
000010 IF LASTCC<>0 THEN EXIT;
000011 INSERT INTO "MX10006"."EMP_PHOTO_RESUME"
000012 FROM FILE('/u/user/mytable.cutdata') BYPASS(WARN);
000013 //
****** **************************** Bottom of Data ****************************
```

Figure 21: Generated Load JCL

The load utility JCL is generated into a temporary data set and an ISPF edit session will be started. You can edit the JCL as needed, save it to another data set using the **CREATE** command and submit the JCL for execution using the **SUBMIT** command. The JOB statement is copied from your ISPF list/log defaults.

6. Relational Unload

In addition to the ability to unload data from a single DB2 table or view (Option 4), MAX DB2/UTIL can unload and reload related data from multiple tables. This feature can be used to build a set of test tables from a subset of production data.

The Relational Unload feature of MAX DB2/UTIL uses information found in the DB2 catalog tables to determine which tables are related via FOREIGN KEY definitions. You can also define "application relationships", which are inter-table relationships that are not known to DB2.

In order to define and use application relationships, you must specify the name of the VSAM file on the Profile panel (Option 0) which will contain these relationships. This file can be shared among multiple users, allowing many people to take advantage of the application relationships once they are defined. If the VSAM file is not specified on the Profile panel, the Relational Unload feature is limited to unloading data based only upon FOREIGN KEY relationships.

The Relational Unload feature provides three options:

6.1 Browse Relationships

This option allows you to view both FOREIGN KEY and application relationships. This can be useful information anytime you are working with DB2 tables.

To view the list of tables that are related to a specific table, first enter the owner ID and table name of a table on the main menu and enter option '**6.1**'. If wild card characters are used for the owner and table name and more than one table matches the pattern, a list of matching table or view will be displayed. You can then select the name from the Table Selection List panel using option '**S**'. If you select a table using option '**6**', a pop-up panel will be displayed, giving you a chance to choose from any of the Relational Unload functions. The Relational Unload functions may not be used with DB2 views.

```
MAX DB2/UTIL ----------- RELATED TABLE SELECTION LIST -------- Row 1 to 3 of 3
COMMAND ===> _____________________________________________ SCROLL ===> PAGE

The following tables are related to MX10006.ITEM
You may select these tables or use END to return to the previous screen.
Use the PROFILE command to change the Relationship File from
'MXS.SQL.RELATE'

/ - Select
  Type Owner    Table Name
_ Appl MX10006  ITEM2
_ 1:M  MX10006  ORDER
_ M:1  MX10006  VENDOR
******************************** Bottom of data ********************************
```

Figure 22: Related Table Selection List panel (Browse)

This panel is displayed after you have selected a table for which the relationships are to be displayed. The `Type` column indicates the type of relationship between the tables. The type is shown as '`Appl`' if the relationship is an application relationship defined in the VSAM file specified on the Profile panel. The type is shown as '`1:M`' for tables that reference the selected table via FOREIGN KEY definitions or as '`M:1`' for tables that are referenced by the selected by in FOREIGN KEY definitions. You can select any of the listed tables using '`/`' to view details about the relationship between the tables.

```
MAX DB2/UTIL -------------- FOREIGN KEY COLUMN LIST ---------- Row 1 to 1 of 1
COMMAND ===> _____________________________________________ SCROLL ===> PAGE

MX10006.ORDER                          | MX10006.ITEM
Seq Foreign Key Column           Type  | Seq Primary Key Column           Type
1   ORD_ITEM                       IN  | 1   ITM_NUMBER                    IN
******************************** Bottom of data ********************************
```

Figure 23: Foreign Key Column List panel

This panel is displayed after you have selected a FOREIGN KEY (1:M or M:1) relationship from the Related Table Selection list. It shows the columns involved in the relationship from both tables.

```
MAX DB2/UTIL ---------------- RELATED COLUMN LIST --------- Row 14 to 22 of 22
COMMAND ===> _____________________________________________ SCROLL ===> PAGE

MX10006.ITEM                          | MX10006.ITEM2
Cardinality . . . 1 (1 or M)          | Cardinality . . . 1 (1 or M)
Seq Column Name                  Type | Seq Column Name                  Type
    ITM_NET                        PD | 14  ITM_NET                        PD
    ITM_NEW_LIST                   PD | 15  ITM_NEW_LIST                   PD
    ITM_NEW_NET                    PD | 16  ITM_NEW_NET                    PD
17  ITM_NUMBER                     IN | 17  ITM_NUMBER                     IN
    ITM_PALLET                     PD | 18  ITM_PALLET                     PD
    ITM_PALLET_INFO                PD | 19  ITM_PALLET_INFO                PD
    ITM_PALLET_QUANT               PD | 20  ITM_PALLET_QUANT               PD
    ITM_RANDOM                     CH | 21  ITM_RANDOM                     CH
    ITM_RCD_CODE                   CH | 22  ITM_RCD_CODE                   CH
******************************** Bottom of data ********************************
```

Figure 24: Related Column List panel (Browse)

This panel is displayed after you have selected an application relationship (Appl) from the Related Table Selection list. It shows the columns involved in the relationship from both tables as well as the cardinality of the relationship. All columns of each table are displayed and the columns that are related are indicated by a number on the left side of panel that is the number of the related column on the right side.

6.2 Edit Relationships

This option allows you to view both FOREIGN KEY and application relationships and the define, delete or alter application relationships. This option is only available if the application relationship file is specified on the Profile panel. Using this option causes the application relationship file to be opened for update. Only one person at a time can have the file opened for update.

To view the list of tables that are related to a specific table, first enter the owner ID and table name of a table on the main menu and enter option '**6.2**'. If wild card characters are used for the owner and table name and more than one table matches the pattern, a list of matching table or view will be displayed. You can then select the name from the Table Selection List panel using option '**S**'. If you select a table using option '**6**', a pop-up panel will be displayed, giving you a chance to choose from any of the Relational Unload functions. The Relational Unload functions may not be used with DB2 views.

```
MAX DB2/UTIL ----------- RELATED TABLE SELECTION LIST -------- Row 1 to 3 of 3
COMMAND ===> _____________________________________________ SCROLL ===> PAGE

The following tables are related to MX10006.ITEM
You may select these tables or use END to return to the previous screen.
Select the NEW line to add a relationship.  Use the PROFILE command to
change the Relationship File from 'MXS.SQL.RELATE'

/ - Select    D - Delete
  Type Owner    Table Name
_ NEW  MX10006_ %_____________________________________________________________
_ Appl MX10006  ITEM2
_ 1:M  MX10006  ORDER
_ M:1  MX10006  VENDOR
****************************** Bottom of data ********************************
```

Figure 25: Related Table Selection List panel (Edit)

This panel is displayed after you have selected a table for which the relationships are to be edited. The Type column indicates the type of relationship between the tables. The type is shown as '`Appl`' if the relationship is an existing application relationship defined in the VSAM file specified on the Profile panel. The Type is shown as '`1:M`' for tables that reference the selected table via FOREIGN KEY definitions or as '`M:1`' for tables that are referenced by the selected by in FOREIGN KEY definitions. The first line under the column headings has a type of '`NEW`' and is used to create new application relationships.

You can select any of the listed tables or the `NEW` line by entering a '`/`' on the line. If you select a FOREIGN KEY relationship (1:M or M:1), you will be shown the columns involved in the relationship, but you will not be allowed to change the relationship. If you select an existing application relationship, you will be allowed to change the definition. You may also delete an existing application relationship by entering a '**D**' on the line. To create a new application relationship, select the `NEW` line and enter the name of the table to which you want create a relationship.

```
MAX DB2/UTIL --------------- TABLE SELECTION LIST ------------ Row 1 to 7 of 7
COMMAND ===> _____________________________________________ SCROLL ===> PAGE

Select one of the tables below using S or / to define a relationship to
table MX10006.ITEM

  Owner    Table Name         Type  Label
_ MX10006  EMP_FORMER         Table MAX DB2/UTIL sample table: deleted rows
_ MX10006  EMP_PHOTO_RESUME   Table MAX DB2/UTIL sample table with LOB columns
_ MX10006  ITEM2              Table
_ MX10006  ORDER              Table
_ MX10006  ORDER_WEEKS        Table
_ MX10006  VENDOR             Table
_ MX10006  VENDOR_OTHER       Table
******************************* Bottom of data ********************************
```

Figure 26: New Relationship Table Selection List panel

This panel is displayed when the NEW line is selected from the Related Table Selection List and a pattern is entered for the owner or name of the related table. All tables which match the pattern are displayed, allowing you to select the one you want for the new application relationship.

```
MAX DB2/UTIL ---------------- RELATED COLUMN LIST --------- Row 24 to 35 of 35
COMMAND ===> _____________________________________________ SCROLL ===> PAGE

Use END to save changes.  Use CANCEL to exit without saving.
Use DELETE to delete existing relationship.

MX10006.VENDOR                        | MX10006.ITEM
Cardinality . . . 1 (1 or M)          | Cardinality . . . M (1 or M)
Seq Column Name                  Type | Seq Column Name                   Type
___ VND_REVIEW                     CH | 24  ITM_ROTATION                   CH
___ VND_SUPPLIER_ADDR1             VC | 25  ITM_SIZE_COUNT                 VC
___ VND_SUPPLIER_CITY              VC | 26  ITM_SLOT_NUMB                  CH
___ VND_SUPPLIER_FAX               VC | 27  ITM_SLOT_TYPE                  CH
___ VND_SUPPLIER_NAME              VC | 28  ITM_STACK                      PD
___ VND_SUPPLIER_PHONE             VC | 29  ITM_TERMINATE_DATE             DT
___ VND_SUPPLIER_STATE             VC | 30  ITM_VEND                       IN
___ VND_SUPPLIER_STRET             VC | 31  ITM_WHSE_ZONE                  PD
___ VND_SUPPLIER_ZIP               VC |
30_ VND_VEND                       IN |
___ VND_WHSE_CODE                  CH |
___ VND_WHSE_ID                    CH |
******************************** Bottom of data ********************************
```

Figure 27: Related Column List panel (Edit)

This panel is displayed after you have selected either an existing application relationship (Appl) from the Related Table Selection list or the NEW line and have selected a table. It shows all of the columns from each of the table which can be related and allows you to indicate which columns are related. The panel also allows you to define the cardinality of the relationship between the tables.

In the example above, the VND_VEND column of the VENDOR table is related to the ITM_VEND column of the ITEM table. This is indicated by the number of the ITM_VEND column (30) entered next to the VND_VEND column. Multiple columns may be related, although each column may only be related to one other column.

Once all of your changes have been made to the relationship definition, you can save the updates by pressing the PF3 (END) key. To discard any changes, enter the **CANCEL** command. To delete the relationship definition, enter the **DELETE** command.

6.3 Unload Related Data

This option allows you to generate the JCL and control statements required to unload and reload related data from DB2 tables. Both FOREIGN KEY and application relationships are considered when determining which rows from which tables are to be unloaded.

The relational unload starts with a table to which selection criteria will be applied to determine which rows will be unloaded. Rows in other tables that are related to the selected rows will also be unloaded.

To start the relational unload process, first enter the owner ID and table name of a table on the main menu and enter option '**6.3**'. If wild card characters are used for the owner and table name and more than one table matches the pattern, a list of matching table or view will be displayed. You can then select the name from the Table Selection List using option '**S**'. If you select a table using option '**6**', a pop-up panel will be displayed, giving you a chance to choose from any of the Relational Unload functions. The Relational Unload functions may not be used with DB2 views.

```
MAX DB2/UTIL -------- RELATED UNLOAD TABLE SELECTION LIST ---- Row 1 to 5 of 5
COMMAND ===> _____________________________________________ SCROLL ===> PAGE

Use GO to unload.  Other commands are: RESET, ALL, NONE and CANCEL

The following tables are related to MX10006.ITEM
This list includes tables related via application relationships as defined
in the Relationship File 'MXS.SQL.RELATE'

Select tables to be unloaded using '/'.
  Owner    Table Name
/ MX10006  ITEM2
/ MX10006  ORDER
/ MX10006  ORDER_WEEKS
/ MX10006  VENDOR
/ MX10006  VENDOR_OTHER
****************************** Bottom of data ******************************
```

Figure 28: Related Unload Table Selection List panel

This panel is displayed after you have selected a table to be used to start the relational unload process. All tables that are related to the initial table are displayed, as well as any tables that are related to those tables, and so on. Initially all of these tables will be listed on this panel and selected to be included in the relational unload. You can deselect any of the tables by blanking out the '/' character next to the table name. If you deselect a table, any tables that are related only to that table will be removed from the list. Once you are satisfied with the list of tables, enter the **GO** command to continue.

The **RESET** and **ALL** commands restore the list of tables to the way they appeared when the panel was first displayed with all tables selected. The **NONE** command deselects all tables in the list, making it easier to select only a few tables. At least one related table must be selected before the **GO** command can be used.

```
MAX DB2/UTIL --------------- COLUMN SELECTION LIST --------- Row 1 to 12 of 31
COMMAND ===> _____________________________________________ SCROLL ===> PAGE
Use GO to unload table MX10006.ITEM
Other commands are: SQL, UNLOAD, RESET, ALL, NONE, SAVE, COPY, END and CANCEL.

                      Default WHERE connector. . . . . AND  (Or or And)
Col Row    Data
Seq Ord A/D Fmt Column Name                    Type
1__ ___  A      ITM_RCD_CODE                  CHAR(2) NOT NULL
                Where: ______________________________________________________
2__ 1__  A      ITM_NUMBER                    INTEGER NOT NULL
                Where: <100__________________________________________________
3__ ___  A      ITM_VEND                      INTEGER NOT NULL
                Where: ______________________________________________________
4__ ___  A      ITM_DESC                      VARCHAR(56) NOT NULL
                Where: ______________________________________________________
5__ ___  A      ITM_SIZE_COUNT                VARCHAR(35) NOT NULL
                Where: ______________________________________________________
6__ ___  A      ITM_WHSE_ZONE                 DECIMAL(1) NOT NULL
                Where: ______________________________________________________
7__ ___  A      ITM_SLOT_NUMB                 CHAR(6) NOT NULL
                Where: ______________________________________________________
8__ ___  A      ITM_SLOT_TYPE                 CHAR(1) NOT NULL
                Where: ______________________________________________________
9__ ___  A      ITM_PALLET_INFO               DECIMAL(5) NOT NULL
                Where: ______________________________________________________
10_ ___  A      ITM_PALLET_QUANT              DECIMAL(5) NOT NULL
                Where: ______________________________________________________
11_ ___  A      ITM_CASE_LENGTH               DECIMAL(5,1) NOT NULL
                Where: ______________________________________________________
12_ ___  A      ITM_CASE_WIDTH                DECIMAL(5,1) NOT NULL
                Where: ______________________________________________________
```

Figure 29: Related Unload Column Selection List panel

Once the **GO** command is entered, the columns of the initially selected table will be displayed in the same Column Selection List as is displayed for Browse, Edit and Unload. For a Relational Unload, only the `Where:` information is used. All other fields are ignored. However, selection criteria that was previously saved for a Browse, Edit or Unload can be loaded and used for a Relational Unload.

In the example above, the Relational Unload will be based on all rows in the `ITEM` table that have an `ITM_NUMBER` that is less than `100`. These rows and rows in other tables that are related to these rows will be unloaded.

```
MAX DB2/UTIL ------------- RELATIONAL UNLOAD OPTIONS -------------------------
COMMAND ===> ____________________________________________________________________

Unload destination. . . . . . . . 1   0. DB2 tables (direct insert)
                                      1. Unload/reload via sequential data set
                                      2. Unload/reload via UNIX file
Target tables:
  New DB2 subsystem . . . . . . . ____      (Blank for direct insert or CREATE)
  New location ID . . . . . . . . ________  (Blank for CREATE)
  New owner ID. . . . . . . . . . MX11006_
  Disposition . . . . . . . . . . 3  1. Append rows  2. Delete existing rows
                                     3. CREATE target table LIKE unload tables

Unload/reload data set or file name pattern with &SSI, &OWN, &TBL and &SEQ:
'MX10006.&OWN.&TBL.&SSI&SEQ'_____________________________________________________

Data set allocation unit name . . SYSALLDA  (Example: SYSALLDA)
  Primary number megabytes. . . . 50___     (1-99999)
  Secondary number megabytes. . . 10___     (0-99999)

Press ENTER to perform the unload request.
Use END to cancel the unload request.
```

Figure 30: Relational Unload Options panel

This panel is displayed once the **GO** command is entered on the Column Selection List. There are three options for where the data is to be unloaded. Once you have entered the required information into all fields of this panel, press ENTER to generate the JCL and control statements for the relational unload.

Unload destination: The data can be unloaded directly into the target DB2 tables or it can be unloaded into either MVS sequential data sets or UNIX files.

Target tables:

New DB2 subsystem: If the data is to be unloaded into MVS data sets or UNIX files, the target tables can be in a different DB2 subsystem than the source tables. Enter the name of the DB2 subsystem of the target tables if it is different from the source tables.

New location ID: If the target tables have a non-blank location ID, enter the target location.

New owner ID: If the target tables have a different owner ID than the source tables, enter the owner ID of the target tables. Since the source and target tables must either have different names or must be in different DB2 subsystems or locations, at least one of these fields must be non-blank.

Disposition: This field determines what happens to the rows or target tables when they are loaded. Option '1' will cause the unloaded rows to be added to any rows that are already in the target tables. Option '2' causes all existing rows in the target tables to be deleted before the unloaded rows are added. Option '3' causes the target table to be created with the same structure as the source tables. If option '3' is used and the target tables already exist, they will be deleted and recreated causing any existing rows to be deleted as well.

Unload/reload data set or file name pattern with &SSI, &OWN, &TBL and &SEQ: This field is used to specify the names of the MVS data sets or UNIX files into which the data will be unloaded. The rows from each source table will be unloaded into a separate data set or file. The pattern entered into this field is used to generate the name of each data set or file, each of which must be unique. The pattern must contain one or more of the following variables:

&SSI	Replaced by the 1 to 4 character DB2 subsystem ID of the source tables.
&OWN	Replaced by the 1 to 8 character owner ID of each source table.
&TBL	Replaced by the name of each source table. When used in a pattern for an MVS data set, the name will be truncated to 8 characters if needed.
&SEQ	Replaced by a 4-digit sequence number. Since **&TBL** is shortened to 8 characters in an MVS data set name, the generated names may not be unique. Including **&SEQ** in the pattern ensures that the data set names will be unique.

In a UNIX file name pattern, the variable names may be specified in either uppercase or lowercase. If the variable name is entered in lowercase, the substituted value will be converted to lowercase letters.

Data set allocation unit name: These fields are only used when the data is to be unloaded to MVS data sets. This information is used to generate the allocations for the data sets.

Primary number megabytes:

Secondary number megabytes:

The unload/reload utility JCL is generated into a temporary data set and an ISPF edit session will be started. You can edit the JCL as needed, save it to another data set using the **CREATE** command and submit the JCL for execution using the **SUBMIT** command. The JOB statement is copied from your ISPF list/log defaults.

CHAPTER 3: SQL COMMAND UTILITY

Introduction

The MAX DB2/UTIL product includes a command driven utility called MAXSQL. MAXSQL may be used in a z/OS or OS/390 batch job, or invoked in TSO to execute SQL commands. In addition to SQL commands, MAXSQL can execute commands to test and react to the success or failure of SQL commands.

General Concepts

Command Syntax

```
command    [command_operands...]
```

Parameters

command	Specifies a valid command. (Refer to the section "*Commands*" on page 74 for a listing of these valid commands.) A command is specified before any command operands and may begin in any column.
command_operands	Command operands immediately follow the command. For SQL commands, the syntax is that of DB2 SQL commands. For other MAXSQL commands, the syntax of each operand is generally a keyword, which may be followed by a value enclosed in parentheses. See each individual command for further details. All lowercase letters (a-z) that are not enclosed within quotes or apostrophes are converted to uppercase before being processed.

Command Continuation

Commands continue from one line to the next until ended with a terminating semi-colon (`;`). The left-most character of a continuation line immediately follows the right-most character of the preceding line. Unless the **SET MARGINS** command is used to limit the columns, the entire input line is processed by MAXSQL.

Comments

Comments are allowed wherever a blank is allowed, both between commands and within commands. Comments start with a '/*' pair of characters and end with a '*/' pair of characters. Comments may not be nested. MAXSQL logically processes each comment as a single blank character.

Note: In OS/390 batch, do not start a comment in column 1 of a line since the '/*' characters will usually be interpreted as the end-of-file of a SYSIN stream.

Input and Output

In batch, MAXSQL reads control statement lines from the DDNAME SYSTSIN and writes messages to SYSTSPRT. Under on-line TSO, MAXSQL writes messages to the terminal and reads control statement lines from the data stack until the data stack is empty and then reads from the terminal. Other files may be referenced by individual commands.

At the end of the input lines, a final semi-colon (;) is added if needed to terminate the last command, then an implicit **STOP MAXCC** command is executed. When MAXSQL is reading input lines from the terminal, you must enter a **CANCEL**, **EXIT**, **QUIT** or **STOP** command to terminate processing.

Command Processing

After each command is processed, the results of the command are summarized in a value set in a variable called '**LASTCC**'. Another variable called '**MAXCC**' is initially set to zero and updated to the **LASTCC** value whenever the **LASTCC** value is greater than **MAXCC**. SQL commands set two more variables: '**SQLCODE**' and '**SQLMSG**'. All of these variables can be displayed by the **SAY** command. The values of **LASTCC**, **MAXCC** and **SQLCODE** can be tested by **IF** commands. The values of **LASTCC** and **MAXCC** can be explicitly changed by **SET** commands.

In general, **LASTCC** will be set to one of the following values, depending on the results of the command:

Value	Reason
0	Successful completion
4	Warning
8	Error
12	Severe error
16	Syntax error

Sample JCL

The following JCL invokes MAXSQL to perform a **GRANT** command:

```
//MAX JOB
//SQL EXEC PGM=MAXSQL
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
 CONNECT DSN1;
 IF LASTCC = 0 THEN
    GRANT EXECUTE ON RXSQL /* example GRANT command */
          TO PUBLIC;
/*
```

Use of the PARM field

MAXSQL processes the **PARM** value as if it were a single command input line. The **PARM** value is not subject to the **SET MARGINS** command, although the **SET MARGINS** command may be included in the **PARM** value. Multiple commands may be included in the **PARM** value if they are separated by semi-colons (`;`). If a final semi-colon is not specified in the **PARM** value, MAXSQL automatically appends a semi-colon to the end of the value.

Commands

Summary

Command	Description
ALLOCATE, FREE	Allocate or free a data set.
CANCEL, EXIT, QUIT, STOP	Terminate MAXSQL processing.
DO...END	Group a series of commands, usually within a **THEN** or **ELSE** clause of the **IF** command.
[EXEC SQL] { ALTER | COMMENT | COMMIT | CONNECT | CREATE | DELETE | DISCONNECT | DROP | GRANT | INSERT | LABEL | LOCK | RENAME | REVOKE | ROLLBACK | SELECT | SET CURRENT | UPDATE } ...	Execute a DB2 SQL statement.
IF...THEN...ELSE...	Test the result of a previous command.
SAY	Display a message.
SET	Set the **LASTCC**, **MAXCC** or **MARGINS**.
SHELL	Issue the **ALLOC**, **FREE CONCAT** or **OUTDES** commands via BPXWDYN or any UNIX Systems Services shell command.
TSO	Issue a **TSO** command.

ALLOCATE

The **ALLOCATE** command is available for dynamic allocation of data sets.

Command Syntax

```
ALLOCATE
        FILE(filename)
                [ DATASET(data set name)                ]
                [ NEW|OLD|MOD|SHR|                      ]
                [ DELETE|KEEP|CATLG|UNCATLG             ]
                [ SYSOUT( x )                           ]
                [ SPACE(nnnnnn|CYL|TRK)                 ]
                [ PRIMARY(nnnnn)                        ]
                [ SECND(nnnnn)                          ]
                [ DIRBLK(nnnnn)                         ]
                [ UNITNAME(unitname)                    ]
                [ DSORG(dsorg)                          ]
                [ RECFM(recfm)                          ]
                [ BLKSIZE(nnnnn)                        ]
                [ LRECL(nnnnn)                          ]
```

Parameters

filename	Specify a valid filename (one to eight characters) to be used to reference the data set to be allocated.
data set name	Specify the 1 to 44 character data set name. This field is optional. If this field is not specified, the file is assumed to be a SYSOUT data set.
NEW|OLD|MOD|SHR	Use one of these four options to specify the original disposition of the data set.
DELETE|KEEP|CATLG|UNCATG	Use one of these four options to specify the final disposition of the data set. This need only be specified to change the disposition of the data set from its input disposition.
X	Specify SYSOUT class for a SYSOUT data set.
nnnnn|CYL|TRK	Specify the type of allocation. If the numeric option is chosen, this is considered a block allocation. The number must be specified as a full 5 digits with the necessary leading zeroes.
nnnnn	This parameter is specified for several operands. Any time a numeric value is specified in this format, it must be a full 5 digits with the necessary leading zeroes. This parameter is used to specify primary and secondary allocation, number of directory blocks, block size, and logical record length.

unitname	Specify the unit name for allocation.
dsorg	Specify the data set organization. Valid values are: PO, POU, DA, DAU, PS, PSU.
recfm	Specify the record format of the data set to be allocated. Record formats supported are: Fixed, Variable, Undefined, Blocked, Spanned, Ascii Control, Machine Control, and Track Overflow.

CANCEL

The **CANCEL** command may be used to terminate MAXSQL processing.

Command Syntax

CANCEL [[(] code [)]]

Parameters

code	Specify **LASTCC**, **MAXCC** or a decimal value from '0' to '4095'. If not specified, the code defaults to **MAXCC**.

DO

The **DO** command may be used to define a group of commands to be executed, usually as the **THEN** or **ELSE** clause of an **IF** command. The group is terminated by an **END** command. Neither the **DO** or **END** commands require a terminating semi-colon nor set **LASTCC** or **MAXCC**, although the commands within the group generally will. **DO...END** groups may be nested.

Command Syntax

DO [commands...] END

Parameters

commands	Specify any valid MAXSQL commands, each terminated by a semi-colon (;).

EXEC SQL

The **EXEC SQL** command may be executed as an SQL command.

Command Syntax

[EXEC SQL] sql-command

Parameters

sql-command	Specify any of the following SQL commands: **ALTER**, **COMMENT**, **COMMIT**, **CONNECT**, **CREATE**, **DELETE**, **DISCONNECT**, **DROP**, **GRANT**, **INSERT**, **LABEL**, **LOCK**, **RENAME**, **REVOKE**, **ROLLBACK**, **SELECT**, **SET** or **UPDATE**. The format of the commands is identical to the DB2 SQL statement as documented in the IBM DB2 SQL Reference except for the following: **CONNECT**, **INSERT**, **SELECT** and **SET**. The syntax of these statements is described below.

Parameter Description

CONNECT

The **EXEC SQL CONNECT** command is used to connect MAXSQL to a DB2 subsystem. It must be successfully executed before any other **EXEC SQL** commands can be processed.

Parameter Syntax:

[EXEC SQL] CONNECT subsystem [PLAN plan]

Sub-Parameters:

subsystem	Specify the 1 to 4 character DB2 subsystem name.
plan	Specify the 1 to 8 character plan name to override the default plan used by MAXSQL.

INSERT

The **EXEC SQL INSERT** command may be used to insert one or more rows into a DB2 table. MAXSQL recognizes the following extension to the DB2 syntax to allow values to be inserted from a file. The standard **INSERT** command with the **VALUES** or **SELECT** clause may also be used.

Parameter Syntax

[EXEC SQL] INSERT INTO table-name
[(column-names)] [other clauses...]
FROM {FILE({ddname|path})|{DATASET|DSNAME|DSN}(dsn)}
[BYPASS(list of SQLCODE values)]
[MESSAGES({OFF|SAY})]

Sub-Parameters

table-name	Specify the name of the table into which the rows are to be inserted. The table name may be qualified with an owner ID. If the owner ID or table name contain special characters, they must be enclosed in quotes.
column-name	Specify a list of the columns whose values are to be supplied from the file. One column name must be specified for each column saved in the file (except ROWID values). If not specified, the column names saved in the file are used and must be defined as columns in the table.
ddname	Specify the **DDNAME** of an allocated data set containing the values to be inserted.
path	Specify a UNIX Systems Services path name enclosed in apostrophes (') or quotes (") that contains the values to be inserted.
dsn	Specify the name of a data set containing the values to be inserted. If the name is not enclosed in apostrophes or quotes and MAXSQL is running in a TSO environment, the current TSO PREFIX will be prefixed to the data set name. MAXSQL will implicitly allocate and free the data set using the **DDNAME** `MAX00001`.
list of SQLCODE values	Specify one or more **SQLCODE** values separated by commas. If the insert of any row results in an error or warning with a **SQLCODE** that matches one of the codes in the list, the insert process will continue with the next values from the file. If a resulting **SQLCODE** does not match one of the listed values, the insert process terminates. Each **SQLCODE** value must be a positive or negative decimal number or may be the keyword `WARN`, which will match all positive **SQLCODE** values.
OFF|SAY	The value specifies the disposition of the messages resulting from an insert that failed, but whose **SQLCODE** was bypassed. **OFF** specifies that the messages are to be suppressed. **SAY** is the default and specifies that the messages are to be printed.

SELECT

The **EXEC SQL SELECT** command may be used to unload a result set to a sequential data set or UNIX file in a number of formats.

Parameter Syntax

```
[EXEC SQL] SELECT ...
INTO {FILE({ddname|path})|{DATASET|DSNAME|DSN}(dsn)}
[FORMAT(format) [CODEPAGE(codepage)] ]
[LIMIT(limit-count)]
[SKIP(skip-count)]
[EVERY(every-count)]
[IGNORE(100)]
FROM ...
```

Sub-Parameters

ddname	Specify the **DDNAME** of an allocated data set.
path	Specify a UNIX Systems Services path name enclosed in apostrophes (') or quotes (").
dsn	Specify the name of a data set. If the name is not enclosed in apostrophes (') or quotes (") and MAXSQL is running in a TSO environment, the current TSO PREFIX will be prefixed to the data set name. MAXSQL will implicitly allocate and free the data set using the **DDNAME** `MAX00001`.
format	Specifies the name of a Data Transformation template defining the format of the data to be written to the data set or file handle. Predefined formats include XML, CSV, TAB and RAW. Additional formats may be defined in the MAXDFLTS module. If the **FORMAT** clause is omitted, the result set will be written in an internal format suitable for the MAX/REXX **EXECUTE USING FILE** statement or the MAX DB2/UTIL **PASTE** or **LOAD** facilities.
limit-count	Specify a positive decimal value to limit the number of rows to be written to the output file. If not specified, the number of rows will not be limited.
skip-count	Specify a positive decimal value to cause the initial rows returned by the **SELECT** to be skipped and not written to the output file. If not specified, no rows will be initially skipped.

every-count	Specify a positive decimal number to cause only every "**every-count**" rows to be written to the output file. If not specified, the default is '1', which will cause all selected rows to be written after any initially skipped rows.
IGNORE(100)	Specify this parameter to cause the SQLCODE '100' value to be ignored that would normally result when no rows are selected and written to the output file.
codepage	Specifies a code page number from 1 to 65535, which overrides the default output code page for the named **FORMAT**.

SET

The **EXEC SQL SET** command may be used to change the values of DB2 special registers.

Parameter Syntax

[EXEC SQL] SET special-register = value

Sub-Parameters

special-register	Specify the name of a modifiable DB2 special register, such as CURRENT SQLID.
value	Specify the new value for the special register. Strings must be enclosed in apostrophes (').

EXIT, QUIT, STOP

The **EXIT**, **QUIT** or **STOP** command may be used to terminate MAXSQL processing. All perform the same function.

Command Syntax

EXIT [[(] code [)]]
QUIT [[(] code [)]]
STOP [[(] code [)]]

Parameters

code	Specify **LASTCC**, **MAXCC** or a decimal value from 0 to 4095. If not specified, the code defaults to **MAXCC**.

IF

The **IF** command may be used to test the result of a previous command and take some action based on if the test is true or false. The keyword **THEN** marks the end of the test clause and the start of the statement to be executed if the test is true. The optional **ELSE** keyword may be specified after the **THEN** statement to mark the start of the statement to be executed if the statement is false. The **IF** command itself does not alter the **LASTCC** or **MAXCC** values, but the statement executed in the **THEN** or **ELSE** clause may do so.

Command Syntax

IF **test THEN command [ELSE command]**

Parameters

test	Specify a conditional expression, comparing **LASTCC**, **MAXCC** or **SQLCODE** to a decimal value. The comparison operator may be one of: =, <>, >, <, >= or <=. Conditional expressions may be combined using the AND, &, OR or | operators and each expression may be enclosed in parentheses.
command	Specify any valid **MAXSQL** command, including an **IF** command. To specify multiple statements to be executed for the **THEN** or **ELSE** condition, enclose those statements in a **DO...END** group.

SAY

The **SAY** command may be used to print a message. The **SAY** command does not alter the **LASTCC** or **MAXCC** values.

Command Syntax

SAY **[string-expression]**

Parameters

string-expression	Specify one or more strings enclosed in apostrophes (') or quotes (") or the keywords **LASTCC**, **MAXCC**, **SQLCODE** or **SQLMSG**. Each token (string or keyword) must be separated by one or more blanks. These separating blanks will be converted to a single blank character. To concatenate tokens without the blank separator, include the concatenation operator '||' between the tokens. To display a message consisting of multiple lines, include the special value '\n' within the strings to indicate the end of each line.

SET

The **SET** command may be used to set LASTCC, MAXCC, and MARGINS values.

Command Syntax

```
SET     LASTCC [=] [(] value [)]
SET     MAXCC [=] [(] value [)]
SET     MARGINS [=] ( start [, end] )
```

Parameters

value	Specify a decimal number from 0 to 4095.
start	Specify a starting column. Any characters on an input line before this column will be ignored.
end	Specify an ending column. Any characters on an input line after this column will be ignored. If not specified, all characters beginning with the starting column will be processed.

SHELL

The **SHELL** command may be used to execute a UNIX command.

Command Syntax

```
SHELL     [string-expression]
```

Parameters

string-expression	Specify one or more strings enclosed in apostrophes (') or quotes (") or the keywords **LASTCC**, **MAXCC**, **SQLCODE** or **SQLMSG**. Each token (string or keyword) must be separated by one or more blanks. These separating blanks will be converted to a single blank character. To concatenate tokens without the blank separator, include the concatenation operator '||' between the tokens. The string expression must be a valid **ALLOC**, **FREE**, **CONCAT** or **OUTDES** command supported by the BPXWDYN command processor. See the 'IBM z/OS 1.4 (or later) Using REXX and z/OS UNIX Systems Services' manual for the syntax. This form of the **SHELL** command is available for all OS/390 and z/OS releases supported by MAXSQL. On z/OS 1.4 or later, the string expression can be any z/OS UNIX Systems Services shell command. If the string expression is not specified, the **SHELL** command simply sets **LASTCC** to `0` or `8`, depending on if UNIX Systems Services is active or not.

TSO

The **TSO** command may be used to execute a TSO command. MAXSQL must be running under TSO (batch or on-line) in order to use the **TSO** command.

Command Syntax

TSO [string-expression]

Parameters

string-expression	Specify one or more strings enclosed in apostrophes (') or quotes (") or the keywords **LASTCC**, **MAXCC**, **SQLCODE** or **SQLMSG**. Each token (string or keyword) must be separated by one or more blanks. These separating blanks will be converted to a single blank character. To concatenate tokens without the blank separator, include the concatenation operator '||' between the tokens. The string expression must be a valid **TSO** command. If the string expression is not specified, the **TSO** command simply sets **LASTCC** to `0` or `8`, depending on if MAXSQL is running under TSO or not.

CHAPTER 4: DATA PRIVACY FUNCTIONS

GENERAL CONCEPTS

MAX DB2/UTIL includes DB2 scalar functions to translate, scramble or unscramble column values. These functions can be used to provide privacy for DB2 data. The functions can be included in DB2 expressions, especially **SELECT** values. The MAX DB2/UTIL data privacy functions return results that are compatible with the similar functions provided with the other MAX Software products, allowing privatized data to be shared between DB2, IMS and VSAM.

The three basic functions included with MAX DB2/UTIL are **MAX.SCRAMBLE**, **MAX.TRANSLATE** and **MAX.UNSCRAMBLE**. All are defined within the MAX schema when MAX DB2/UTIL is installed for a DB2 subsystem. The installation process also grants access to the functions to PUBLIC (all users).

An important use of the MAX DB2/UTIL data privatization functions is to allow access to DB2 tables containing sensitive data to users who should not need to see the actual data values. For instance, programmers may need access to tables containing address information for testing purposes, but due to HIPAA regulations should not be able to view actual customer addresses. These programmers could be granted access to a view of the tables but not the *actual* tables. These views could scramble the address information.

An example a privatized view definition could be:

```
DEFINE VIEW PRIVATE (NAME, ADDRESS) AS SELECT NAME,
MAX.SCRAMBLE(ADDRESS,'@@ALPHAMERIC',5,CURRENT SQLID)
FROM CUSTOMER
```

Function Descriptions

TRANSLATE

The **TRANSLATE** function translates from one set of characters to another. There are two basic forms of the **TRANSLATE** function. The first form allows the set of source and target characters to be selected by name. The second form allows the sets of source and target characters to be specified as character strings.

The data type of the returned value depends on the data type of the translated value. If the value is a character string (CHAR or VARCHAR), the result will be a VARCHAR or the same length. If the value is a CLOB or BLOB, the result will be of the same type and length. The result can be cast to the desired data type.

Command Syntax

```
MAX.TRANSLATE(value,name)
MAX.TRANSLATE(value,target,select)
```

Parameters

value	Specify a DB2 expression that results in a character value. The data type can be a character string, CLOB or BLOB value.
name	Specify a string with the name of a translation table. The named translation table must be defined in the MAXDFLTS module.
target	Specify a string of 1 to 255 characters. These are the set of characters to which selected characters in the value will be translated.
select	Specify a string of the same number of characters as the target string. These are the set of characters from which characters in the value will be selected for translation.

Example

```
MAX.TRANSLATE(ADDRESS,'9876543210','0123456789')
```

In this example, only the numeric characters in the **ADDRESS** column are translated since only the digits are specified in the selection string. The translations will occur as follows: 0 → 9, 1 → 8, 2 → 7, 3 → 6, 4 → 5, 5 → 4, 6 → 3, 7 → 2, 8 → 1, 9 → 0.

SCRAMBLE
UNSCRAMBLE

The **SCRAMBLE** and **UNSCRAMBLE** functions scramble or unscramble data. For the same inputs, **UNSCRAMBLE** will reverse the operation of the **SCRAMBLE** function. There are two forms of each of these functions. The first form uses a fixed "PIN" value to determine the scrambling algorithm. The second form accepts an additional "key" value that will be used to modify the initial PIN value, thus allowing the scrambling algorithm to be dependent on another value.

The data type of the returned value depends on the data type of the translated value. If the value is a character string (`CHAR` or `VARCHAR`), the result will be a `VARCHAR` or the same length. If the value is a `CLOB` or `BLOB`, the result will be of the same type and length. The result can be cast to the desired data type.

Command Syntax

MAX.SCRAMBLE(value,select,pin)
MAX.SCRAMBLE(value,select,pin,key)
MAX.UNSCRAMBLE(value,select,pin)
MAX.UNSCRAMBLE(value,select,pin,key)

Parameters

value	Specify a DB2 expression that results in a character value. The data type can be a character string, `CLOB` or `BLOB` value.

select	Specify a string of up to 255 characters used to specific the set of characters in the value that will be scrambled or unscrambled. Any characters in the value which do not appear in the selection string will not be changed. The string can be specified as one of the following names, which provide an easy way to select commonly used characters: '@@ALPHA': The set of letters A-Z and a-z. '@@LOWER': The set of lowercase letters a-z. '@@UPPER': The set of uppercase letters A-Z. '@@NUMERIC': The set of digits 0-9. '@@ALPHAMERIC': The set of letters and digits A-Z, a-z and 0-9. If the empty string is specified, the set of all 256 characters will be scrambled or unscrambled.
pin	Specify a non-negative integer value. This value will be used to build the scramble and unscramble tables for the selected characters.
key	Specify an expression that results in a string of up to 255 characters. These characters will be used to modify the PIN value to generate a unique set of scramble and unscramble tables. Some possible expression values are the name of the primary column or a DB2 special register like CURRENT SQLID, which will cause the values to be scrambled depending on primary key value or the user's ID.

Example

```
MAX.SCRAMBLE(ADDRESS,'0123456789',5,CURRENT SQLID)
```

In this example, only the numeric characters in the **ADDRESS** column are scrambled to other digits. The set of digits to which the characters will be scrambled depends on the user's ID. The function could have also been coded as:

```
MAX.SCRAMBLE(ADDRESS,'@@NUMERIC',5,CURRENT SQLID).
```

CHAPTER 5: DATA SET NAME LIST FUNCTIONS

Introduction

The DSNL provides a unique and customized front-end for accessing MAX DB2/UTIL. It can be thought of as a "Project Organizer" that allows a user to organize their table names in the manner which is most natural for them.

The DSNL eliminates the need to continually enter Owner ID, Table/View name, and DB2 subsystem criteria DSN information needed for accessing your tables with MAX DB2/UTIL. A DSNL is maintained for each user or shared between users and saved across TSO sessions.

You can create and maintain multiple DSNLs to allow further grouping of tables by function. With this capability, you can, for example, group all tables used for a given application, or group tables undergoing conversion or quality assurance activities. If appropriate, a given table name may appear in multiple DSNLs.

DSNLs are stored as PDS members in a Project DSN that can be established by assigning the **MAXCDSN** variable name to a standard PDS name in the MAX startup procedure. Refer to the MAX Software Product Installation Guide for more information on this procedure.The default is to use the user's ISPF profile data set. The Project DSN can also be changed when creating a new DSNL with the **CREATE** command or using the **PROFILE** command to list the contents of a Project DSNL.

DSNL is a set of functions that is part of the larger companion product MAX/PDF. MAX/PDF provides for organizing access to all kinds of files (i.e. VSAM, SAM, PDS, IMS, etc.) besides DB2 tables and requires a separate license to use. However, the DSNL extension to store and access DB2 tables is included in your MAX DB2/UTIL license and is described in this chapter.

```
DSNL=DB2UTIL in MAXSPF.TABLES                                    ENTRY DELETED
COMMAND ===>                                                  SCROLL ===> CSR
B -rowse data set   E -dit data set     Y -Data set Utility   X -MAX Data/Util
S -elect entry      D -elete entry      I -nsert entry        C -opy entry
A -dd after         U -pdate entry      R -epeat entry        M -ove entry
       --------------- DATA SETS/COMMANDS/DATABASES ----------------- MEMBER
_     1 * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    ________
_     2 *   DB2 SYSTEM TABLES                                   *    ________
_     3 * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    ________
_     4 DB2 SUBSYS=DSN1 OWNID=SYSIBM   TABLE/VIEW=%                B ________
_     5 DB2 SUBSYS=DSN1 OWNID=SYSIBM   TABLE/VIEW=SYS%             B ________
_     6 DB2 SUBSYS=DSN1 OWNID=SYSIBM   TABLE/VIEW=SYS%             2 ________
_     7 DB2 SUBSYS=DSN1 OWNID=SYSIBM   TABLE/VIEW=LOCATIONS        P ________
_     8                                                              ________
_     9 * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    ________
_    10 *    DB2 TEST TABLES                                      *    ________
_    11 * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    ________
_    12 DB2 SUBSYS=DSN1 OWNID=MA10002  TABLE/VIEW=%                  ________
_    13                                                             ________
_    14 EMP_CURRENT - CURRENT EMPLOYEES                             ________
******************************** Bottom of data ********************************
```

Figure 31: Data Set Name List panel

The following explains the various fields on the sample Data Set Name List panel, which could have been invoked by entering '`DB2UTIL`' as Project name on the command line.

The first line displays information on the current DSNL; you can have more than one DSNL, but this panel will display one at a time. The PDS in which the DSNL is stored is also displayed.

Primary commands may be entered at the `COMMAND ===>` field.

Following are the available functions that can be used.

- The numbers (1-14) to the left of the DSNL indicate the number of the DB2 table to be accessed.
- Line commands may be entered in the column to the left of the DSNL entry numbers. For example, '**E**' for Edit.
- The table as shown in DSNL entry #12 indicates all DB2 tables owned by `MA10002` are to be accessed in the DSN1 DB2 subsystem.
- The text string in entry #14 is an example of a description added to explain the contents of the DSNL entry.

The functions shown above are established when entries are added to the DSNL or when they are updated once they exist in the DSNL.

Creating and Maintaining the DSNL

A DSNL is created using the **CREATE** primary command. For information on procedures for creating a DSNL, refer to the **CREATE** primary command description.

Individual DSNL entries are added or changed by using the (**I**)nsert or (**U**)pdate line commands. For information on and procedures for maintaining a DSNL, refer to the line command descriptions contained in the following section.

Using the Data Set Name List

A unique feature of DSNL is the row numbers to the left of each data set entry in the list. The Data Set Name List panel contains 14 visible entries (6 table, 8 comment). These data set entries are numbered so that they may be identified and processed by actual name or by number. The remaining entries are accessible by scrolling the DSNL forward.

The numbering of the table entries provides users with various methods of using the DSNL to select a table for processing:

1. Type a line command to the left of the table entry, then press ENTER.
2. Specify in the command field the number displayed to the left of the table. This would produce the same result as typing an '**S**' next to the table.
3. Select a table and specify edit by typing the edit command next to the number entered in the command field. For example, typing the characters '`14E`' on the command line would process the fourteenth entry from the DSNL directly, bringing the table up immediately in Edit mode with MAX DB2/UTIL. This same technique can be used for entering any of the line commands as primary commands.

Command Stacking

As you become familiar with your own DSNLs, you will probably find that you have memorized the list number of the most frequently referenced of your databases. Using the ISPF capability of "command stacking" you can access your databases directly from the ISPF Primary Option panel.

In the following example, we stack the command **MAX DSNL** to run the DSNL, followed by the number '14' to select the fourteenth entry in the DSNL for processing. This feature allows you to process any entry in the list without displaying any intervening panels, by entering only a few keystrokes.

```
                          ISPF Primary Option Menu
Option ===> MAX DSNL;14

0  Settings      Terminal and user parameters            USERID    - MAX001
1  MAX/Brow      IAXREXX  Display source data             TIME      - 09:44
2  MAX/Edit      MAXREXX  Change source data             TERMINAL  - 3278
3  Utilities     Perform utility functions               PF KEYS   - 24
4  Foreground    Interactive language processing
5  Batch         Submit job for language processing
6  Command       Enter TSO or Workstation commands
7  Dialog Test   Perform dialog testing
8  LM Facility   Library administrator functions
9  IBM Products  IBM program development products
10 SCLM          SW Configuration Library Manager
I  IMS/UTIL      MXRXV310 IMS DataBase Utilities
D  DB2/UTIL      MXRXV310 DB2 Utilities
M  MAX/DSNL      MXRXV310 Data Set Name List
U  MAX/DUTL      MXRXV310 Data file utilities
V  MAX/PDF       MXRXV310 Dataset and DASD utilities
S  SDSF          Spool Display and Search Facility

Enter X to Terminate using log/list defaults
```

Figure 32: ISPF Primary Option panel

DSNL Primary Commands

Primary commands generally apply to an entire DSNL. Enter primary commands at the `COMMAND ===>` prompt located in the upper left corner of a panel utilizing the following guidelines:

- Enter a blank space to separate command operands. **Do not use the cursor keys** to achieve spacing.
- Insert or expand operands using the insert mode of your keyboard.
- Optionally, enter multiple commands in the `COMMAND` line by entering a semicolon (`;`) between each command. This process is known as "Command Stacking".

The following primary commands are discussed in this section:

C (Change)
CA (Change All)
COPY
CREATE
DELETE
F (Find)
LISTCAT
MOVE
PROFILE
RC (Repeat Change)
RESET
RF (Repeat Find)
RESET
View Another DSNL

C (Change)/CA (Change All)

Use the **C** command to change the next occurrence of a string that is found in the DSNL. Use the **CA** command to change every occurrence of a string in the entire DSNL.

The commands have the following format:

C	**fromstring tostring**
C	**fromstring tostring**
C	**fromstring tostring**
CA	**fromstring tostring**
CA	**fromstring tostring**
CA	**fromstring tostring**
C	(note that a **C** with no data will repeat the prior **C** command)

Operand Definitions

fromstring is the search string. The **fromstring** can be a quoted (case sensitive) or unquoted (case insensitive) string. If the **fromstring** contains embedded blanks, then it must be quoted. If the **fromstring** contains single quotes, then it must be quoted using double quotes. If the **fromstring** contains double quotes, then it must be quoted using single quotes.

tostring will appear in the case that it was entered. However, if the value is a data set name, the editing will force it to uppercase. If the **tostring** contains embedded blanks, then it must be quoted. If the **tostring** contains single quotes, then it must be quoted using double quotes. If the **tostring** contains double quotes, then it must be quoted using single quotes.

The **C** command will look at all of the values that comprise an entry in the DSNL, although some of the values may not be visible. It will begin with the entry currently positioned at the top of the panel. If a match is found, the **fromstring** will be changed to the **tostring** and the entry will be positioned to the top of the panel. Every occurrence of the value in that entry will be changed.

The **CA** command will look at every value in every entry of the DSNL. It will look at values that may not be visible on the panel. If a match is found, the fromstring will be changed to the tostring and the search will continue. Upon completion, the DSNL will be positioned to the first entry in the DSNL.

If no match is found, the DSNL will remain positioned as it was and a message will be returned noting that the value could not be found.

COPY

With the **COPY** command, you can copy an existing DSNL into the DSNL you are viewing. The copied DSNL is inserted after the location you specify with the **A** (Add after) line command (see "*Line Commands Used in DSNL*" on page 102). The copied list remains unchanged.

The **COPY** command has the following format:

COPY dsnlname

In the above command, **dsnlname** is a 1 to 8 character name of an existing DSNL.

CREATE

You can create a new DSNL by using the **CREATE** command.

The **CREATE** command has the following format:

CREATE dsnlname

In the above command, **dsnlname** is a 1 to 8 character name of a DSNL.

Upon execution of this command, the **CREATE** a Project Data Set Name List panel is displayed, allowing you to create a new DSNL using a combination of several different methods to retrieve data set names and include them in the new DSNL.

If END is pressed upon initial entry, a DSNL is created with one null entry.

Option '4' allows you to change the project DSN where DSNLs are stored.

The following example shows the result of entering a **CREATE NEW DSNL** command, to create a new DSNL named NEWDSNL in MAXSPF.TABLES.

```
MAX/PDF --------- CREATE A PROJECT DATA SET NAME LIST     0 ENTRIES IN NEWDSNL
 COMMAND ===>

 Projects data set name (where DSNLs are stored/fetched).
   PROJECT DSN. . . . : MAXSPF.TABLES

 Specify the data set name(s) to include in the new PROJECT DSNL=NEWDSNL

 Select one of the following. Then press Enter.
 _   1 - Select data sets from Catalog
     2 - Select data sets from another DSNL list
     3 - Insert an Entry directly
     4 - Change Project DSN where DSNLs are stored

 Enter a high level qualifier (option 1), or DSNL name (option 2) below:
  DSNAME LEVEL, or DSNL    ===>


  Press END to display newly created DSNL.
 Enter CANCEL to return without creating DSNL.
```

Figure 33: **CREATE** a Project Data Set Name List panel

When the panel is initially displayed, enter an '**S**' line command for each entry you want to include in the new DSNL.

The following panel displays the result of specifying option 1 - Select data sets from Catalog as well as specifying a high level qualifier in the **CREATE DSNL** function.

```
 ------------------------- INSERT ENTRIES FROM CATALOG---------------- IDC3012I
 COMMAND ===>                                                SCROLL ===> CSR
 S -elect entries that you want inserted into  DSNL=NEWDSNL  .
   -------------------------------- DATASETS --------------------------COBOL
 _ MXS.MAXPDF.EXECS
 S MXS.MAXPDF.JCL
 _ MXS.MAXPDF.MESSAGES
 S MXS.MAXPDF.PANELS
 _ MXS.MAXPDF.TABLES
 ******************************Bottom of data********************************
```

Figure 34: Create Selection List panel

In this example, the second and fifth entries in the list were selected to be added by placing '**S**' (as indicated by arrows) next to them, then pressing ENTER.

DELETE

Use the **DELETE** command to eliminate an existing DSNL.

The **DELETE** command has the following format:

```
DELETE dsnlname
```

In the above command, dsnlname is a 1 to 8 character name of an existing DSNL.

If you delete the DSNL you are viewing, a **PROFILE** command display is shown for you to select another DSNL to view. See the PROFILE command on page 99 for additional information.

Before any **DELETE** is processed, an additional panel is displayed to confirm the request.

Note: Deleting a DSNL will not delete the databases in the list.

F (Find)

Use the **F** command to position to the next occurrence of the string to the top of the panel.

The **F** command has the following format:

```
F       string
F       string
F       (entered with no data will repeat the prior find command)
```

Operand Definitions

string can be a quoted (case sensitive) or unquoted (case insensitive) **string**. If **string** contains embedded blanks, then it must be quoted. If **string** contains single quotes, then it must be quoted using double quotes. If **string** contains double quotes, then it must be quoted using single quotes.

The **F** command will look at all of the values that comprise an entry in the DSNL, although some of the values may not be visible. If a match is found, the entry will be positioned to the top of the panel. If no match is found, the DSNL will remain positioned as it was and a message will be returned noting that the value could not be found.

LISTCAT

With the **LISTCAT** command, you can display a list of data sets from the OS catalog.

The **LISTCAT** command has the following format and/or aliases:

LISTCAT	**XXXX.XXXX**	full qualifier
LISTCAT	**XXXX.XXX***	partial qualifier
LISTC	"	
LC	"	

In the above command, qualifier is any high level data set name qualifier. If you omit the qualifier, a panel appears where you can enter it. **LISTCAT** supports both full and partial qualifiers. The high order qualifier, however, must always be fully specified. This is an MVS requirement.

The following example panel displays the result of entering a **LC MXS.MXRXTEST DSNL** command.

```
LISTCAT=MXS.                                                    IDC0002I
COMMAND ===>                                                SCROLL ===> CSR
B -rowse data set   E -dit data set   Y - Data set Utility   X - MAX Data/Util
S -elect entry      C -opy entry
        ------------------ DATASETS ------------------
_     1 MXS.MXRXTEST.BIGLRECL
_     2 MXS.MXRXTEST.FLEET
_     3 MXS.MXRXTEST.KSDS
_     4 MXS.MXRXTEST.SAM
****************************** Bottom of data ********************************
```

Figure 35: **LISTCAT** Primary Command in DSNL panel

MOVE

Use the **MOVE** command to move the contents of an existing DSNL into the DSNL you are viewing. It is inserted after the location specified with the **A** line command and its name is deleted from the list of available DSNLs, upon successful execution of this command.

The **MOVE** command has the following format:

MOVE **dsnlname**

In the above command, dsnlname is a 1 to 8 character name of an existing DSNL

PROFILE

The **PROFILE** command displays a list of all DSNLs currently available to the user.

The **PROFILE** command has the following format:

PROFILE

Upon entry, a panel is displayed allowing you to change the Project DSN where DSNLs are stored. The last 10 Project DSNs used are also displayed and may be selected by placing an 'S' next to one of them. Upon entry, a list of all DSNLs currently available in that Project DSN is then displayed.

There are three line commands that are specific to the **PROFILE** command and made available to users after the PROFILE command list has been displayed. These commands are:

D (Delete) - Line Command
R (Rename) - Line Command
S (Select) - Line Command

The following example panels display the results of the **PROFILE** command in MAX/PDF DSNL.

The current Project DSN of MAXSPF.TABLES is displayed for confirmation or change.

```
DSNL=QADSNL in MAXSPF.TABLES                                    Row 1 of 77
 C .-------------------------------------------------------------------------.
 B | CONFIRM PROJECTS DATA SET NAME                                          | 1
 S | COMMAND ===>                                                            |
 A |                                                                         |
   | Specify the projects data set name (where DSNLs are stored/fetched).    |
 _ |   PROJECT DSN. . . . : MAXSPF.TABLES                                    | OG
 _ |                                                                         | CE
 _ | or type S next to choice and press ENTER to select PROJECT DSN          | AT
 _ |                                                                         | __
 _ | _ => MAXSPF.TABLES                                                      | F
 _ | S => MXS.IMS.TABLES                                                     | __
 _ | _ => MAXSPF.TABLES                                                      | __
 _ | _ => MXS.IMS.TABLES                                                     | __
 _ | _ => MAXSPF.TABLES                                                      | __
 _ | _ => MXS.IMS.TABLES                                                     | __
 _ | _ => MAXSPF.TABLES                                                      | __
 _ | _ => MXS.IMS.TABLES                                                     | __
 _ | _ => MAXSPF.TABLES                                                      | __
 _ | _ => MXS.IMS.TABLES                                                     | __
 _ |                                                                         | __
 _ | Press ENTER to confirm data set name specified.                         | __
 _ | Enter END   to return without changing.                                 | __
 _ '-------------------------------------------------------------------------' __
 _    19 MXS.MAXREXX.TABLES.TEST                                  ________
 _    20 MXS.MAXREXX.LAYOUTS                                      ________
```

Figure 36: **PROFILE** Project DSN Confirmation panel

Entry of an 'S' next to MAX.IMS.TABLES selects it as the new Project DSN whose profile is displayed as follows:

```
LIST ALL DSNLs in MXS.IMS.TABLES                                  Row 1 of 65
 COMMAND ===>                                                  SCROLL ===> CSR
 S -elect for processing            D -elete                         R -ename
   NAME    MSG            -- CREATED --    -- CHANGED --      CUR   MOD USER
 _ ALLIANZ
 _ BENCH
 _ BYTEME
 _ CAV240
 _ CA154
 _ CK4
 _ CMVS
 _ DB2
 _ IBMREXX
 _ IMS
 _ IMSTRIAL
 _ IMS61
 _ JANDSNL
 _ LDCT
 _ MAKEREL
 _ MARTEMP
 _ MARTEST
 _ MAXDEVL
 _ MAXGML
 _ MAXPROJ
```

Figure 37: **PROFILE** Primary Command in DSNL panel

The following three primary commands (**DELETE**, **RENAME**, **SELECT**) are **PROFILE** sub-commands and can only be entered after you issue the **PROFILE** command.

DELETE

To delete a DSNL, type the single letter '**D**' in the command area to the left of the data set. When you press the ENTER key, the DSNL is deleted.

Note: Deleting a DSNL will not delete the data sets in the list.

RENAME

To rename a DSNL, type the single letter '**R**' in the command area to the left of the data set. A panel is then displayed requesting the new name for the data set.

SELECT

To select a DSNL for processing, type the single letter '**S**' in the command area to the left of the data set. Press the ENTER key and the data set name list is presented.

RC (Repeat previous C command)

Use the **RC** command to position to the next occurrence of the string entered in the previous **C** command and make the change as was entered.

RC

No data string is entered with this command. The repeat change will repeat the previous **C** (change) command.

RESET

The **RESET** command clears any outstanding **MOVE** or **COPY** commands, pending '**A**' (Add after) line commands, as well as any **ADD PENDING** or **COPY**/**MOVE** pending commands.

The **RESET** command has the following format and/or aliases:

RESET
RES

RF (Repeat previous F command)

Use the **RF** command to position to the next occurrence of the string found by a prior '**F**' command that has been positioned to the top of the panel.

The **RF** command has the following format:

RF

No data string is entered with this command. The Repeat find will find the data that was entered for the prior **FIND** command.

View Another Data Set Name List

To view a specific DSNL, type the name of the DSNL at the command prompt.

This command has the following format:

dsnlname

In the above command, the **dsnlname** specified is the 1 to 8 character name of an existing DSNL.

Line Commands Used in DSNL

Line commands generally apply to individual data set entries in a DSNL. Enter line commands to the left of a data set name entry. The command will affect only the entry next to it.

The following line commands are discussed in this section:

A	Add after
B	Browse
C	Copy
D	Delete
E	Edit
J	Submit a member to the JES queue from this data set
I	Insert
M	Move
R	Repeat
S	Select
U	Update
X	MAX Data/Util, MAX DB2/UTIL, or MAX IMS/UTIL
Y	Utilities Command

A (Add after)

The **A** (Add after) command is used with primary commands to move or copy entire DSNLs, or with line commands to move or copy DSNL entries.

Typing an '**A**' in the command area to the left of a specified entry inserts the copied or moved DSNL or DSNL entry after that position in the list.

B (Browse)

To browse a DSNL entry, type the single letter '**B**' in the command area to the left of the data set.

The results of the **B** (Browse) command depend upon the type of data set selected. If the selected data set is a partitioned data set (PDS) or PDS concatenation, then the MAX/PDF MSL is displayed and the default line command is browse.

If the selected data set is not a partitioned data set (PDS) or PDS concatenation, the ISPF browse is invoked. If the data set cannot be processed by the ISPF Browse (such as in the case of a VSAM data set), MAX Data/Util is invoked (this requires installation of the companion MAX Data/Util product). If the entry is IMS type, MAX IMS/UTIL is invoked (this requires installation of the companion MAX IMS/UTIL product). If the entry is DB2 type, MAX DB2/UTIL is invoked.

C (Copy)

To copy a DSNL entry, type the single letter '**C**' in the command area to the left of the DSNL entry to be copied and type an '**A**' to the left of the DSNL entry indicating the target location.

Press the ENTER key and the entry with the '`C`' is copied after the target location marked with the '`A`'.

D (Delete)

To delete a DSNL entry, type the single letter '**D**' in the command area to the left of the entry. A confirmation panel will be presented showing the complete detail for the requested entry before the entry is actually removed from the list.

Note: Deleting a DSNL entry will not delete the data set. To delete a data set from disk, use the Utilities sub-command **D** (Delete) (see "*Y (Utilities Commands)*" on page 116 for further information on this line command).

E (Edit)

To edit a DSNL entry, type the single letter '**E**' in the command area to the left of the data set. The results of the **E** (Edit) line command depend upon the type of data set selected. If the selected data set is a partitioned data set (PDS) or PDS concatenation, then the MAX/PDF MSL is displayed. The default line command is Edit.

If the data set cannot be processed by the ISPF editor (such as in the case of a VSAM data set, a SAM file containing records longer than 255 bytes, or an extremely large SAM file), MAX Data/Util will be invoked (this requires installation of the companion MAX Data/Util product). If the entry is IMS type, MAX IMS/UTIL is invoked (this requires installation of the companion MAX IMS/UTIL product). If the entry is DB2 type, MAX DB2/UTIL is invoked.

I (Insert)

To insert a DSNL entry, type the single letter '**I**' in the command area to the left of the data set where an insertion is to be made. The following panel will be displayed to provide choices of what to insert.

```
DSNL=MAXDEVL -------------- INSERT OPTIONS ------------------ USER=MX11003
 COMMAND ===>                                                 DATE=2003/03/05
                                                              TIME=11:28:21
 _   Select type entry to insert.

      1  FUNCTION  - ISPF function
                    (Menu PANEL, PROGRAM, CLIST, REXX EXEC, TSO COMMAND)

      2  DATA SET  - PDS, PDS concatenation, Sequential, or VSAM file

      3  IMS DB    - IMS/UTIL DB

      4  DB2 TABLE - DB2 Database Table

      5  UNIX File - Enter path name

      6  DSNAMES   - Select data sets from Catalog
         Enter a high level qualifer (option 6) below:
         DSNAME LEVEL  ===>

 Press ENTER to continue, or END to return.
```

Figure 38: Insert Options panel

The following display and describe each **I** (Insert) option from the panel above.

Insert: Select option 1. FUNCTION

The following panel is displayed:

```
DSNL=MX10002 ------------- INSERT FUNCTION ------------------ USER=MX10002
 COMMAND ===>                                                  DATE=2002/11/12
                                                               TIME=13:50:12
 FUNCTION - PANEL, CMD or PGM (see syntax below):
 SELECT  ===>

 Syntax:    { PANEL(panel-name) < OPT(option) >                          }
            { CMD(command) < LANG(APL) > < MODE(LINE|FSCR) >             }
            { PGM(program-name) < PARM(parameters) > < MODE(LINE|FSCR) > }
            < NEWAPPL < (application-id) >  < PASSLIB > | < NEWPOOL >  >
 Specify Description to appear in DSNL:
DESCRIPTION ===>
 LIBDEF data sets:                                    STACK/COND/UNCOND/volser
 ISPLLIB ===>                                              ===>
 ISPMLIB ===>                                              ===>
 ISPPLIB ===>                                              ===>
 ISPSLIB ===>                                              ===>
 ISPTABL ===>                                              ===>
 ISPTLIB ===>                                              ===>
         ===>                                              ===>
         ===>                                              ===>
         ===>                                              ===>

 GO process data sets/cmds;  END return with updates;  CANCEL return no updates
```

Figure 39: Insert Function panel

In this example, the program ISPYXDR has been entered with the description of Dialog Test.

Field Descriptions:

COMMAND ===>: Primary commands may be entered at the command line.

SELECT ===>: The target of a valid ISPF **SELECT** function should be entered following the SELECT ===>. The function should be a panel, a TSO command or a program. The syntax is shown below the entry field.

DESCRIPTION ===>: Text may be entered to explain the function. It will be displayed on the DSNL entry line. If nothing is entered, the function will be displayed on that line.

LIBDEF DATA SETS: Use LIBDEF to define the application level libraries that will be in effect while the application is running.

ISPLLIB ===>: Load module library

ISPMLIB ===>: Message library

ISPPLIB ===>: Panel library

ISPSLIB ===>: Skeleton library

ISPTABL ===>: Table output library

ISPTLIB ===>: Table input library

STACK/COND/UNCOND/volser:

STACK: Use this application level library regardless of the existence of an application level library being previously defined. Restore any prior allocations upon completion of using these libraries. This option is the default.

COND: Use this application level library only if no application level library was previously defined.

UNCOND: Use this application level library regardless of the existence of an application level library previously defined.

volser: Enter the volume serial number for USER DEFINED data sets. (See the following.)

USER DEFINED Data Sets: Following the LIBDEF statements are fields available to enter up to 3 user defined data sets to be allocated and used by the function, command or program. Enter the **DDNAME** in front of the arrow and the Data Set name following the arrow. If the data set is not in the catalog, enter the Volser under the column STACK/COND/UNCOND/volser.

Insert: Select option 2. DATA SET

The following Insert Entry panel is displayed. In this example, the data set has been entered.

MAX001.MAXTEST.EXECS

```
DSNL=MAXDEVL -------------- INSERT ENTRY -------------------- USER=MX10002
 COMMAND ===>                                                 DATE=1999/11/02
 A - Allocate new data set    C - Catalog data set            TIME=10:09:25
 R - Rename entire data set   U - Uncatalog data set
 D - Delete entire data set   I - Data set information   X - Compress data set
 *** Delete name with ERASE EOF - overtype to make changes ***

 _   DSN ===> MAX001.MAXTEST.EXECS                        VOLSER ===>
              PDS concatenations to the entry listed above
 _   DSN ===>                                            VOLSER ===>
 _   DSN ===>                                            VOLSER ===>
 _   DSN ===>                                            VOLSER ===>
 _   DSN ===>                                            VOLSER ===>
 _   DSN ===>                                            VOLSER ===>
 _   DSN ===>                                            VOLSER ===>
 _   DSN ===>                                            VOLSER ===>
 _   DSN ===>                                            VOLSER ===>
  MEMBER ===>
COMMANDS ===>
DESCRIPTION ===>
              COBOL Copybook data set and member name ----*
     DSN ===>                                            VOLSER ===>
  MEMBER ===>
 GO process datasets/cmds;  END return with updates;  CANCEL return no updates
```

Figure 40: Insert Data Set panel

The utility functions shown at the top of the Insert Data Set panel may be used with all DSNs shown on the Insert Data Set panel.

Field Descriptions:

COMMAND ===>: Primary commands may be entered at the command line.

DSN ===>: Data set names may be entered on the lines indicated with a DSN ===>. Up to nine data sets may be specified for each DSNL entry; concatenating all data sets specified.

VOLSER ===>: If a data set is not cataloged, the volume serial number must be entered to the right of the data set name after the VOLSER ===>, to uniquely locate and identify the data set.

MEMBER ===>: A member name or pattern may be entered in this field to select a specific member to be edited or browsed or to select a subset of members to be included in the Member Selector List.

COMMANDS ===>: Commands may be entered in this field that will be automatically executed when this entry is selected. Alpha commands bypass the menu and are invoked immediately.

DESCRIPTIONS ===>: A text string may be entered following the DESCRIPTIONS ===> to explain the contents of the DSNL entry.

COBOL Copybook data set and member name —*

The data set name of the library containing the COBOL copybook member shown below may be entered in this field.

The member name of a COBOL copybook may be entered in this field. The presence of a COBOL Copybook data set name and member name specifies that a COBOL copybook will to be used to display information from a SAM or VSAM file when this entry is selected for browse or edit.

Insert: Select option 3. IMS DB

and the following entry panel is displayed:

```
DSNL=TESTIMS ---------------- INSERT IMS DB ----------------- USER=MX11005
COMMAND ===>                                                  DATE=2001/11/06
                                                              TIME=16:44:10
END  to return with updates  CANCEL to return no updates  GO process IMS/UTIL
DOWN to scroll forward       UP     to scroll backward
                                                                 More:     +
SPECIFY PSB PANEL VARIABLES
IMSID        ===> IMS1________
PSB NAME     ===> DFSSAM03
PCB/DBD NAME ===> #1______       (*=Display DB PCB selection list)
IMS RUN MODE ===> DLI            (BMP, DLI)
MAPPING DSN  ===> 'MXS.IMS.COPYLIB'_______________________________
MAPPING MEM  ===> DI21PARC
COMMANDS     ===> ________       (Alpha command is invoked immediately)
DESCRIPTION  ===> IMS/UTIL DFSSAM03/DBPCB01 - DLI MODE_____________________

PROFILE VARIABLES UPDATED
AGN CODE     ===> ________       (BMP run mode only)
DBDLIB 1     ===> 'IMS.DBDLIB'_____________________________________
DBDLIB 2     ===> _____________________________________________
PSBLIB 1     ===> 'IMS.PSBLIB'_____________________________________
PSBLIB 2     ===> _____________________________________________
User LOADLIB ===> _____________________________________________
User RESLIB  ===> _____________________________________________

DLI ALLOCATIONS   DDNAME       DATASET NAME
DB DSN 1     ===> DI21PART     'IMS.DI21PART'_____________________________
DB DSN 2     ===> DI21PARO     'IMS.DI21PARO'_____________________________
DB DSN 3     ===> ________     _____________________________________________
DB DSN 4     ===> ________     _____________________________________________
   ETC.
DB DSN 24    ===> ________     _____________________________________________..
```

Figure 41: Insert IMS DB panel

MAX IMS/UTIL parameters to process `PSB=DFSSAM03  PCB=#1` in `IMSID=IMS1` and DLI run mode have been entered. Upon selection, these and any other entries will be passed to MAX IMS/UTIL for processing.

Field Descriptions:

`COMMAND ===>`: Primary commands may be entered at the command line.

`IMSID ===>`: IMS subsystem ID to connect.

`PSB NAME ===>`: PSB name to process.

PCB NAME ===>: PCB name or relative database PCB number.

IMS RUN MODE ===>: BMP or DLI (offline batch).

MAPPING DSN ===>: The data set name of the library containing the COBOL/PL/I copybook mapping criteria member shown below may be entered in this field.

MAPPING MEM ===>: The member name of a COBOL/PL1 copybook mapping criteria that will be used to display IMS segments in formatted mode. Specify DEFAULT to request default mapping criteria.

COMMANDS ===>: MAX IMS/UTIL menu commands may be entered in this field that will automatically be executed when this entry is selected. Alpha commands bypass the MAX IMS/UTIL menu and are invoked immediately.

DESCRIPTION ===>: A text string may be entered to explain the DSNL entry.

AGN Code ===>: The one-to-eight character group name for inter-region communication security. This value becomes the AGN parameter which is passed to the IMS DFSRRC00 program when initiated to edit, unload, or load database segments in BMP mode. The logged on user must be authorized to use the specified application group name.

DBDLIB 1,2 ===>: The name of up to two partitioned data sets containing standard DBD load modules defined for each IMS database to be accessed. These load modules are produced from the standard output of the IMS DBDGEN process.

PSBLIB 1,2 ===>: The name of up to two partitioned data sets containing standard PSB load modules defined for each IMS database to be accessed. These load modules are produced from the standard output of the IMS PSBGEN process.

User LOADLIB ===>: The name of a partitioned data set (DSORG=PO) that is concatenated second to the TASKLIB data set list.

User RESLIB ===>: The name of an authorized IMS RESLIB partitioned data set that is concatenated third to the TASKLIB data set list. The data set is also concatenated to the DFSRESLB ddname for DLI run mode.

DLI ALLOCATIONS ===>: The DDname and data set name of up to 24 IMS Database data sets to be allocated when running in DLI mode.

Insert: Select option 4. DB2 TABLE

and the following insert entry panel is displayed:

```
DSNL=MX10002 --------------- INSERT DB2 TABLE --------------- USER=MX11002
 COMMAND ===>                                                  DATE=2003/01/29
                                                               TIME=14:21:49
 SPECIFY DB2 PANEL VARIABLES
 OWNER ID        ===> ________          (Wild cards (% or _) may be used)
 TABLE/VIEW NAME ===> ____________________________________________________
                      ____________________________________________________
                      ________________  (Wild cards (% or _) may be used)
 DB2 SUBSYSTEM   ===> ____

 INITIAL DISPLAY ===> FORMATTED__       (Formatted, Horizontal)
 CRITERIA DSN    ===> ____________________________________________________
 CURRENT SQLID   ===> ________
 COMMANDS        ===> ________          (Alpha command is invoked immediately)
 DESCRIPTION     ===> ____________________________________________________

 GO process DB2 table;  END return with updates;  CANCEL return no updates
```

Figure 42: Insert DB2 Table panel

MAX DB2/UTIL parameters to process `DB2 TABLE = SYSIBM.LOCATIONS` in `subsystem = DSN1` have been entered. Upon selection, these and any other entries will be passed to MAX DB2/UTIL for processing.

Field Descriptions:

`COMMAND ===>`: Primary commands may be entered at the command line.

`OWNER ID ===>`: High-level qualifier of DB2 table. Use the '`%`' wildcard when any number of characters can be substituted. Use one or more '`_`' wildcards when a single character can be substituted.

`TABLE/VIEW NAME ===>`: DB2 Table or View name. Use the '`%`' wildcard when any number of characters can be substituted. Use one or more '`_`' wildcards when a single character can be substituted.

`DB2 SUBSYSTEM ===>`: DB2 subsystem ID to connect.

`INITIAL DISPLAY ===>`: Initial display format to use (Formatted, Horizontal).

`CRITERIA DSN ===>`: The optional data set and/or member name in parentheses containing previously saved selection criteria. If member name is not specified, a member list is displayed for selection.

`CURRENT SQLID ===>`: Optional user ID passed to DB2. Blank defaults to log on user ID.

`COMMANDS ===>`: Menu option to be passed to MAX DB2/UTIL when entry is selected. Alpha commands bypass the MAX DB2/UTIL menu and are invoked immediately.

`DESCRIPTION ===>`: A text string may be entered to explain the DSNL entry.

Insert: Select option 5. UNIX File

This option allows you to display a directory in the UNIX file system or a file in the UNIX file system.

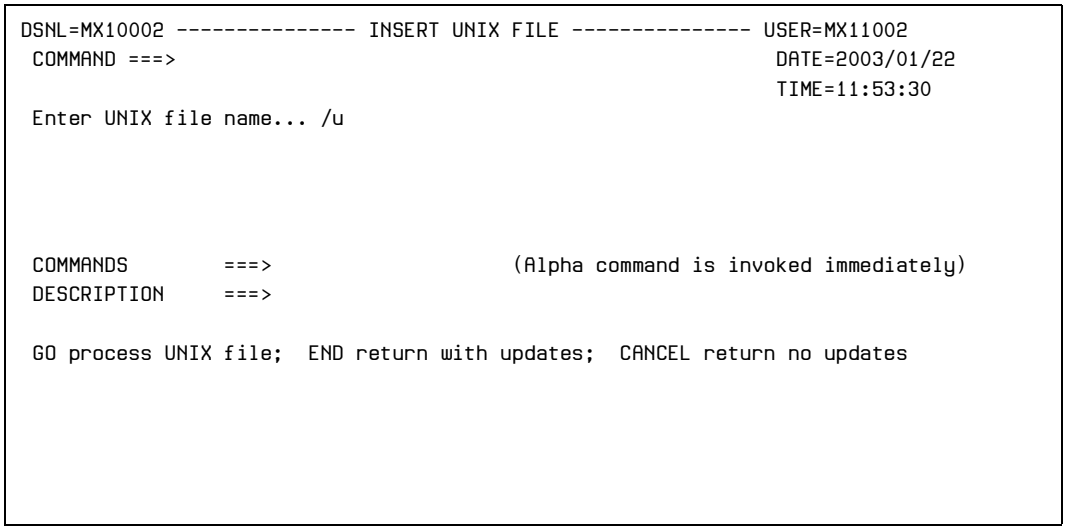

Figure 43: Insert UNIX File panel

Field Descriptions:

UNIX file name ===>: Enter the directory or a complete File name to be viewed.

COMMANDS ===>: Menu option to be passed to the UNIX File display system when the option is selected.

DESCRIPTION ===>: a text string may be entered to explain the DSNL entry.

On the previous panel, the directory description '/U' was entered. When this entry is selected, the directory will be presented as follows:

```
 FSL /u                                                          MAX/PDF VNNN
COMMAND ===>                                                  SCROLL ===> PAGE
A - ttributes    B - rowse    D - elete    E - dit    V - iew
  Message Permissions User    Group Size    Last modified    File name
_          drwx------ MX11002 SYS1  8192 2003/01/13 08:56:34 ./
_          drwxr-xr-x MX11002 SYS1  8192 2003/01/21 14:16:10 ../
_          drwxr-xr-x MX11002 SYS1  8192 2003/01/22 11:55:16 dbond/
_          drwxr-xr-x MX11002 SYS1  8192 2002/11/13 16:21:59 ibmuser/
_          drwxr-xr-x MX11002 SYS1  8192 2000/10/31 15:32:58 open1/
_          drwxr-xr-x MX11002 SYS1  8192 2000/10/31 15:33:04 open2/
_          drwxr-xr-x MX11002 SYS1  8192 2000/10/31 15:33:08 open3/
_          drwxr-xr-x MX11002 SYS1  8192 2000/10/31 15:33:14 p390/
****************************** Bottom of data ********************************
```

Figure 44: View UNIX File panel

Insert: Select option 6. DSNAMES

With associated high level qualifier(s) and a list of data set names from the catalog will be presented. On the Insert Options panel, the high level qualifier(s) for the data sets to be listed must be entered.

For example, if you enter `MXS.MXRXV153` in the DSNAME LEVEL field on the Insert Options panel, the following panel will be presented. If you had entered only `MXS` as a high level qualifier, likely a longer list of data set names would be displayed.

```
------------------------- INSERT ENTRIES FROM CATALOG---------------- IDC3012I
 COMMAND ===>                                                SCROLL ===> CSR
 S -elect entries that you want inserted into  DSNL=MAXDEVL  .
   ------------------------------- DATASETS --------------------------COBOL
 _ MXS.MXRXV153.EXECS
 _ MXS.MXRXV153.JCL
 _ MXS.MXRXV153.LOADLIB
 _ MXS.MXRXV153.LPALOAD
 _ MXS.MXRXV153.MESSAGES
 _ MXS.MXRXV153.OBJECT
 _ MXS.MXRXV153.OBJECTC
 _ MXS.MXRXV153.OBJECTR
 _ MXS.MXRXV153.PANELS
 _ MXS.MXRXV153.TABLES
 _ MXS.MXRXV153.TEMPLOAD
 ******************************** Bottom of data ********************************
```

Figure 45: Insert Entries from Catalog panel

Enter '`S`' to select the entry or entries to be inserted into the DSNL list.

Line Command Descriptions

J (submit a Job)

To submit a member of a data set to the JES job queue, enter a 'J' in the command area to the left of the DSNL entry for the data set that contains the member. Type the member name into the MEMBER field to the right of the data set name. Press the ENTER key. The member will be submitted to the JES job queue for processing.

M (Move)

To move a DSNL entry, type the single letter 'M' in the command area to the left of the DSNL entry to be moved and an 'A' to the left of the DSNL entry that indicates the target location. When you press the ENTER key, the entry with the 'M' is moved after the target location marked with the 'A'.

R (Repeat)

To repeat a DSNL entry, type the letter 'R' in the command area to the left of the entry to be repeated. The DSNL Insert Entry panel is displayed allowing you to make changes to the newly created entry.

S (Select)

To select a DSNL entry for processing, type the single letter 'S' in the command area to the left of the data set.

The results of the **S (Select)** line command depend upon the data set selected. If the entry selected is a partitioned data set (PDS) or PDS concatenation, a MSL is displayed. If the selected entry is a VSAM or sequential file and MAX Data/Util has been installed, MAX Data/Util is invoked. If this selected entry is IMS, MAX IMS/UTIL is invoked (product must be installed). If this selected entry is DB2, MAX DB2/UTIL is invoked (product must be installed).

Point & Shoot is available to select an entry from the DSNL for processing. Position the cursor to the entry in the DSNL and press the ENTER key. This will be the same as entering an 'S' to select the entry for processing.

U (Update)

The **U (Update)** line command is used to update/change a DSNL entry. You issue this command by typing the letter 'U' in the command area to the left of the DSNL entry. When you press the ENTER key, an Update Entry panel is displayed.

X (MAX Data/Util)

MAX Data/Util is a related product for processing data type files; that is, files that are SAM, or VSAM, that contain data which may optionally be mapped using copybooks. MAX Data/Util provides a complete set of data manipulation utilities including: **BROWSE**, **EDIT**, **COPY/EXTRACT**, **UPDATE**, **IDCAMS**, **COMPARE**, **RECORD MAPPING**, **SELECTION CRITERIA**, etc. for processing data files. (This requires installation of the MAX Data/Util product). If IMS type, MAX IMS/UTIL is invoked (if product is installed). If DB2 type, MAX DB2/UTIL is invoked (if product is installed).

Y (Utilities Commands)

The Data Set Utilities panel is similar to other MAX/PDF panels, allowing a choice of many actions from a single panel. Data sets are entered at the "DSN" prompts. The panel is invoked by entering 'Y' to the left of a data set name listed in a DSNL is shown below.

Any of the line commands **A**, **R**, **D**, **C**, **U**, **I** or **X** can be issued to the left of the data set names. This gives you the opportunity to perform any of these functions on any data set(s) displayed on the following panel.

```
DSNL=MAXDEVL ------------ DATASET UTILITIES ----------------- USER=MX10001
 COMMAND ===>                                                 DATE=2000/10/11
                                                              TIME=12:45:57
 A - Allocate new data set               C - Catalog data set
 R - Rename entire data set              U - Uncatalog data set
 D - Delete entire data set              I - Data set information
 X - Compress data set

 *** Enter utility option next to the dataset below ***

 _   DSN ===> MXS.TEST.PDS2                                 VOLSER ===>
              PDS concatenations to the entry listed above
 _   DSN ===>                                               VOLSER ===>
 _   DSN ===>                                               VOLSER ===>
 _   DSN ===>                                               VOLSER ===>
 _   DSN ===>                                               VOLSER ===>
 _   DSN ===>                                               VOLSER ===>
 _   DSN ===>                                               VOLSER ===>
 _   DSN ===>                                               VOLSER ===>
 _   DSN ===>                                               VOLSER ===>
```

Figure 46: Data Set Utilities panel

The following section describes the utility functions that are available when using the MAX/PDF Data Set Utilities panel.

Enter utilities commands to the left of a data set name entry. The command will affect only the entry next to it.

The following utilities commands, as displayed in the previous figure, are discussed in this section:

A	Allocate
C	Catalog
D	Delete
X	Compress
I	Information
R	Rename
U	Uncatalog

A (Allocate)

To allocate a new data set, first type the name of the new data set next to `DSN ===>`, then, the single letter '`A`' in the command area to the left of the data set. A panel is displayed which requests the attributes for the new data set.

Note: If you want to model the attributes of the new data set after an existing data set you should start by displaying the attributes of that data set with the **I (Information)** command.

```
Menu  RefList  Utilities  Help
 ------------------------------------------------------------------------------
                            Allocate New Data Set
 Command ===>
                                                                        More:     +
 Data Set Name  . . . : MXS.TEST.NEWPDS

 Management class . . .                (Blank for default management class)
 Storage class  . . . .                (Blank for default storage class)
  Volume serial . . . . MAX003         (Blank for system default volume) **
  Device type . . . . .                (Generic unit or device address) **
 Data class . . . . . .                (Blank for default data class)
  Space units . . . . . CYLINDER       (BLKS, TRKS, CYLS, KB, MB, BYTES
                                        or RECORDS)
  Average record unit                  (M, K, or U)
  Primary quantity  . . 1              (In above units)
  Secondary quantity    1              (In above units)
  Directory blocks  . . 0              (Zero for sequential data set) *
  Record format . . . . FBA
  Record length . . . . 121
  Block size  . . . . . 8107
  Data set name type  : PDS            (LIBRARY, HFS, PDS, or blank)  *
                                       (YY/MM/DD, YYYY/MM/DD
  Expiration date . . .                 YY.DDD, YYYY.DDD in Julian form
```

Figure 47: **A(llocate)** New Data Set panel

C (Catalog)

To catalog a data set, type the single letter 'C' in the command area to the left of the data set. You must specify the VOLSER where the data set resides to catalog a data set.

X (Compress)

To compress a data set, type the single letter 'X' in the command area to the left of the data set. No confirmation panel is displayed.

Note: You cannot compress VSAM data sets with this command.

D (Delete)

To delete an entire data set from disk, type the single letter 'D' in the command area to the left of the data set. A confirmation panel is displayed before the data set is deleted. Once the data set is deleted, there is no way to undelete the data set.

Note: You cannot delete VSAM data sets with this command.

I (Information)

To display data set information, type the single letter 'I' in the command area to the left of the data set. A panel showing the attributes of the data set is displayed.

Note: You cannot display information on VSAM data sets with this command.

The following panel is presented as a result of using the **I (Information)** utilities command in MAX/PDF.

```
Data Set Information
 Command ===>

 Data Set Name  . . . : MXS.TEST.PDS2

 General Data                          Current Allocation
  Volume serial . . . : MAX004          Allocated cylinders : 1
  Device type . . . . : 3380            Allocated extents . : 1
  Organization  . . . : PO              Maximum dir. blocks : 25
  Record format . . . : FB
  Record length . . . : 80
  Block size  . . . . : 3120           Current Utilization
  1st extent cylinders: 1               Used cylinders  . . : 1
  Secondary cylinders : 0               Used extents  . . . : 1
                                        Used dir. blocks  . : 1
  Creation date . . . : 2000/10/11      Number of members . : 0
  Referenced date . . : 2000/11/02
  Expiration date . . : ***None***
```

Figure 48: **I(nformation)** Utilities Command panel

R (Rename)

To rename an entire data set, type the single letter '`R`' in the command area to the left of the data set. A panel is displayed requesting the new name for the data set.

Note: You cannot rename VSAM data sets with this command.

U (Uncatalog)

To uncatalog a data set, type the single letter '`U`' in the command area to the left of the data set. No confirmation panel is displayed. Once the data set is uncataloged, the only information remaining in the system about it will be in the VTOC of the VOLSER where it resides, if it exists.

Note: You cannot uncatalog VSAM data sets with this command.

INDEX

D

E

P

Q

R

S

T

U

V

W