

IBM DB2 Table Editor



User's Guide

Version 4 Release 3

IBM DB2 Table Editor



User's Guide

Version 4 Release 3

Note:

Before using this information and the product it supports, be sure to read the general information under "Notices."

Second Edition (November 2002)

This edition applies to the following releases and to all subsequent releases and modifications until otherwise indicated in new editions:

- IBM® DB2® Table Editor for Multiplatforms, Version 4 Release 3 Modification 2 (program number 5724-B33)
- IBM DB2 Table Editor for iSeries™, Version 4 Release 3 Modification 2 (program number 5697-G84)
- IBM DB2 Table Editor for z/OS™, Version 4 Release 3 Modification 2 (program number 5697-G65)

© Copyright International Business Machines Corporation 2001, 2002. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© Copyright Rocket Software 2001, 2002.

Contents

About this book	vii
Who should read this book	vii
Service updates and support information	vii
Where to find information	vii
How to send your comments	vii
Summary of changes	ix
Enhancements to DB2 Table Editor components	ix
Enhancements to DB2 Table Editor ISPF component	x

Part 1. Introduction 1

Chapter 1. Overview of IBM DB2 Table Editor	3
DB2 Table Editor connectivity	3
Call Level Interface (CLI)	3
Distributed Relational Database Architecture (DRDA)	4
Communications	4
LU 6.2 and CPI-C	4
TCP/IP and WinSock	4
Hardware Requirements	5
Software Requirements	5
Licensing	5
Chapter 2. Configuring your network environment	7
Configuring your SNA environment	7
Configuring your TCP/IP environment	8
Configuring your CLI environment	9
Chapter 3. Configuring the iSeries Environment	11
Terminology	11
Authorities	11
DRDA Port	11
To verify that the DRDA port is in Listen state:	11
User Profile CCSID	12
Setting Server Parameters	12
To Set Server Parameters:	12
Enabling DB2 Table Editor to list iSeries Physical and Logical files	13
To enable DB2 Table Editor to list iSeries Physical and Logical Files:	13
Chapter 4. Installing DB2 Table Editor's Graphical User Interface	15
To install DB2 Table Editor:	15
Advanced installation	15
Configuring for Unattended Installation	15
File server installation	18
Chapter 5. DB2 Table Editor Concepts	21

Part 2. DB2 Table Editor Console 23

Chapter 6. Overview of DB2 Table Editor Console	25
Servers	25
Maintaining packages	25

Governing	26
Sample Tables	26
Chapter 7. Using DB2 Table Editor Console	27
Starting DB2 Table Editor Console.	27
Using SNA networks	27
Configuring DB2 Table Editor.	27
Creating a server	27
Testing the connection to the database server	29
Binding the DB2 Table Editor packages	29
Setting up governing	32
Sample tables	36
Changing your password in DB2 Table Editor.	36
To Change your password.	37
Configuring a Server that has bound packages associated with it	37
To configure a server that already has packages associated with it:	37

Part 3. DB2 Table Editor Developer 39

Chapter 8. Overview of DB2 Table Editor Developer	41
What is DB2 Table Editor Developer?	41
What can I do with DB2 Table Editor Developer?	41
Chapter 9. Working with DB2 Table Editor Developer	43
Creating new forms	43
Creating a new form using the Forms wizard	43
Creating a form manually	45
Creating a new form by importing a panel from DBEDIT.	48
Working with forms	49
Opening a form.	49
Setting general form attributes	50
Saving a form	51
Printing a form	52
Closing a form	52
Deleting a form	52
Adding a control to a form	52
Deleting a control from a form	53
Setting control properties	53
Creating a LOB control on a form	54
To create a LOB control on a form:	54
Adding an accelerator key.	55
To add an accelerator key to a button control:	55
Using special command characters	56
Working with validation rules	57
Creating a validation rule	57
Creating a validation group	58
Deleting a validation rule	58
Deleting a validation group	59
Locking when Editing a Table	59
Migrating Forms to a Different Environment	60
To migrate a form to a different environment:	60

Part 4. DB2 Table Editor Java Player and DB2 Table Editor User 63

Chapter 10. Overview of the DB2 Table Editor Component and the Java Player Component	65
Chapter 11. Installing DB2 Table Editor Java Player	67
Installing DB2 Table Editor Java Player	67
Installing DB2 Table Editor Java Player as an application:	67
Installing DB2 Table Editor Java Player as an Applet	68
Command Line Parameters	71
Chapter 12. Working with the DB2 Table Editor Component and the Java Player Component	73
Opening the application.	73
Opening DB2 Table Editor Windows User	73
Opening DB2 Table Editor Java Player component.	73
Connecting to an Informix Database using the DB2 Table Editor Java Player component	73
To connect to an Informix database using DB2 Table Editor Java Player component:	73
Opening a form.	74
To open a form that is stored locally:	74
To open a form that is stored on a server using DB2 Table Editor for Windows:	74
To open a form that is stored on a server from DB2 Table Editor Java Player component:	74
Editing a table	75
Editing a table	75
Working with primary keys.	76
Working with the Table Editor	77
Working with Rows in DB2 Table Editor	77
Working with Columns in DB2 Table Editor	78
Working with Cells in DB2 Table Editor	79
Saving work to the database.	79
Working with LOBs (Large Objects)	79
Indicating which application will be used to open Large Objects (LOBs)	80

Part 5. DB2 Table Editor Control Center Plug-in 81

Chapter 13. Overview of the DB2 Table Editor Control Center plug-in	83
Using the DB2 Table Editor Control Center Plug-in.	83
Chapter 14. Installing the DB2 Table Editor Control Center Plug-in	85
Installing the DB2 Table Editor Control Center Plug-in:	85
To install the DB2 Table Editor Control Center Plug-in:	85

Part 6. DB2 Table Editor for the ISPF interface. 87

Chapter 15. Overview of DB2 Table Editor, ISPF Component.	89
Chapter 16. Customizing DB2 Table Editor for ISPF	91
Licensing	91
Customization summary	91
Customizing DB2 Table Editor	91
Step 1. Create the control file	91
Step 2. Bind the packages	91
Step 3. Bind the plans	92

Step 4. Bind the packages and plans for all other sub systems to which you want to connect.	92
Step 5. Edit SETISAMP members ETI and ETICLIST.	92
Step 6. Optional: Add DB2 Table Editor to the DB2 Administration Tool Launchpad	94
Completing customization	95
Step 1. Start DB2 Table Editor	95
Step 2. Setup the configuration file for DB2 Table Editor.	95
Step 3. Set the DB2 subsystem parameters	96
Step 4. Setup the configuration file and set the DB2 subsystem parameters for all other DB2 Subsystems to which you want to connect	97
Setting up a data sharing environment	97
Permissions	97
Chapter 17. Working with DB2 Table Editor for ISPF	99
Getting started	99
The DB2 Table Editor main panel	99
Opening a table	100
To open a table:	100
Editing a table.	103
Locking	103
Updating.	103
Inserting a blank row	104
Repeating rows	105
Copying rows	105
Editing a row	106
Editing a long cell	106
Getting information about a column heading.	108
Deleting rows	108
Undoing an action	108
Committing changes to DB2	109
Searching a table	109
Searching a table and changing the contents of cells in the table	110

Part 7. Appendixes 113

Appendix A. Troubleshooting	115
Troubleshooting for DB2 Table Editor Administrative Console Component, DB2 Table Editor Component and Java Player Component.	115
SDF (server definition file) does not appear in Server List.	115
Installing DB2 Table Editor Java player as an applet.	115
Appendix B. DB2 Table Editor ISPF Component Commands	117
Appendix C. The use of color in DB2 Table Editor for ISPF.	121
Appendix D. Data Type Abbreviations Used in DB2 Table Editor for ISPF	123
Appendix E. Notices	125
Trademarks.	127
Index	129

About this book

This book contains information on installing and using DB2[®] Table Editor.

Who should read this book

This book is intended for all IBM[®] DB2 Table Editor users including, developers, database administrators and system administrators.

Service updates and support information

To find service updates and support information, including software FixPaks, PTFs, Frequently Asked Questions (FAQs), technical notes, troubleshooting information, and downloads, refer to the following address:

www.ibm.com/software/data/db2imstools/support.html

Where to find information

The Data Management Tools Library Web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following Web address:

www.ibm.com/software/data/db2imstools/library.html

IBM Redbooks[™] that cover Data Management Tools are available from the following Web address:

www.ibm.com/software/data/db2imstools/support.html

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other IBM DB2 Table Editor documentation:

- Use the online reader comment form located at www.ibm.com/software/data/db2imstools/rcf/
- Send your comments by e-mail to dmtinfo@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of IBM DB2 Table Editor, and, if applicable, the specific location of the text that you are commenting on (for example, a page number or table number).
- Fill out one of the forms at the back of this book and return it by mail, by giving it to an IBM representative, or by faxing it to 1-800-426-7773.

Summary of changes

This section summarizes significant changes that are implemented in this release.

Enhancements to DB2 Table Editor components

The following enhancements apply to these components:

- DB2 Table Editor Component
- DB2 Table Editor Java™ Player Component
- Db2 Table Editor Administrator Component
- DB2 Table Editor Developer Component

Support for viewing and editing LOB columns:

You can now view and edit LOB columns.

Benefit: You can now work with LOB columns without leaving DB2 Table Editor.

Stored procedure support for populating lists:

You can now populate lists with stored procedures.

Benefit: You can now use stored procedures in lists.

New button actions:

There are 2 new button actions available to DB2 Table Editor forms developers:

- Run an SQL action - this allows the running of user defined SQL statements or stored procedures.
- Open Form - this allows the creation of linked forms

Benefit: These two new button actions broaden the spectrum of actions available when designing forms.

New table editing features:

With DB2 Table Editor Version 4.2, you can now do the following:

- Find and replace within selected rows, columns, or the entire grid
- Columns in the grid:
 - Can be sorted
 - The order can be changed
 - Can be locked
 - Can be set to force uppercase
- Rows can be inserted, deleted or duplicated.
- Cells can be edited, copied, cut, pasted, launched or zoomed
- Visual indicator for primary keys
- Save at end option that enables editing without committing changes until the end of the edit session.

Benefit: These new features increase the range of table editing activities that you can do without leaving DB2 Table Editor.

Referential integrity support:

You can now use DB2 Table Editor to update primary keys and edit related tables.

Benefit: You can update primary keys and edit tables from within DB2 Table Editor, instead of having to go outside of DB2 Table Editor to do this.

Enhancements to the Edit Table wizard:

You can now do the following in the Edit Table Wizard:

- Select which columns to include in a form
- Specify row and sort conditions to determine how your data will be displayed
- Specify row limits
- Specify the locking mode for the table that is being edited
- You can view the data in a table in read only mode
- Table editing specifications can be saved and reused

Benefit: The enhancements to the Edit Table wizard make it easier to edit tables in a way that is tailored to your needs.

DB2 Control Center plug-in extension:

The DB2 Control Center plug in allows you to access DB2 Table Editor directly from the DB2 Control Center.

Benefit: From the DB2 Control Center, you can select the table that you want to edit and DB2 Table Editor will open with that table open and ready to edit.

Enhancements to DB2 Table Editor ISPF component

The following enhancements apply to the DB2 Table Editor ISPF component:

Row-at-a-time Editor/Form Mode:

You can now view data vertically in DB2 Table Editor ISPF component, one row at a time.

Benefit: In form mode, you can move from one row to another (go to the next row or the previous row). You can also zoom into large columns. At any time, you can go back to tabular mode.

Edit Remote Data via Alias:

With proper licensing, DB2 for z/OS™ and OS/390® tables residing at remote locations can be edited by specifying the defined alias name.

Benefit: You can now use DB2 Table Editor ISPF component to access data that resides in databases in multiple locations.

INSERT into Empty Table:

If you select a table that contains no data, DB2 Table Editor ISPF component will display an empty table and allow you to insert one or more rows of data. Once data is inserted, you can edit as usual.

Benefit: You can now use DB2 Table Editor ISPF component to populate DB2 Tables.

VARCHAR Updates in Zoom mode:

When making changes to VARCHAR columns on the Zoom Panel, DB2 Table Editor ISPF Component will automatically calculate the length of the column based on the input value and trailing "pad" characters.

Benefit: This helps make using DB2 Table Editor ISPF component easier.

Flexible Null Indicator:

If you edit a value that is NULL, Table Editor will automatically turn the NULL indicator off.

Benefit: This prevents the loss of any changes made to your data.

Abbreviated or Full Data Type Names:

You can choose to see abbreviated or full data type names on the column selection panel.

Benefit: If you prefer to see the full data type names when working in DB2 Table Editor ISPF component this is now possible.

Undo for Change:

You can now use the Undo line command on rows that are changed. The row changes will be undone and reset to their original values.

Benefit: Now, along with updated and deleted rows, the actions done to changed rows can now be undone.

Improved Scrolling:

Numerous scrolling enhancements have been made to DB2 Table Editor ISPF component.

Benefit: Persistent cursor positioning and column justified scrolling are now supported making working in DB2 Table Editor ISPF component easier to use.

Display Current[®] Row Position:

Two row counters are displayed. The first counter indicates the number of the row currently displayed at the top of the tabular section of the ISPF panel. The second counter indicates the total number of rows available for display. For example, "Row 5 of 3000".

Benefit: It is now easier to keep track of where you are within a table when you are editing or browsing that table.

Thousands Separator in Numeric Columns:

Table Editor now shows you the thousands separator (comma, period) in a numeric column.

Benefit: This makes numbers greater than 999 easier to read.

Part 1. Introduction

DB2 Table Editor is a product that makes it easy to view and modify your DB2 tables. The primary function of DB2 Table Editor is to modify data stored in any database in IBM's DB2 family of databases. There are four different components of DB2 Table Editor that serve different purposes.

- DB2 Table Editor Console component - allows you to configure your connection to DB2 Databases.
- DB2 Table Editor component - allows you to edit DB2 tables using custom made forms (created in the DB2 Table Editor Developer component) or through creating a form as you go along.
- DB2 Table Editor Java player component - allows you to edit DB2 tables using custom made forms (created in the DB2 Table Editor Developer component) or through creating a form as you go along through a Java interface.
- DB2 Table Editor Developer component - Allows you to create custom forms to be used to edit DB2 tables.

Chapter 1. Overview of IBM DB2 Table Editor

DB2 Table Editor is a product that makes it easy to view and modify your DB2 tables. The primary function of DB2 Table Editor is to modify data stored in any database in IBM's DB2 family of databases. There are four different components of DB2 Table Editor that serve different purposes.

- The DB2 Table Editor Console component - allows you to configure your connection to DB2 Databases.
- The DB2 Table Editor component - allows you to edit DB2 tables using custom made forms (created in the DB2 Table Editor Developer component) or through creating a form as you go along.
- The DB2 Table Editor Java player component - allows you to edit DB2 tables using custom made forms (created in the DB2 Table Editor Developer component) or through creating a form as you go along through a Java interface.
- The DB2 Table Editor Developer component - Allows you to create custom forms to be used to edit DB2 tables.

DB2 Table Editor can also be incorporated into the DB2 Control Center as an add-in so that you can launch DB2 Table Editor directly from the table that you want to edit. For more information on the DB2 Table Editor Control Center Plug-in, see Chapter 13, "Overview of the DB2 Table Editor Control Center plug-in" on page 83.

DB2 Table Editor connectivity

In order to access data stored in a DB2 database, DB2 Table editor must connect to that database. There are two ways in which DB2 Table Editor components can connect to DB2:

- Using the Open Group's Distributed Relational Database Architecture™ (DRDA®)
- Using the DB2 for Multiplatforms on Windows® Call Level Interface (CLI)

Call Level Interface (CLI)

Included as part of the DB2 for Multiplatforms on Windows client is a call level interface (CLI) component. CLI is an application programming interface for relational database access. Using CLI, client applications (such as DB2 Table Editor) can connect to DB2 for Multiplatforms servers and run SQL statements. In addition, with DB2 Connect™, CLI applications can connect to mainframe (z/OS, OS/390 or VM & VSE) databases.

In order for DB2 Table Editor to use CLI to connect to DB2, you first use the DB2 for Multiplatforms facilities to define your database servers and how to connect to them. This configuration is outside the control of DB2 Table Editor. Once this configuration is complete, however, DB2 Table Editor simply needs the alias defined for a particular database in order to connect to it.

The primary advantage to using CLI is simplified configuration, in that if a database is already defined in the DB2 for Multiplatforms client, its alias is the only piece of configuration information needed in DB2 Table Editor. The disadvantages to using CLI are that DB2 Table Editor performance using CLI is generally substantially worse than when using a DRDA connection, and that DB2 Table Editor only supports connecting to workstation and MVS™ or z/OS and OS/390 databases with CLI (you must use DRDA connections to access VSE & VM or AS/400® databases).

Distributed Relational Database Architecture (DRDA)

DB2 Table Editor and DB2 are both distributed relational database applications that operate together in a client/server relationship. Each component plays a separate role in this relationship: DB2 Table Editor as the client or requester, and DB2 as the server. DB2 Table Editor and DB2 implement and adhere to a common architecture, the Open Group's Distributed Relational Database Architecture (DRDA). This architecture is a comprehensive and detailed blueprint that specifies the layers and functions required in a client/server distributed database application. Because DB2 Table Editor implements the DRDA requester specification, it is capable of connecting to any database that adheres to and implements the DRDA server architecture. Here are the IBM database products that contain a DRDA server component and are capable of communicating with DB2 Table Editor:

- DB2 for Multiplatforms for z/OS and OS/390, DB2 for z/OS and OS/390, and DB2 for MVS
- DB2 Server for VSE & VM and SQL/DS™
- DB2 for AS/400
- DB2 for Multiplatforms and DB2 Common Server
- DB2 Parallel Edition
- DataJoiner®

Your DB2 Table Editor license determines which of the DB2 family of products you can connect to with your copy of DB2 Table Editor.

Communications

One component of DRDA describes the communications protocol that participants in the architecture must use. It specifies that requesters and servers must communicate either via the SNA LU 6.2 architecture, or TCP/IP protocols.

Tip: When using CLI to connect to a database, you must perform similar network configuration; however, this configuration is performed as part of the DB2 client configuration, rather than as part of the DB2 Table Editor configuration. The following discussion on configuring your SNA and TCP/IP environments applies only to DRDA connections.

LU 6.2 and CPI-C

LU 6.2 is an SNA communications architecture. APPC (Advanced Program-to-Program Communication) is a language based on the LU 6.2 architecture. A developer of SNA transaction programs has many different implementations of APPC from which to choose. Even though each implementation of APPC adheres to the LU 6.2 architecture, two implementations of APPC might not be the same. Therefore, programs that rely on one vendor's APPC implementation might not work with another vendor's implementation. CPI-C (Common Programming Interface-Communications) is a common programming interface that solves this problem.

CPI-C is a programming interface that implements the APPC verb set. Therefore, applications that require the use of the APPC verb set can instead be written using CPI-C in order to achieve SNA vendor independence. DB2 Table Editor is an application that is written using CPI-C.

TCP/IP and WinSock

TCP/IP is a collection of protocols. WinSock (Windows Sockets) is a standard, common programming interface that implements the TCP protocol. Applications that

require the use of the TCP/IP protocols can be written using WinSock to achieve TCP/IP vendor independence. DB2 Table Editor is an application that is written using WinSock.

Hardware Requirements

Hardware requirements are those needed to run the Windows software as described in "Software Requirements."

DB2 Table Editor has the following requirements:

- Approximately 16 MB of free disk space on a Windows machine
- A minimum of 32 MB memory. Additional memory improves performance.

Software Requirements

DB2 Table Editor runs on the following operating systems:

- Microsoft® Windows 95
- Microsoft Windows 98
- Microsoft Windows NT® 4.0
- Microsoft Windows 2000
- Microsoft Windows XP

Licensing

IBM DB2 Table Editor can access DB2 on a variety of platforms depending on which License Use Management (LUM) key that you have. The three different versions are as follows:

1. Using the Multiplatform LUM key, you can access DB2 running on the following platforms:
 - Linux
 - Linux 390
 - AIX®
 - SUN Solaris
 - HP-UX
 - OS/2®
 - SCO
 - NUMA-Q®
 - Windows NT
 - Windows 2000
2. Using the iSeries™ LUM key, you can access DB2 running on the following platforms:
 - AS/400
 - iSeries
3. Using the z/OS LUM key, you can access DB2 running on the following platforms:
 - z/OS
 - MVS
 - OS/390

The DB2 Table Editor license is enforced based on the platform to which IBM DB2 Table Editor is connecting (The platform where DB2 resides).

You can have more than one License Use Management key installed. This would permit you to connect to DB2 on more than one platform. If you want to add an additional license, the new key must be installed by reinstalling the product using a version that includes the new LUM key in the `disk1` directory. This process does the license enrollment as part of the install, but does not overwrite any existing keys.

Note: The java code does not use the enrollment process to validate the keys. For the DB2 TableEditor Java player application and applet, copy the *.lic file that you want to add to the appropriate directory. This will activate the key.

Chapter 2. Configuring your network environment

Before you install DB2 Table Editor, you need to configure the required network infrastructure. DB2 Table Editor Technical Support cannot provide support for configuring your network infrastructure.

Configuring your SNA environment

In an SNA network, DB2 Table Editor must be able to establish an LU 6.2 session with DB2, using the CPI-C interface. This connectivity is not provided with DB2 Table Editor. The third-party product that you use to provide connectivity must be installed, configured, and working before you install or use DB2 Table Editor. The process of implementing LU 6.2 connectivity between Microsoft Windows and DB2 can be a complex task, depending on your SNA network environment. The Windows-based SNA products that can be used, and the different ways those products can be used, are far too numerous and complex for this book to describe in detail. You must rely on your in-house SNA networking staff and your SNA software vendor's technical support services to implement and support your network configuration. DB2 Table Editor Technical Support cannot provide support for these issues.

Note: For more information on LU 6.2 and CPI-C, see "LU 6.2 and CPI-C" on page 4.

The following table indicates the configurations with which DB2 Table Editor has been tested and officially supports. The columns under Win 3.x, Win 9x, and Win NT/2000 indicate whether the 16-bit version or the 32-bit version of DB2 Table Editor works with the listed SNA product on that platform.

Table 1. DB2 Table Editor Configurations

SNA Product	Win 3.x	Win 9.x	Win NT/2000
Attachmate EXTRA! APPC Client 3.11	16-bit		
Attachmate EXTRA! for Windows 95 6.2		32-bit	
Attachmate EXTRA! for Windows 95 6.4		32-bit	
Attachmate EXTRA! for Windows NT 6.2			32-bit
IBM APPC Networking Services for Windows 1.0	16-bit		
IBM Communications Server for OS/2 Warp 4.1	16-bit		
IBM Personal Communications AS/400 and 3270 4.1 for Windows 95		32-bit	
IBM Personal Communications AS/400 and 3270 4.1 for Windows NT			32-bit
IBM Personal Communications AS/400 and 3270 4.1 for OS/2	16-bit		
IBM Personal Communications 4.2 for Windows 95		32-bit	
Microsoft SNA Server 2.11, 3.0, or 4.0 - Windows 3.x Client	16-bit	16-bit	16-bit
Microsoft SNA Server 2.11, 3.0, or 4.0 - Windows 95 Client		32-bit	

Table 1. DB2 Table Editor Configurations (continued)

SNA Product	Win 3.x	Win 9.x	Win NT/2000
Microsoft SNA Server 2.11, 3.0, or 4.0 - Windows NT Client			32-bit
NetSoft DynaComm/Elite 3.51 APPC	16-bit		
Novell Netware for SAA® 2.0	16-bit		
Novell Netware for SAA 2.2		32-bit	32-bit
WallData Rumba 95/NT 5.0		32-bit	32-bit

DB2 Table Editor should work with other products that provide a WinCPIC interface. The products listed are simply the ones that have been tested and are supported.

Note: Make sure that you have the latest corrective service or maintenance for your SNA product. Contact your SNA software vendor's technical support services personnel for this information.

Configuring your TCP/IP environment

To access a DB2 server using TCP/IP, DB2 Table Editor must be able to establish a TCP/IP connection from the local host (the system on which DB2 Table Editor is running) to the remote host (the system on which DB2 is running) and remote port (the port on which DB2 is listening).

You can connect to DB2 using TCP/IP from DB2 Table Editor on iSeries. For information on doing this, see "To Set Server Parameters:" on page 12.

The process of implementing TCP/IP connectivity between Microsoft Windows and DB2 is generally a much easier task than in the corresponding SNA scenario. You must rely on your in-house TCP/IP networking staff and your TCP/IP software vendor's technical support services to implement and support your network configuration. DB2 Table Editor Technical Support cannot provide support for these issues. DB2 Table Editor requires WinSock 1.1 to interface to the installed TCP protocol stack. The following table indicates the configurations with which DB2 Table Editor has been tested and officially supports. The columns under Win 3.x, Win 9x, and Win NT/2000 indicate whether the 16-bit version or the 32-bit version of DB2 Table Editor works with the listed TCP/IP product on the specified operating system

Table 2. DB2 Table Editor Tested/Supported Configurations

TCP/IP stack	Win 3.x	Win 9x	Win NT
Microsoft Windows 95 TCP protocol		16-bit, 32-bit	
Microsoft Windows NT TCP protocol			16-bit, 32-bit
Microsoft TCP/IP-32 for Windows for Workgroups 3.11b	16-bit		

DB2 Table Editor should work with other products that provide a WinSock 1.1 interface. The products listed are simply the ones that have been tested and are supported.

Configuring your CLI environment

To access a DB2 server using CLI, the 32-bit version of DB2 Table Editor must be able to establish a CLI connection from the local host (the system on which DB2 Table Editor is running) to the remote host (the system on which DB2 UDB is running) via the DB2 for Multiplatforms client.

The process of implementing CLI connectivity between Microsoft Windows and DB2 is generally the most simple form of connectivity to implement. All of the connectivity information is defined in the DB2 for Multiplatforms client. You must still rely on your in-house networking staff to implement and support your network configuration. DB2 Table Editor Technical Support cannot provide support for these issues. DB2 Table Editor can access databases through DB2 for Multiplatforms client version 5.2 or higher. CLI connections are supported for the following databases: DB2 for Multiplatforms, DB2 Common Server, DB2 for MVS, DB2 for z/OS and OS/390 and DB2 DataJoiner. To connect to DB2 for MVS or z/OS and OS/390, you must also have DB2 Connect installed locally or as a gateway.

Note: Although DB2 TableEditor supports CLI connectivity to DB2 for MVS and DB2 for z/OS and OS/390, you are strongly urged to use a DRDA connection for your DB2 for MVS or DB2 for z/OS and OS/390 databases.

Chapter 3. Configuring the iSeries Environment

Before using DB2 Table Editor to connect to a database on the iSeries platform it is necessary to ensure that your iSeries environment is configured properly.

Terminology

DB2 Table Editor uses SQL Terminology. The following table shows the relationship between iSeries system terms and SQL relational database terms.

Table 3. The relationship between iSeries and SQL terminology

iSeries Terms	SQL Terms
Library - Groups related objects and allows you to find the objects by name.	Collection - Consists of a library, a journal, a journal receiver, an SQL catalog, and optionally a data dictionary. A collection groups related objects and allows you to find the objects by name.
Physical file - A set of records.	Table - A set of columns and rows.
Record - A set of fields.	Row - The horizontal part of a table containing a serial set of columns.
Field - One or more characters of related information of one data type.	Column - The vertical part of a table of one data type.
Logical file - A subset of fields and records of one or more physical files.	View - A subset of columns and rows of one or more tables.
SQL Package - An object type that is used to run SQL statements.	Package - An object type that is used to run SQL statements.
User Profile	Authorization name or Authorization ID .

Authorities

- DB2 must be configured to accept the requests from DB2 Table Editor and the users must have the proper authorities specified in DB2 for the requests to be carried out.
- The user profile that is used to install DB2 Table Editor must have SECADM authority.

DRDA Port

The DRDA Port must be in Listen state in order for the installation to complete successfully.

To verify that the DRDA port is in Listen state:

1. Type NETSTAT on the AS/400 command line.
2. Select Option 3 (Work with TCP/IP connection status).
3. Locate for DRDA under the Local Port column and ensure that the State = Listen. If the State = Listen, go to the next section, if not continue to step 4.
4. If DRDA is not there, DDM TCP/IP services must be started on the AS/400. To start the DDM TCP/IP server:
 - a. Type STRTCPSVR *DDM on the iSeries command line.

- b. Locate DRDA under the Local Port column and ensure that the State = Listen. If the State = Listen, go to the next section, if not continue to the User Profile CCSID section.

Tip: To ensure that the DDM server, which opens the DRDA port, will autostart when the STRTCP command is issued during system startup:

1. Type GO TCPADM at the AS/400 command line.
2. Select Option 2 (Configure TCP/IP Applications)
3. Select Option 5 (Change DDM TCP/IP Attributes)
4. Change the Autostart server prompt to yes.

User Profile CCSID

The iSeries is shipped with the QCCSID system value set to 65535. In a DRDA environment, this is not desirable, as it does not allow for any translation to occur. Therefore, change the user profile of the user (or users) who will be using DB2 Table Editor to have the CCSID for US English which is 37. To change the CCSID, using US English as an example, from the AS/400 command line, type:

```
CHGUSRPRF USRPRF <USERID> CCSID(37)
```

Setting Server Parameters

When using the DB2 Table Editor Console component, you are always editing a particular server definition file. A server definition file (or SDF) contains all of the technical information needed by DB2 Table Editor to access any number of database servers.

To Set Server Parameters:

1. Start the DB2 Table Editor Console component.
2. Click the New button. The Server Parameters window opens.
3. In the Server name field, type the name of the server that you are defining. Each database server that you access with DB2 Table Editor must be defined and given a server name. There are no restrictions on what this name can be—it is intended to be a descriptive label for the server, used only in DB2 Table Editor.
4. In the RDB Name field, type the name of the relational database to which you want to connect using DB2 Table Editor. The RDB (relational database) name is taken from the iSeries Relational Database Directory Entry, which can be found by using the Display Relational Database Directory Entry command from the iSeries command line:
DSPRDBDIRE
5. Select the type of network connection that applies to your environment.
 - a. If you selected Connect using TCP/IP:
 - 1) In the Host Name field, type a host name for the database that you specified in the RDB name field. This can be either the Host Name.Domain Name or the server's TCP/IP address.
 - 2) In the Port Number field, Type a port number for the database server to use. The default port for DB2 to listen for DRDA TCP/IP conversations on the AS/400 is 446.

Note: There is a possibility that the default port (446) will already be assigned, but not active, or that the port has been changed to

another value. If either of these is the case, you will receive the WSAECONNREFUSED error message when DB2 Table Editor attempts to connect to the server.

- b. If you selected Connect using CPI-C, type a symbolic destination name in the Symbolic destination name field.
 - c. If you selected Connect using DB2 UDB for Windows CLI, select a database alias from the Database Alias list.
6. Check the Enable load balancing check box if you want load balancing to be enabled.
 7. Click Test Connection to test the connection to the server.
 8. Click OK.

Enabling DB2 Table Editor to list iSeries Physical and Logical files

The SQL statements that create the DB2 Table Editor database objects by default only allow object types of T (Tables) and V (Views) to be displayed in a list. Any AS/400 file can be accessed using DB2 Table Editor. However, in order for all file types, including Physical and Logical files, to be listed you must modify the SQL script as described in the following steps.

To enable DB2 Table Editor to list iSeries Physical and Logical Files:

1. From the Create Objects window, scroll down to the following statement:

```
CREATE VIEW RDBI.TABLE_VIEW2
```
2. In the “CREATE” statement, modify the “WHERE” clause to add ‘P’ for Physical files and ‘L’ for Logical files as follows:

```
WHERE A.TABLE_TYPE IN ('T','V','P','L');
```
3. Click OK. Once the packages have been successfully created, the Bind Packages window will open.
4. Click Next. The Permissions window opens.
5. In the User IDs box, type the user IDs to which you want to grant permission to run the packages in DB2 Table Editor.

Note: In order to run DB2 TableEditor, a user must have permission to run the user packages in order to use DB2 Table Editor. DB2 Table Editor creates the collection containing these packages on the iSeries with authority PUBLIC * EXCLUDE. You must grant appropriate authority for those users to that Collection/Library on the iSeries.

6. Click Grant. Permissions are granted on the server to the user IDs that you specified.
7. Click Next. The Check Table Authorizations window opens.
8. Select either check or do not check for unassigned tables.

Note: Tables with no permissions are not visible to DB2 TableEditor and do not appear in table lists. Once they are located, you can assign permissions to these tables, making them visible in lists.

Click Next. If you checked the Check for unassigned tables check box and unassigned tables are found, the Grant Table Authorization window opens.

9. From the Grant Table Authorization window, grant the necessary table authorizations, if necessary.
10. Click Finish.

Chapter 4. Installing DB2 Table Editor's Graphical User Interface

The following instructions will tell you how to install all of the components of DB2 Table Editor including:

- The DB2 Table Editor Console component
- The DB2 Table Editor component
- The DB2 Table Editor Java player component
- The DB2 Table Editor Developer component

To install DB2 Table Editor:

1. Insert the DB2 Table Editor CD-ROM into your CD-ROM drive.
2. Browse to the *Disk1* directory and double click *setup.exe*. The DB2 Table Editor install starts.
3. Follow the steps in the DB2 Table Editor Install wizard. For specific information on items in the install wizard, press the Help button to access the online help.
4. When the installation is complete, start the DB2 Table Editor Console component and create a server definition file for each server that you want to access using DB2 Table Editor. For more information on creating a server definition file, see "Creating a server" on page 27.

Note: Server definition files must be set up for the servers that you want to access using DB2 Table Editor before you can access these servers using the DB2 Table Editor Developer component, DB2 Table Editor component, or Java Player component. Use the DB2 Table Editor Console component to setup server definition files.

Advanced installation

After you complete the initial installation of DB2 Table Editor, you are ready to install it on other machines. To make this process more efficient, DB2 Table Editor is able to perform a variety of server and unattended installations. You can pre-define user options by modifying the server definition file, or the CPIC Provider DLL.

Configuring for Unattended Installation

You can setup an unattended installation of DB2 Table Editor with predefined user options. An unattended installation allows you to select the installation options for your DB2 Table Editor users before beginning the installation process. This will prevent you from needing to install and configure each instance of DB2 Table Editor individually.

To setup an unattended installation of DB2 Table Editor:

1. Copy the disk image directories used to install DB2 Table Editor from the distribution CD-ROM to your local hard drive.
2. Define the user options for DB2 Table Editor. There are two common options that you can predefine for your users:
 - Database Servers - you can define the database servers that users will access when they are using DB2 Table Editor. For more information on Database Servers, see "Creating a server" on page 27.

- SNA product - you can specify which SNA Product you are using. If you are using a CLI or TCP/IP connection, you do not need to define this setting. For more information on SNA networks, see "Using SNA networks" on page 27.

To define the above user options:

- Open the file named *DB2Forms.reg*, located in the *Disk1* directory in the DB2 Table Editor installation directories that you copied to your hard drive in step one.
- Navigate to the following section of the file:
[HKEY_CURRENT_USER\Software\IBM\RDBI\Options].
- Specify values for the server definition file and the CPICDLL (which is used to specify the type of SNA product that you are using) options in this file. Ensure that all back slashes (\) are doubled. For example, specify *c:\dir* (not *c:\dir*) and that both the option name and value are enclosed in double quotes.

Example:

```
"ServerDefinitionsFile"="f:\windows\sdf.ini"
"CPICDLL"="c:\windows\system\wincpic.dll"
```

- Save options file on Disk1.** After you edit and save the appropriate predefinition file, copy it to *Disk1* of the DB2 Table Editor installation diskettes.
- Open the file named *setup.ini*, located in the *Disk1* directory in the DB2 Table Editor installation directories that you copied to your hard drive in step one.
 - Edit *setup.ini* as to set the configuration options as desired. The variables that can be set are as follows:

Table 4. Variables

[Options]	Possible Settings	Effect
AutoInstall=	0,1	Specify 1 to perform an unattended installation. All other settings in <i>setup.ini</i> are ignored if this parameter is set to 0.
FileServerInstall=	0,1	Specifies whether DB2 Table Editor is already installed on a file server. If the setting is 0, all DB2 Table Editor files are installed to the directory specified in the InstallPath variable. If the setting is 1, DB2 Table Editor must be previously installed into the directory specified in the InstallPath variable.

Table 4. Variables (continued)

[Options]	Possible Settings	Effect
SetupType=	0,1,2	<p>Specifies the type of installation to perform.</p> <ul style="list-style-type: none"> • 0 indicates a typical installation • 1 indicates a compact installation • 2 indicates a custom installation. If you select 2, you must indicate which components to install. For more information on components, see Table 5. <p>Note: Option 2, the custom installation, is strongly recommended. Most users do not require the Console and Developer components, and the custom installation lets the system administrator give the user the most efficient installation possible.</p>
InstallPath=	<directory>	<p>Specifies the directory that will receive the DB2 Table Editor installation (if FileServerInstall=0) or the file server directory that already contains the installation (if FileServerInstall=1).</p>
ProgramGroup=	<program group name>	<p>Specifies the name of the program group that will be created (under Program Manager) or the program folder that will be created (for the Windows 95, Windows NT 4.0 and Windows 2000 Start menu).</p>

Table 5. DB2 Table Editor Components

[Components]	Settings	Effect
Admin=	0,1	<p>Specifies whether to install the DB2 Table Editor Console component program files. If FileServerInstall=1, files are not copied to the local machine, but program group icons are still created. Always set this option to 0 for user installations.</p>

Table 5. DB2 Table Editor Components (continued)

[Components]	Settings	Effect
Developer=	0,1	Specifies whether to install the main DB2 Table Editor Developer component program files. If FileServerInstall=1, files are not copied to the local machine, but program group icons are still created.

For Example:

```
[Options]
AutoInstall=1
FileServerInstall=0
SetupType=0
InstallPath=C:\Program Files\DB2 Table Editor
ProgramGroup=DB2 Table Editor
```

This sample *setup.ini* file specifies an unattended installation. This sample file would produce a typical installation with files copied to the *C:\Program Files\DB2 Table Editor* directory, and a program group or program folder created named *DB2 Table Editor*.

5. Save *setup.ini* to the *Disk1* in the DB2 Table Editor installation directories that you copied to your hard drive in step one.
6. Run the installation from the source diskettes or server. The installation proceeds automatically.

File server installation

DB2 Table Editor can be installed to a file server. This method of installation allows users to maintain only minimal files on their client machines. The main DB2 Table Editor files reside on the file server. The files that are installed on the client are listed below.

Table 6. Client Installed Files

File name	Description	Version
CTL3D32.DLL		2.31.000 (Windows NT 3.51 only)
MFC42.DLL	MFC	4.21.7303
MSVCRT.DLL	C runtime	5.00.7303

To achieve a serve installation, you must first install DB2 Table Editor on a server, then on the client machines.

To install DB2 Table Editor to a file server:

1. Install DB2 Table Editor to the server from the administrator computer Perform a standard DB2 Table Editor installation at the administrative computer. When prompted for the installation path specify a path that resides on the file server rather than the administrative computer. For information on standard installation, see "To install DB2 Table Editor:" on page 15.
2. Start the DB2 Table Editor Console component.
3. Use the DB2 Table Editor Console component to configure DB2 Table Editor at the file server. For more information on the DB2 Table Editor Console component, see Chapter 7, "Using DB2 Table Editor Console" on page 27.

4. The server installation is complete. Proceed to the client installation.

To install DB2 Table Editor on a client machine

1. Copy the disk image directories used to install DB2 Table Editor from the distribution CD-ROM to your local hard drive.
2. Configure DB2 Table Editor for unattended installation and specify the client installation option. For more information on configuring for an unattended installation, see "To setup an unattended installation of DB2 Table Editor:" on page 15. A sample Setup.ini file configured for unattended client installation would be as follows:

```
[Options]
AutoInstall=1
FileServerInstall=1
SetupType=2
InstallPath=f:\DB2Forms
ProgramGroup=DB2Forms
[Components]
Admin=0
Developer=1
```

This *setup.ini* file specifies an unattended installation, where files have already been installed on a file server, in the *f:\DB2Forms* directory. A custom installation will be performed, installing icons for the main DB2 Table Editor Developer component files, but not the DB2 Table Editor Console component program files. A program group or program folder named *DB2 Table Editor* is created, with icons referring to the *f:\DB2Forms* directory.

3. Run the DB2 Table Editor installation from the source diskettes or server. The installation proceeds automatically.

Chapter 5. DB2 Table Editor Concepts

The following are concepts that will help you understand DB2 Table Editor and how it functions.

Attributes

An attribute in DB2 Table Editor Developer defines the data values and behaviors that are associated with a form or control. Attributes are used to define which data is linked to a form and its controls, when and how the data is presented, as well as the function of each control in the form.

For example, form attributes can define which database tables are linked to a form, and how multiple tables are joined. Similarly, control attributes can define which data values are displayed in a list box, and how often those values are refreshed.

Controls

A control in DB2 Table Editor Developer is an individual component of a form. Controls allow users to view and edit data. Examples of controls include edit boxes, list boxes, and buttons.

Data Data is the smallest unit in a database, but the most important. Examples of data might be the names, addresses, phone numbers, and Social Security Numbers for all of the employees of a company. All this data can be stored in a table.

Database

A database is a software application that stores sets of electronically organized information on a computer. It contains a collection of tables, which contain data. DB2 is a database product built by IBM.

Forms

A form is a vehicle through which you view and change table data. In DB2 Table Editor, forms play different roles depending on the component of DB2 Table Editor that you are using.

- In DB2 Table Editor Developer, you create forms for use in the DB2 Table Editor component and the Java Player component. These forms can allow the user to access, search, and edit tables in DB2 databases table to which the form is linked.
- In DB2 Table Editor component and the Java Player component, a form is the object that you work with to view and change table data. It is similar to a paper form, in that it contains blank spaces, or fields, for you to fill in. In addition, it allows you to fill in some of the fields and retrieve information, or enter new information and update the database.

Note: The forms are built by a developer in DB2 Table Editor Developer. These forms will be highly customized to your organization, your data, and your needs.

Full-screen editor

A full-screen editor is a special form that presents data from the database as a grid or in a spreadsheet format. This type of form allows edits directly in the grid itself and includes the ability to insert and delete rows, sort columns, and find and replace values.

Join A join is a connection between two tables, using a field common to both tables. For example, there could be a table that containing personal

information for all the employees of a company (employee numbers, names, addresses, and Social Security numbers for each employee). In the same database there could be another table containing employee numbers and department numbers. In order to avoid repeating the employee personal information in the table containing department numbers, there is a join from the employee number in the personal information table, to the employee number in the department table; this allows the information to be linked, but not repeated.

Primary Table

The primary table is the table that is designated by the form developer as the updateable table. For each form there is only one primary table, but there can be a number of secondary tables, which are usually related to the primary table by join conditions. Insert, Update and Delete buttons on a form operate only on the primary table and perform the appropriate database modifications for the action (insert, update and delete) using the current form values for all controls that are bound to a primary table column.

Properties

A property in DB2 Table Editor Developer defines the appearance of a form or its controls. The developer uses properties to define the layout of the form, as well as the physical characteristics of controls.

Examples of form properties are background color and form dimensions. Examples of control properties are text font, color and control dimensions.

Tables

A table in DB2 Table Editor refers to a table that is contained in a DB2 database. A DB2 database contains data that is organized into smaller collections, or tables. Each form created with DB2 Table Editor Developer is linked to one or more tables, providing the data source for the form. DB2 Table Editor component and Java Player component are both used to access data stored in tables.

Validation rules

A validation rule in DB2 Table Editor Developer defines allowable values for the contents of a control. The developer uses validation rules to establish limits which govern what value a user can enter into a control, as well as the format of the value entered.

For example, a validation rule can require a user to enter a valid date format in a control. Validation rules can work individually or in groups.

Part 2. DB2 Table Editor Console

DB2 Table Editor Console is the component of DB2 Table Editor that is used to connect to DB2 Databases. Once you have connected to a DB2 Database, you can access that database using DB2 Table Editor Java Player or DB2 Table Editor Windows User.

Chapter 6. Overview of DB2 Table Editor Console

This chapter explains the basic concepts related to administering DB2 Table Editor and gives instructions for getting started. For details on specific tasks, use the online help. There are four basic tasks that you perform with DB2 Table Editor Console:

- Define and configure the database servers that are accessed by DB2 Table Editor Java Player and DB2 Table Editor Windows User
- Create database tables, bind database packages, and grant authority on user packages
- Set up governing
- Create sample database tables

For more information on the tasks in the list above, see Chapter 6 or the DB2 Table Editor online help.

Servers

Each database server that you or your users access with DB2 Table Editor must be defined with DB2 Table Editor Console. When you define a database, you give the database server a server name. There are no restrictions on what this name can be, it is intended to be a descriptive label for the server, used only in DB2 Table Editor. This name is all that a user of the DB2 Table Editor component or the Java Player component needs to know in order to access that server. DB2 Table Editor Console is used to define each server, giving it a name and also specifying the technical information that DB2 Table Editor needs to access it. This is similar to defining a data source in ODBC. The parameters that you must specify include:

- The RDB name for the server (also known as the location name in DB2 for OS/390 or MVS terminology, or the database name in DB2 for Multiplatforms or DB2 Common Server terminology), or the database alias if connecting through CLI.
- The appropriate network connection information if connecting via DRDA--either the CPI-C symbolic destination name or TCP/IP host name and port number.

Maintaining packages

To run distributed SQL at any DB2 database, you must bind a package at the database that contains the SQL that you want to run (excluding dynamic SQL). You use DB2 Table Editor Console to choose the collection name and options for the packages that it requires, and to automatically bind the packages at a server. The DB2 Table Editor packages refer to the set of DB2 tables that DB2 Table Editor uses, which might not exist at a server. If so, you must create these tables before binding the packages. DB2 Table Editor Console can automatically determine which tables need to be created, and it allows you to automatically create them. Finally, after you bind the DB2 Table Editor packages, you must grant authority to your end users to run the user packages. Again, DB2 Table Editor Console can automatically grant or revoke this authority to the users that you specify.

Governing

DB2 Table Editor incorporates a comprehensive resource governor that restricts the actions that a user can perform in DB2 Table Editor and places limits on the resources that a user can consume. This governing feature allows you to provide distributed access to DB2 to your users with the confidence that this will not have a negative impact on your overall database or network performance. Using DB2 Table Editor Console, you can define sets of limits and restrictions, which are called resource limits groups. You can then assign users to a resource limits group, according to the governing that you want performed for those users.

Sample Tables

DB2 Table Editor provides nine sample tables that you can use while learning DB2 Table Editor before you begin working with your own tables. The sample tables are used throughout the DB2 Table Editor Getting Started Guides and online help as examples. They contain information about a fictitious electrical parts company.

Table 7. Sample Tables

Sample table name	Contains information about
DBE.APPLICANT	The prospective employees of the company.
DBE.INTERVIEW	The interview schedule for prospective employees.
DBE.ORG	Organization of the company by department (within division).
DBE.PARTS	Materials supplied to the company.
DBE.PRODUCTS	Products produced by the company.
DBE.PROJECT	Company projects.
DBE.SALES	Sales information for the company.
DBE.STAFF	The employees of the company.
DBE.SUPPLIER	Other companies who supply materials to the company.

Chapter 7. Using DB2 Table Editor Console

This chapter will walk you through the major steps involved in configuring DB2 Table Editor for your organization. These are the four steps that you must perform in order to configure the application:

- Creating a server
- Testing the connection to the database server
- Binding packages
- Setting up governing

Starting DB2 Table Editor Console

When you install DB2 Table Editor, you can also install DB2 Table Editor Console. If you install DB2 Table Editor Console, an icon to start DB2 Table Editor Console is added to the Windows 9x, Windows 2000, and Windows NT 4.0 Start Menu. You can use this icon to start Console, or you can run it directly by starting the program DBADMIN.EXE. If this program file does not exist, you can reinstall DB2 Table Editor to copy it from the installation disks. Be sure to specify the Complete install option, or specify Custom and then choose to install DB2 Table Editor Console. Using DB2 Table Editor Console is an administrator task. An end user should not have a need to run DB2 Table Editor Console. However, there is no security risk if an end user runs DB2 Table Editor Console. These users are restricted in what they can do by existing database and file-sharing security mechanisms.

Using SNA networks

If you are connecting to any database servers using an SNA network, specify the SNA software that you are using before proceeding in DB2 Table Editor Console. From the main window in DB2 Table Editor Console, select Edit --> Options. On the Options dialog box in the CPI-C Options group, specify the name of the DLL that your SNA software provides for CPI-C applications.

Configuring DB2 Table Editor

Creating a server

When you are using DB2 Table Editor Console, you are always editing a particular server definition file. A server definition file (or SDF) contains all of the technical information needed by DB2 Table Editor to access any number of database servers.

There are two basic ways that you can use server definition files. You can allow each user to have their own SDF, or you can create a single SDF that is shared by multiple users over a file-sharing network.

The advantage of the "sharing" approach is that it centralizes the administration of the SDF--you only need to create and maintain a single file, and your users need only to point to that file when they run DB2 Table Editor. With either approach, you specify the SDF to use on the Options window in DB2 Table Editor. To access the options window, select View --> Options.

From within DB2 Table Editor Console, you can create a new SDF for a user or group of users by selecting File --> New command or File --> Save As. To open and work with a different SDF, select File --> Open.

To Create a server:

1. Start DB2 Table Editor Console
2. Click the New button to define a new server. The Server Parameters dialog box opens.
3. In the Server Name field, type the name for your server. This can be any name that you choose. It is the name by which users refer to this server throughout DB2 Table Editor and DB2 Table Editor Console. It does not necessarily need to be the name the server is known by on the network.

Note: If you are using the Java Player, the name of the server must match the database alias name specified in the Client Configuration Assistant (CCA). If these do not match, when you create a form in the Java player, the name of the server from the server definition file will be inserted and the two will not match. This will result in an error.

4. In the RDB name field, type the name of the database on the server. This is the name of the database. In DB2 for z/OS and OS/390 or MVS terminology, this is also known as the location name. If you are unsure of the value to enter here, there is an easy way to determine the correct value. Using a tool other than DB2 Table Editor, run the following query at the server (you can use any table if DBE.STAFF does not exist):

```
SELECT CURRENT SERVER FROM DBE.STAFF
```

5. In the Network Connection box, specify the type of connection that you are using to connect to your database. Select one of the following connection types:
 - Select Connect Using TCP/IP to specify that you will be connecting to the database using the TCP/IP protocol. Type the host name in the Host name field. If you enter a TCP domain name for the host name, DB2 Table Editor resolves that name to an address using the gethostbyname socket call. Alternatively, you can directly specify the host address in dotted decimal notation (for example, "111.22.33.44"). Type the port number in the Port number field.

Note: Not all DB2 servers support TCP/IP connections. Refer to the documentation for the server you are configuring for more information.

- Select Connect Using CPI-C to specify that you will be using the CPI-C protocol to connect to the database. In the Symbolic destination name field, enter the name for the symbolic destination for the database. If you are connecting to DB2 via CPI-C: DB2 Table Editor requires a product that implements CPI-C in a Windows environment. Before installing DB2 Table Editor, you must configure this software and defined a CPI-C symbolic destination name for the server. This is a name defined in your SNA software that refers to a set of parameters (known as a CPI-C side information record) for establishing a connection to a partner application (DB2) over the network. For more information, see your SNA software's documentation.
 - Select Connect Using DB2 UDB for Windows CLI to specify that you will be connecting through DB2 for Multiplatforms on Windows CLI. Type the alias for the database to which you will be connecting in the Database alias field. The database alias is a name that refers to a database defined within your DB2 for Multiplatforms client software. If you are connecting to DB2 via CLI you must have DB2 for Multiplatforms version 5.2 or greater installed locally. Before installing DB2 Table Editor you must configure the client software and define a database alias name for the server within the DB2 for Multiplatforms client
6. Click OK The server definition file (SDF) is created.

Note: The default location for the server definition file on Windows NT and Windows 2000 is the WINNT directory.

Testing the connection to the database server

To ensure that DB2 Table Editor can establish a connection to the database server test the connection to the server.

To test the connection to the database server:

1. From the DB2 Table Editor Console window, click the Edit button. The Server Parameter dialog box opens.
2. Click Test Connection to test the connection to the database server. If there are problems with your network configuration, DB2 Table Editor Console displays an error message dialog box.

Note: If you are connecting to DB2 via an SNA network, the SNA link and the LU 6.2 session between the DB2 Table Editor LU and the DB2 LU must be active to establish the connection.

There are a few errors that can occur in DB2 Table Editor Console when establishing a connection to the server; a problem at this point indicates a problem with the network configuration rather than DB2 Table Editor Console. Common errors include:

- Failure to activate the SNA software or start the SNA node
 - Failure to activate the SNA link
 - Failure to properly configure an LU 6.2 session between the DB2 Table Editor LU and the DB2 LU
3. If this is the first time you have edited the server definition file, the Set User Information window will open. In the Set User Information window do the following:
 - a. In the User ID field, type the user ID for the owner of the package.
 - b. In the Password field, type the password for the owner of the package.
 - c. *Optional:* If the database server to which you are connecting uses accounts, type the account name for the owner of the package in the Account field.
 - d. Click OK.

Binding the DB2 Table Editor packages

After you verify that DB2 Table Editor can establish a connection to the database server, you are ready to bind the DB2 Table Editor packages.

To bind the DB2 Table Editor packages:

1. Select the server in the DB2 Table Editor Console main window, then click Edit. The Server Parameters window opens.
2. From the Server Parameters window, click the Manage button. A pop-up menu opens.
3. From the Manage button pop-up menu, click Packages. The Package Properties window opens.
4. In the Collection ID field, type the collection name. The collection name is used to identify the packages that are created for use by DB2 Table Editor Console. Four packages are bound in this collection:
 - DBEP1, which is used by DB2 Table Editor
 - RAARDBIL, which is used by DB2 Table Editor

- RAARDBI2, which is used by DB2 Table Editor
- RAARDBIA, which is used by DB2 Table Editor Console, and contains the SQL required for administrator functions.

Note: If you have not bound the packages for the current version of the DB2 TableEditor, you will need to choose the Bind the packages option to bind the latest versions.

5. In the Owner ID field enter the ID for the owner of the package.
6. In the Decimal Delimiter box, specify whether you want to use a comma or a period as the decimal delimiter for this package.
7. In the String Delimiter box, specify whether you want to use an apostrophe or a quote as the string delimiter for this package.
8. Click Next. The Set User Information window opens.

Note: If you have already tested the connection to the server, the Set User Information window will be skipped and the Check Objects window will open. In this case, go directly to Step 13. For more information on testing the connection to the server see “Testing the connection to the database server” on page 29.

9. In the User ID field, type the user ID for the owner of the package.
10. In the Password field, type the password for the owner of the package.
11. *Optional:* If the database server to which you are connecting uses accounts, type the account name for the owner of the package in the Account field.
12. Click **OK**. The Check Objects window opens.
13. From the Check Objects window, specify one of the following:
 - Select Assume that none of the objects already exist to specify that none of the database objects that are required already exist in the database to which you are connecting.
 - Select Automatically check which objects already exist to have DB2 Table Editor check to see if any of the database objects that are required already exist in the database to which you are connecting.

Note: In order to check for the existence of the required objects, the SELECT authority is required on the following tables, depending on the type of database server:

DB2 for Multiplatforms for z/OS and OS/390, DB2 for z/OS and OS/390, DB2 for MVS	DB2 for VM & VSE, SQL/DS
- SYSIBM.SYSDATABASE	- SYSTEM.SYSDBSPACES
- SYSIBM.SYSTABLESPACE	- SYSTEM.SYSCATALOG
- SYSIBM.SYSTABLES	- SYSTEM.SYSINDEXES
- SYSIBM.SYSINDEXES	- SYSTEM.SYSCOLUMNS
- SYSIBM.SYSCOLUMNS	

DB2 for Multiplatforms for AS/400	DB2 for Multiplatforms, DB2 Common Server, DB2 Parallel Edition, DB2 DataJoiner
<ul style="list-style-type: none"> - QSYS2.SYSTABLES - QSYS2.SYSINDEXES - QSYS2.SYSCOLUMNS 	<ul style="list-style-type: none"> - SYSCAT.TABLESPACES (except DB2 Parallel Edition and DB2 DataJoiner, version 1) - SYSCAT.TABLES - SYSCAT.INDEXES - SYSCAT.COLUMNS

- Select Assume that all of the objects already exist to specify that all of the database objects that are required already exist in the database to which you are connecting.

Note: DB2 TableEditor Requires that some database objects such as tables and indexes for example, exist in DB2 for its use. If you had a previous version of DB2 Table Editor installed that connected to this database, it is possible that some of these objects may already exist.

14. Click Next. If you selected Assume that none of the objects already exist, or if you selected Automatically check which objects already exist and some of the database objects need to be created, the Create Objects window will open. If you selected Automatically check which objects already exist and all of the database objects already exist, or you selected Assume that all of the objects already exist, the Bind Packages window opens.
15. *Optional:* From the Create Objects window, verify the syntax of the statements that are in the window, modify any of the statements if necessary and click Next. The Bind Packages window opens.
16. From the Bind Packages window, select the Bind the packages radio button.

Note: Binds the packages at the database server. You must have the BINDADD authority and the CREATE IN authority for the specified collection ID at the database server in order to bind the DB2 Table Editor packages. In addition, you must have the following authorities:

Table	Authorities
DBE.OBJECT_VIEW	Select, Insert, Update, Delete
DBE.JOBJECT_VIEW	Select, Insert, Update, Delete
RDBI.PROFILE_VIEW	Select, Insert, Update
RDBI.RESERVED	Select
RDBI.RESOURCE_VIEW	Select, Insert, Update, Delete
RDBI.TABLE_VIEW2	Select
RDBI.USER_ADMIN_VIEW	Select
RDBI.USER_AUTHID_VIEW	Select

SQL errors with an SQL code of -204 indicate that a required table does not exist. Click Create Objects... to create it. An SQL code of -551 or -552 indicates that the authorization ID on the bind (either your user ID or the specified owner ID for the bind) lacks at least one authority that is required to bind the packages

17. *Optional:* To replace any existing packages with the new packages that you are creating, check the Replace existing packages (if any) check box.
18. *Optional:* To keep existing authorizations on the packages that you are creating, check the Keep existing authorizations on packages check box.

19. Click Next. If you selected the Bind the packages radio button, the packages will be bound. The Permissions window opens.
20. From the Permissions window, type the user IDs for the users to whom you want to grant the authority to run the packages that you are creating. You can type multiple User IDs in the box by separating them with a comma or a space. Specifying a user id of "PUBLIC" will grant permission to all user IDs.

Note: In the iSeries environment, DB2 TableEditor creates the SB2 Table Editor packages on the iSeries with authority PUBLIC * EXCLUDE. You must grant appropriate authority for users to that Collection/Library on the iSeries in order for them to use DB2 Table Editor on the iSeries.

21. Click Grant. This change in authority will be made at the server.
22. Click Next. The Check Table Authorizations window opens.
23. From the Table Authorizations window, specify whether you want DB2 Table Editor to check to see that all of the tables in the DB2 database to which you are connecting have at least one user with table authorization on them. Tables with no user authorizations on them do not always appear in table lists that are seen by the users. If you select Check for tables with no authorizations, DB2 Table Editor will check to see if there are any tables in the database that do not have any authorizations on them. If tables without authorizations are found the Grant Table Authorizations window will open and you will be able to specify the user ID to which you want to grant permissions for the tables.
24. *Optional:* From the Grant Table Authorizations window, From the Tables list, select that table to for which you want to grant authorities.
25. In the User ID field, type the user ID to which you want to assign select authority for the selected table.

Note: You must have the authority to assign SELECT permission to use this dialog box.

26. Click Grant. The Privilege is granted to the specified user.
27. Repeat steps 24 to 26 for all of the tables listed in the Tables list.

Setting up governing

DB2 Table Editor incorporates a resource governor that limits and controls the use of database and communications resources. You use DB2 Table Editor Console to set up the resource limits groups that manage, control, and restrict this resource usage.

The governing function of DB2 Table Editor is always active. If you do not explicitly set up resource limits, governing based on default limits still occurs.

This section describes how to use DB2 Table Editor Console to set up resource limits to effectively manage, control, and restrict resource usage:

Questions about governing

Here are a few of the most frequently-asked questions regarding DB2 Table Editor governing:

- **How do I set up DB2 Table Editor governing?**

To set up DB2 Table Editor governing, create a resource limits group and assign DB2 Table Editor users to that group. After a resource limits group is defined and a DB2 Table Editor user is assigned to it, that user's resource usage is limited as defined by that group.

- **What are the resources that I control?**

These resources are governed:

- Amount of time to wait for a response from a database server
- Amount of time a connection to a database server can remain unused
- Maximum number of rows returned from the database server for a query
- Maximum number of bytes returned from the database server for a query
- Maximum number of simultaneous connections to a database server
- Whether a database can be accessed from the DB2 Table Editor user interface
- The accounting string to pass to the database server
- Whether to fetch all rows immediately or only as-needed

• **What is a resource limits group?**

A resource limits group is a collection of limits and controls on the resources that are governed by DB2 Table Editor. You can control resource consumption by user, day of the week, and by time of day. For example, a resource limits group can contain one set of limits that is effect weekdays between 8:00 am and 6:00 p.m. and another that is in effect on weekends and off-hours.

• **Where are resource limits groups stored?**

To prevent users from circumventing limits that you establish, resource limits groups are securely stored in a database table at the database server. Specifically, resource limits groups are stored in the table named, *RDBI.RESOURCE_TABLE*. A view named *RDBI.RESOURCE_VIEW* must be defined on this table because DB2 Table Editor accesses that view, not the table. DB2 Table Editor Console is used to maintain resource limits groups. To use DB2 Table Editor Console to maintain resource limits groups, you must have the authorization to run the DB2 Table Editor Console package. This prevents unauthorized users from changing the limits that are established by the administrator.

• **What is the default resource limits group?**

Users who are not explicitly assigned to a resource limits group are governed by the limits that are defined in the default resource limits group.0 The system administrator is responsible for creating and maintaining the default resource limits group, which is named: *<Default>*.

• **How does DB2 Table Editor relate a resource limits group to a user?**

The relationship between a DB2 Table Editor user and a resource limits group is stored in a table at the database server, specifically, the table named *RDBI.PROFILES_VIEW*. DB2 Table Editor Console is used to maintain user and resource limits group relationships in this table.

• **How does DB2 Table Editor determine which profile to use?**

When DB2 Table Editor connects to a database server, you provide user information (user ID and password), which is validated by the database server. If the user information is valid, DB2 Table Editor determines which resource limits group to use by first locating the correct profile for the user. This is done by searching the *CREATOR*, *ENVIRONMENT*, and *TRANSLATION* columns in the *RDBI.PROFILES_VIEW* table. DB2 Table Editor searches profile values in this order:

CREATOR	ENVIRONMENT	TRANSLATION
User ID	WINDOWS	from current national language of DB2 Table Editor
User ID	NULL	from current national language of DB2 Table Editor

CREATOR	ENVIRONMENT	TRANSLATION
SYSTEM	WINDOWS	from current national language of DB2 Table Editor
SYSTEM	NULL	from current national language of DB2 Table Editor

Open and restricted enrollment

Every user must have a user profile. Under restricted enrollment, if a user profile with a matching creator does not exist, the user is denied access to the server. With open enrollment, if a user profile with a matching creator does not exist, DB2 Table Editor will look for a user profile with a creator that is equal to SYSTEM. Access to the server will be granted only if a profile is found in one of these ways. With open enrollment in effect, every user has access to the SYSTEM profile, giving every user potential access to the server.

You can create unique profiles for some users and allow other users to use the SYSTEM default profile; you can also delete the SYSTEM profile, thus preventing those who do not have unique profiles from accessing the server.

- **How does DB2 Table Editor determine which resource limits group is in effect?**

The *RESOURCE_GROUP* column in the *RDBI.PROFILES_VIEW* table contains the name of the resource limits group that is in effect for users identified by the profile entry. If this field is null or blank, DB2 Table Editor will assume a default value of the user ID.

DB2 Table Editor then searches the *RESOURCE_GROUP* and *RESOURCE_OPTION* columns in the *RDBI.RESOURCE_VIEW* view, looking for the following rows:

RESOURCE_GROUP	RESOURCE_OPTION
RESOURCE_GROUP from RDBI.PROFILES_VIEW	SHUTTLE_MAIN
RESOURCE_GROUP from RDBI.PROFILES_VIEW	SHUTTLE_SCHED1
RESOURCE_GROUP from RDBI.PROFILES_VIEW	SHUTTLE_SCHED2
RESOURCE_GROUP from RDBI.PROFILES_VIEW	SHUTTLE_SCHED3
RESOURCE_GROUP from RDBI.PROFILES_VIEW	SHUTTLE_SCHED4

If no matches are found, the user is governed by the default resource limits group. Otherwise, the user is covered based on the limits and controls found in these rows.

- **How does DB2 Table Editor determine which schedule is in effect?**

After the resource limits group is determined, DB2 Table Editor determines which schedule in the group is in effect.

A schedule is uniquely identified by a schedule number. In addition to specifying a unique number, you must also specify an effective day of the week and time of day range. The From- and To Time and the From- and To Day for the schedule define when the limits and controls are in effect.

If more than one schedule is defined to be in effect at the same time, the DB2 Table Editor governor will use the one with the lowest schedule number.

Note: All ranges are inclusive, so that if the From Time is 08:00, the schedule is in effect exactly at 08:00:00. If the To Time is 17:00, the schedule is in effect at 17:00:59 but not in effect at 17:01:00.

If the From- and To Times wrap around midnight (for example 5 PM through 8 AM), this creates, in effect, two separate blocks of time each day that the schedule becomes active. Consider the following as an example:

From Day

Monday

To Day

Friday

From Time

17:00 (5 PM)

To Time

08:00 (8 AM)

This schedule is in effect only on weekdays. For each weekday, this schedule is in effect for the hours 5 PM to midnight, and midnight to 8 AM. For example, this schedule is in effect each Monday morning from midnight to 8 AM as well as each Monday evening from 5 PM to midnight.

Enabling DB2 Table Editor governing capabilities

Governing is always active in DB2 Table Editor. If no explicit limits are in effect, governing is performed based on the limits in the default resource limits group.

There are three basic steps to establishing explicit governing or resource limits:

1. Create the resource limits group.
2. Create schedules in the resource limits group.
3. Assign users to the resource limits group.

Creating a resource limits group

1. From the DB2 Table Editor Console, select the server that you are currently working with and click the Edit button. The Server Parameters dialog box opens.
2. Click the Manage button. A pop up menu opens.
3. From the pop-up menu, select Resource Limits. The Resource Limits Group List dialog box opens, showing a list of all the resource groups defined at the server.
4. Select the resource limits groups in the list on which you want to base the new group and click the New button. The New Resource Limits Group List dialog box opens.
5. Type a name for the group in the Group Name field. There are no restrictions on the name that you enter.
6. *Optional:* Type any comments, up to 80 characters in length, that describe the new resource limits group.
7. *Optional:* If the Create this group using schedules from check box is enabled, the group that you selected as a model has schedules that you can copy into the new group. Check this check box to create the new group with copies of all of the schedules contained in the model group. Otherwise, the new group will contain no schedules.
8. Click OK to create the resource limits group.

Creating schedules in a new resource limit group

1. Select the resource limits group for which you want to create schedules in the Resource Limits Group List dialog box and click the Edit button. The Edit Resource Limits Group dialog box opens.
2. Click the New button. The Resource Limits Group Schedule dialog box (Main Tab) opens so you can create a new schedule. If you select a schedule in the Schedule List before you click this button, the selected schedule is used as a model for the new schedule.
3. Enter the required values on each of the eight tabs and click OK to create the new schedule.

Assigning users to the new resource limits group

1. Click the Assign button on the Resource Limits Group List dialog box. The Assign User Profiles dialog box opens.

Note: Type the first user ID you want to assign in the Show user profiles with "creator" matching field or matching pattern. If you want to work with multiple user IDs, and click the Refresh List button, DB2 Table Editor Console retrieves all the user profiles stored in the *RDBI.PROFILES_VIEW* table that match the value you entered and displays them in the Not Assigned or Assigned lists.

Tip: If the user ID that you want to assign does not have an entry in the *RDBI.PROFILES_VIEW* table, click the Create New button to create the new user profile.

2. Select the appropriate user IDs and use the Assign and Unassign buttons to move them to either list.
3. Click OK.

Sample tables

DB2 Table Editor provides nine sample tables that you can use while learning DB2 Table Editor before you begin working with your own tables. The sample tables are used throughout the documentation as examples. They contain information about a fictitious electrical parts manufacturing company. For a complete listing of the DB2 Table Editor sample tables, see "Sample Tables" on page 26

Creating sample tables

1. Select the server at which you want to create the sample tables and click the Edit button. The Server Parameters dialog box opens.
2. Click the Manage button. A pop-up menu opens.
3. From the pop-up menu, select Sample Tables. A dialog box opens, warning you that the sample tables will overwrite any previous versions of the sample tables that exist on the server.
4. Click Yes to create the sample tables. The sample tables are created.

Changing your password in DB2 Table Editor

You can change your password at the database server from DB2 Table Editor on some DB2 Platforms.

Note: Not all versions of DB2 Support password changing. For example, on the OS/390 or z/OS platform feature is currently supported only by DB2 for OS/390 Version 5 and later.

To Change your password

1. Select the server at which you want to create the sample tables and click the Edit button. The Server Parameters dialog box opens.
2. Click Set User Info. the Set User Information window opens.
3. Click Change. The **New password** and Confirm new password fields display.
4. Type your new password in the New password and Confirm new password fields and click OK. Your database server password is changed.

Configuring a Server that has bound packages associated with it

To configure a server that already has bound packages associated with it, you must follow slightly different steps than when you configure a server that does not have bound packages already associated with it.

To create a new entry in the SDF, you must either bind the packages with a different Collection ID, or add the information about the packages that you want to use to that SDF.

It is necessary to go through the steps below because information is not written to the SDF otherwise. If you do not go through the steps, DB2 Table Editor will have no Collection ID specified for this database server and it will attempt to use the default 'NULLID'.

To configure a server that already has packages associated with it:

1. Open the Package Properties window by selecting the server on which you want to work and Clicking **Edit**. The server Parameters window opens. From the Server Parameters window, select Manage --> Packages. The Package properties window opens.
2. From the Package Properties window, select a Collection ID that has been used to bind packages previously.
3. Check the Assume all objects already exist check box.
4. Click Next.
5. Check the Do not Bind the Packages check box.
6. Click Next. The Grant Permissions window opens.
7. Click Next. The Check Authorizations window opens.
8. Click Finish.

Part 3. DB2 Table Editor Developer

DB2 Table Editor Developer is a development environment in which you can build table editing applications.

Chapter 8. Overview of DB2 Table Editor Developer

Learn how you can use DB2 Table Editor Developer.

What is DB2 Table Editor Developer?

DB2 Table Editor Developer is a development environment in which you can build table editing applications. Using DB2 Table Editor Developer's drag-and-drop interface, you can build forms that provide a simple, graphical interface to IBM's DB2 database, without programming. The forms you design with DB2 Table Editor Developer allow your users to access, search, and edit data that resides on a linked DB2 database. Although DB2 Table Editor Developer is designed to interact with DB2, you can also use IBM's DB2 DataJoiner to connect your forms to virtually any other database application.

The forms you create using DB2 Table Editor Developer are platform-independent. When you build a form, DB2 Table Editor Developer automatically translates your work into XML. This means that your form will work with the DB2 Table Editor component and the Java Player component of DB2 Table Editor. In addition, you can use applet tags to place any form directly onto a web page, such as a corporate intranet site, granting platform-independent access to many users from one location.

What can I do with DB2 Table Editor Developer?

Using DB2 Table Editor Developer, you can build custom table-editing applications that provide users a simple, form-based interface to DB2 database tables. DB2 Table Editor Developer helps you create forms that perform query-by-example, table editing and many other tasks. Your forms can join tables, include validation rules, run SQL (including DB2-specific SQL), and access multiple DB2 platforms. With DB2 Table Editor Developer's intuitive, drag-and-drop development environment, you can build custom forms quickly, without programming. You can store your forms centrally on a DB2 server, for easy modification and distribution. Users can access forms over standard internet, intranet, or LAN connections.

Chapter 9. Working with DB2 Table Editor Developer

Before working with DB2 Table Editor Developer, you must define and configure the database servers to which you want to connect using DB2 Table Editor Console. For more information on this, see “Configuring DB2 Table Editor” on page 27

This chapter contains information about creating forms, adding controls to a form, and working with forms.

Creating new forms

There are three methods of creating new forms:

- Using the Forms wizard
- Manually
- Importing a panel from DBEdit

You can use DB2 Table Editor to create custom forms. You can also automatically create forms from tables with the DB2 Table Editor Form Wizard feature, or by importing panels from the mainframe application, DBEdit.

When creating a form you specify the **primary table** that the form will be used to access and modify. A primary table is the table that is designated by the form developer as the updateable table. For each form there is only one primary table, but there can be a number of secondary tables, which are usually related to the primary table by join conditions. Insert, Update and Delete buttons on a form operate only on the primary table and perform the appropriate database modifications for the action (insert, update and delete) using the current form values for all controls that are bound to a primary table column.

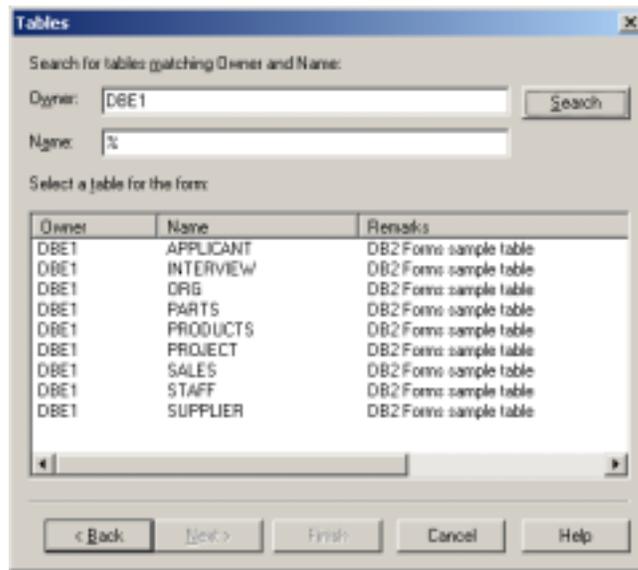
Creating a new form using the Forms wizard

You can create a form automatically using the Forms Wizard. Forms created in this way initially interact with one table.

To create a form using the Forms wizard:

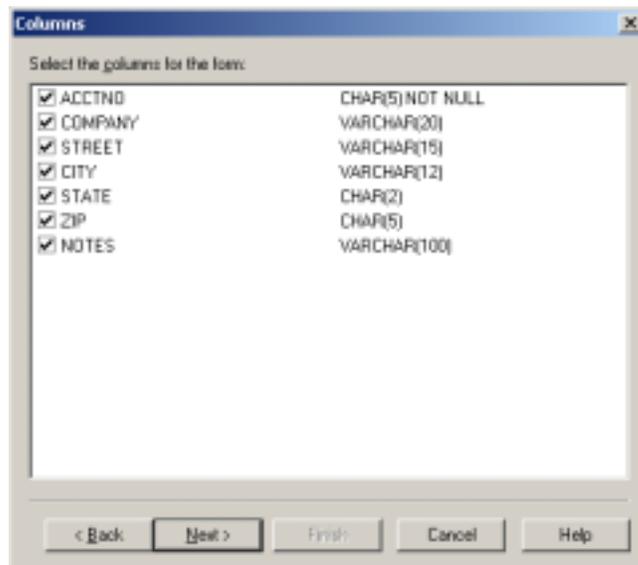
1. From the DB2 Table Editor Developer main window, select File --> Form Wizard. The Server dialog box opens.
2. Select the server that stores the database that you want the form to use, and click Next. The Tables window opens.
3. From the Tables window, in the Owner field, type the owner of the table that you want the form to use. This field is case sensitive. Use the % character as a wildcard character.
4. In the Name field, type the name of the table that you want the form to use. This field is case sensitive. Use the % character as a wildcard character.

- Click Search. A list of tables matching those criteria is returned in the lower portion of the Tables dialog box.



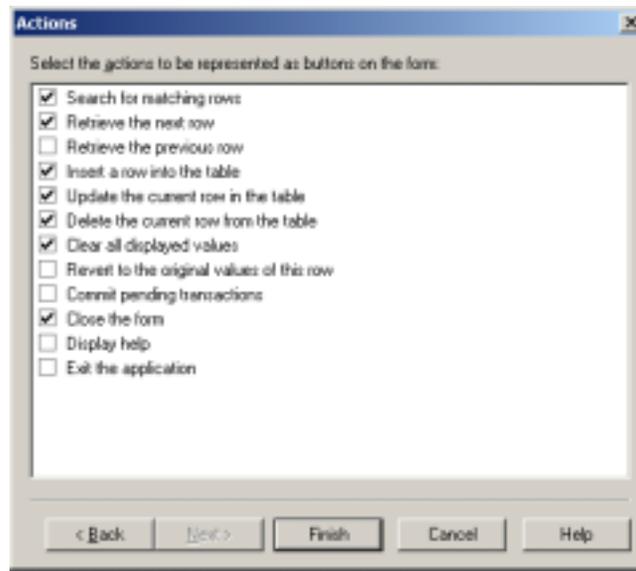
Note: You can choose from a list of tables stored at a database server. In the Tables dialog box, enter the special command character "%" in the Owner and Name fields (and in the Type Database Name, and Table Space Name field, which are available if you are accessing a z/OS or OS/390 system), and click Search. A list of all tables on the server is returned. You can also specify an owner, to list only that owner's tables.

- From the list in the Tables window, select the table that you want the form to access, and click Next. The Columns dialog box opens, displaying a list of all of the columns in the table.



- Ensure that all of the columns that you want to use on your form are checked.
- Click Next. The Actions dialog box opens, displaying a list of actions that the form can run.

Note: These actions are represented on the form as buttons.



9. Ensure that the actions that you want the form that you are creating to use, are checked, and click **Finish**. The new form opens in the main window. You can now save or run the file.

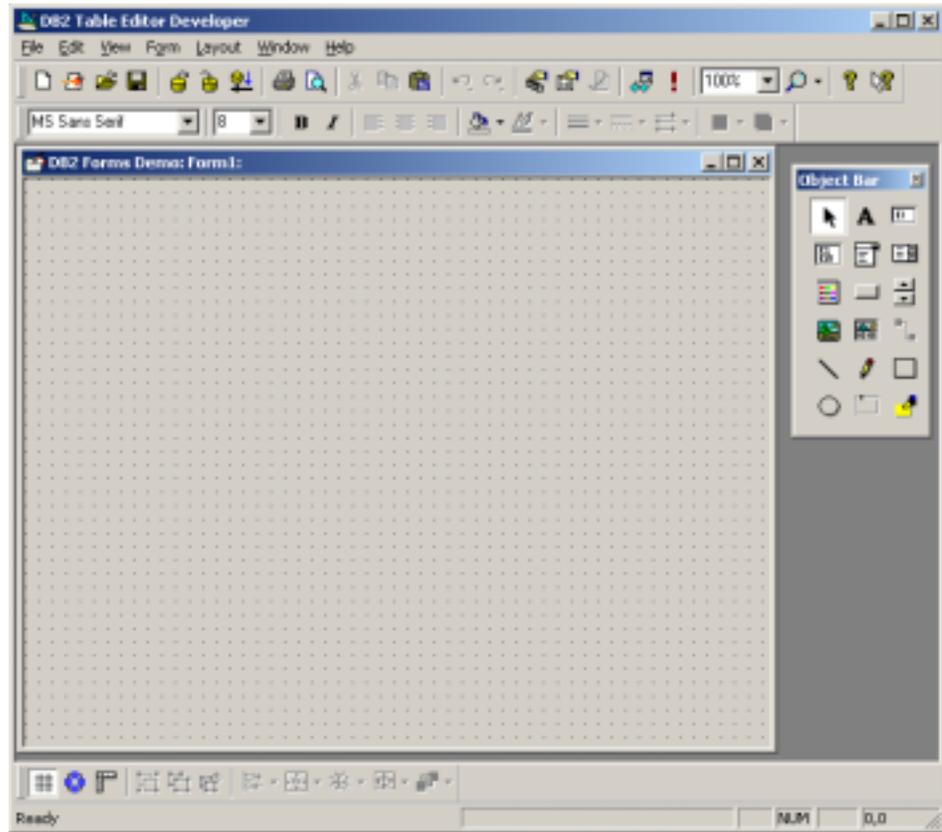
Note: Forms created using the Form Wizard are initially linked to one table in the database. To modify the form to reference more than one table see "Creating a Form Manually" below.

Creating a form manually

When creating a form manually (without using the Forms Wizard), you complete two major tasks: drawing the form, then attaching the form to a database, a table, and columns.

To create a form manually:

1. From the DB2 Forms Developer main window, select File --> New. A blank form opens.

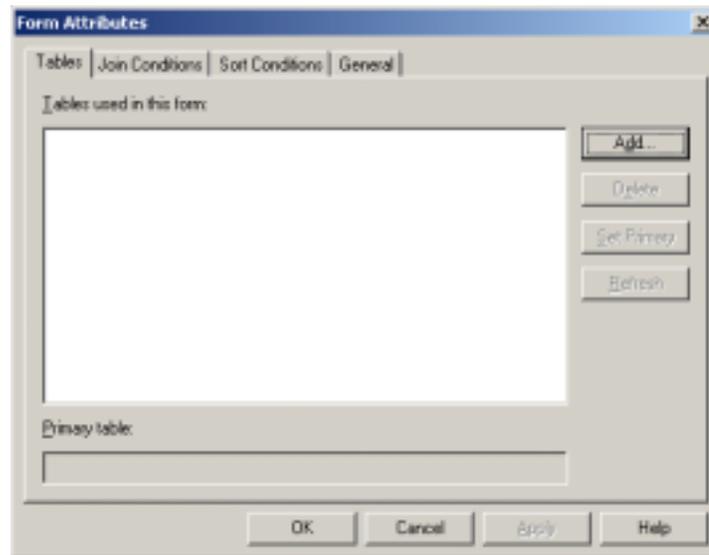


2. Add the necessary controls and text to the form. To add controls to the form, drag and drop the controls from the Object Bar window to the area where you want the control to be on the form. When you have finished adding controls and text to the form, save or run the form by selecting File --> Save As.
3. Select Form --> Set Server The Server window opens.
4. From the Server window, select the server that you want the form to access, and click **OK**. The name of the server appears in the title bar. The form uses the server that you selected.
5. Select Form --> Set User Information. The Set User Information dialog box opens.



6. In the User ID field, type your user ID for the server that you want to access.
7. In the Password field, type your password.
8. *Optional*: If your server uses accounts, specify your account in the Account field.

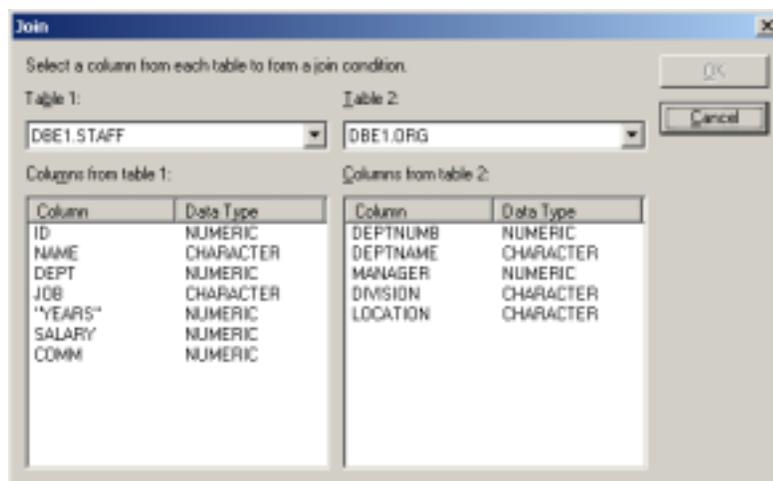
9. Click **OK**. You are connected to the database.
10. Ensure that no control has the focus (is selected) on your form. Select Form --> **Attributes**. The Form Attributes dialog box opens.



11. Click Add. The Add Table dialog box opens.
12. From the Add Table window, Enter the Table owner in the Owner field.
13. Enter the Table name in the Name field.

Tip: You can choose from a list of tables stored at the database server that you specified for your form. On the Add Table dialog box, leave the Owner and Name fields blank, and click List. A list of all forms on the server is returned. You can also specify an owner, and list only that owner's forms. To select the tables you want to use, choose each table and click Add. Click Close when you have finished selecting tables.

14. Repeat steps 12 and 13 to add as many tables as necessary.
15. Click OK. The Add Table window closes.
16. Optional: To join two tables:
 - a. Click the Join Conditions tab. The Join Conditions page opens.
 - b. Click Add. The Join window opens.

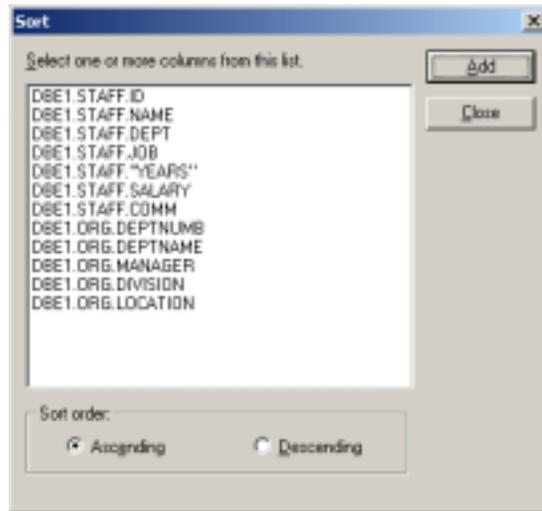


- c. From the two table drop-down lists, select the tables you want to join.

- d. From the columns lists, select the column from each table on which you want perform the join.

Note: You must choose columns with matching data types. Join columns with matching data. For example, join columns that both contain "Last_Name" or "Product_ID_Number" data.

- e. Click OK. The Join window closes.
17. Click the **Sort Conditions** tab. The Sort Conditions page opens.
18. Click **Add**. The Sort window opens.



19. Select each column you want to sort, and click the **Ascending** radio button or the **Descending** radio button to specify whether the data will be sorted in ascending or descending order.
20. Click **Select**.
21. When you have specified all of the necessary sort conditions, click **Close**. The Sort window closes and your form displays the data in the order that you specified.
22. Click **OK** to close the Form Attributes dialog box.
23. Save the form by selecting File --> Save.

Creating a new form by importing a panel from DBEDIT

You can import a panel that was created for the DBEDIT application.

Note: Once a form is imported to DB2 TableEditor, it can no longer be opened in DBEdit.

To import a file from DBEDIT:

1. From DB2 Table Editor Developer main window, select File --> Import from DBEDIT. The Import Panel from DBEDIT window opens.



2. From the Server list, select the server where the DBEDIT panel is stored.
3. In the Catalog Owner field, type the user ID for the catalog owner.
4. In the Panel Owner field, type the user ID for the panel owner.
5. In the Panel ID field, type the panel ID.
6. Click **OK**. the Import Panel from DBEDIT window closes and DB2 Table Editor Developer imports and opens the DBEDIT panel that you have imported.

Note: Once a form is imported to DB2 TableEditor, it can no longer be opened in DBEdit.

Working with forms

This section describes the basics of working with a DB2 Table Editor form. The sections in this chapter include:

- Opening a Form
- Saving a Form
- Previewing a Form
- Printing a Form
- Closing a Form
- Deleting a Form
- Setting General Form Attributes

Opening a form

You can open a DB2 Table Editor form that is saved on a local file or from a database server.

Opening a form saved locally

You can open a form previously saved to a location on your hard drive.

To open a form that is saved locally:

1. From the main window of DB2 Table Editor Developer, select File --> Open. The Windows Open dialog box opens.
2. Navigate to the form you want to open and click **Open**. The form opens in the DB2 Table Editor Developer main window.

Opening a form that is stored on a database server

You can open a form that is stored on a database server.

To open a form that is stored on a database server:

1. From the main window of DB2 Table Editor Developer, select File --> Open From Server. The Open From Server window opens.
2. From the Server list, select the server where the form is stored.
3. In the Owner field, type the user ID for the catalog owner.
4. In the Name field, type the name of the form that you want to open.

Tip: You can choose from a list of forms stored at a database server. On the Open From Server dialog box, leave the Owner and Name fields blank, and click List. A list of all forms on the server is returned. You can also specify an owner and list only that owner's forms.

5. Click OK. The form opens in the main window.

Setting general form attributes

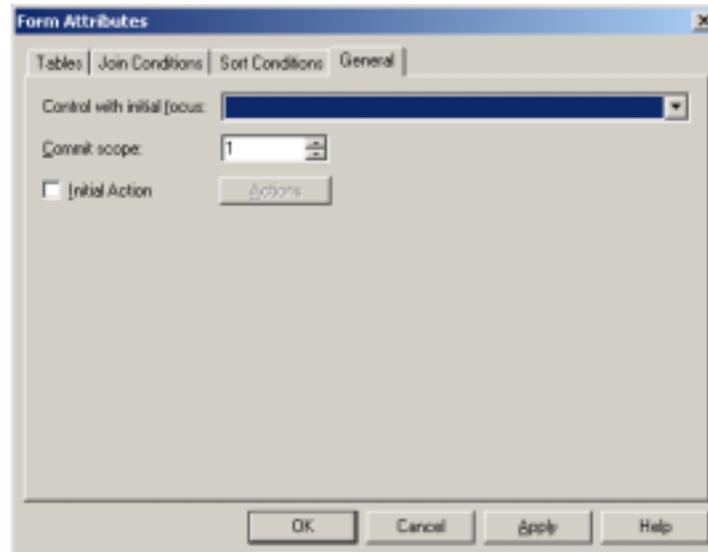
You can specify an initial action, initial focus, and commit scope for your form.

Specifying an initial action

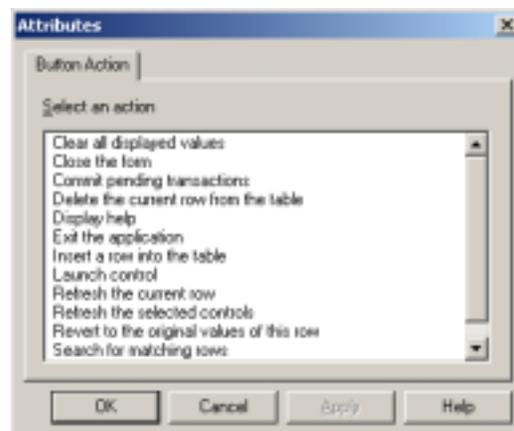
You can specify an initial action for a form. An initial action is an automatic event that occurs each time the form is opened. An example would be to automatically display Help when the form is opened.

To specify an initial action for a form:

1. Select Form --> Attributes The Form Attributes window opens.
2. Select the General tab. The General page opens.



3. Check the **Initial Action** check box, and click **Actions**. The Attributes window opens.



4. From the Attributes window, select an initial action, and click **OK**. The initial action is set. When the form is opened by the user, it will automatically run the specified action. The Form Attributes dialog box closes.

Specifying an initial focus

You can specify an initial cursor focus for a form. When the user first opens the form, the control with initial focus is the active control.

To specify the initial focus:

1. Select Form --> Attributes The Form Attributes window opens.
2. Click the General tab. The General page opens.
3. From the Control with initial focus list, select the control that you want to have the initial focus.
4. Click **OK**. The Form Attributes dialog box closes. When opened by the user, the form will automatically set the cursor focus to the specified control.

Specifying a commit scope

You can specify a commit scope for a form. The commit scope is the number of actions that must take place before a unit of work is completed and the data is committed to the DB2 database.

To specify a commit scope:

1. Select Form --> Attributes The Form Attributes window opens.
2. Select the **General** tab, The General page opens.
3. In the Commit scope field, type the maximum number of rows to insert before committing changes to the database.

Note: If this value is omitted or zero, all of the rows are inserted before a commit occurs. Otherwise, this value specifies the maximum number of rows to insert before committing changes to the database. All **SAVE** actions are subject to the commit scope.

4. Click **OK**. The Form Attributes window closes.

Saving a form

You can save a DB2 Table Editor form to a local file or to a database server.

Saving a form locally

You can save a form to a file on your local hard drive. Once a form has been saved to a file or database it can be distributed to users for use in searching and editing tables.

To save a form locally::

1. From the main window of DB2 Table Editor Developer, select File --> Save As. The Save As window opens.
2. Navigate to the location where you want the form to be saved.
3. Type the name for the form, and click **Save**. DB2 Table Editor Developer saves the active form to a local file.

Saving a form to a database server

You can save a form at a database server. Once a form has been saved to a file or database it can be distributed to other users for use in searching and editing tables.

To save a form to a database server:

1. From the main window of DB2 Table Editor Developer, select File --> Save at Server. The Save Form window opens.
2. In the Owner field, type the user ID for the catalog owner.
3. In the Name field, type the name of the form that you want to open.
4. *Optional:* In the Comment field, type any comments about the form.
5. *Optional:* To share the form with other users check the Share the object with other users check box.

Note: Choosing to share the file with other users means that other DB2 TableEditor Developer users who are authorized on that server will be able to access and edit your form.

6. Click OK. DB2 Table Editor Developer saves the active form to the database server.

Printing a form

You can print a DB2 Table Editor Developer form.

To print a form:

1. Select File--> Print. The Windows Print dialog box opens.
2. Select the desired print options, and click OK. The active form prints.

Closing a form

Remember to save your work before closing a DB2 Table Editor form.

To close a form, select File --> Close.

Deleting a form

You can delete a DB2 Table Editor form that is stored in a local directory or on a database server.

Deleting a form that is stored locally

To delete a form stored on your local hard drive, use your system's local file management application. For example, you can use Windows Explorer to delete a DB2 Table Editor file stored on your local hard drive.

Deleting a form that is stored on a server

You can delete a DB2 Table Editor form from a database server.

To delete a form that is stored on a server:

1. Select File --> Open From Server. The Open From Server Window opens.
2. From the Server list, select the server that stores the form that you want to delete.
3. In the Owner field, type the name of the owner of the form, and click List. The Object List window opens and it contains a list of all of the forms that are on the specified server and are owned by the specified owner.
4. From the Object list window, select the form that you want to delete, and click Delete. A confirmation dialog box will open.
5. Click **Yes** to confirm the deletion. DB2 Table Editor Developer deletes the selected form from the server.

Adding a control to a form

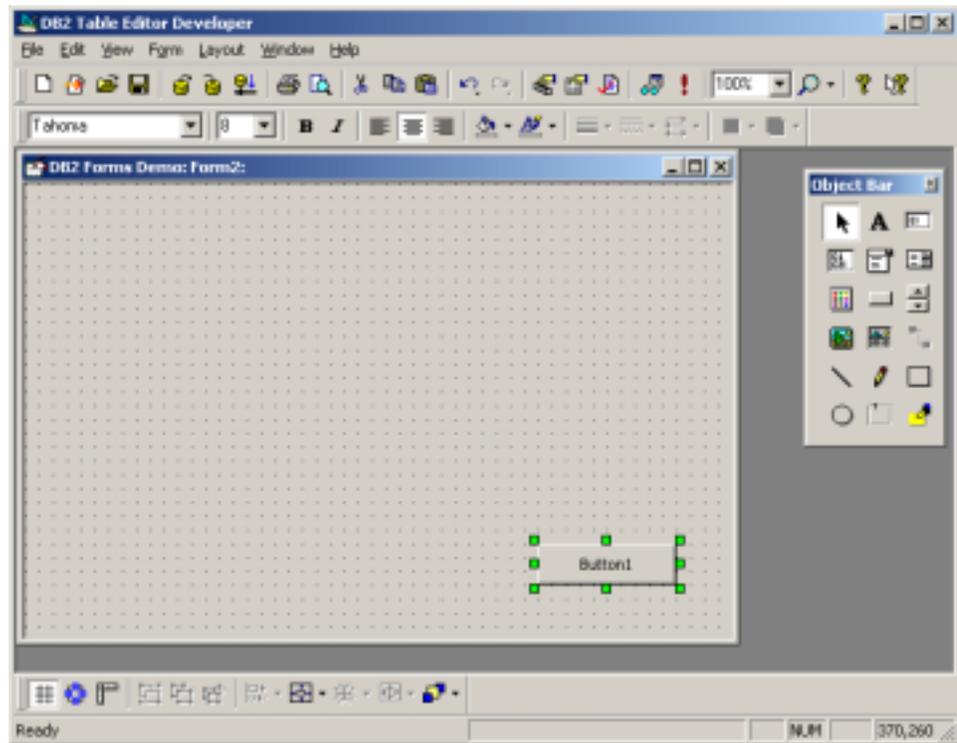
You can add a control to a form with a simple drag-and-drop procedure.

Tip: For a complete list of the name and function of each control, see the Control Reference in the DB2 Table Editor Developer online help system.

To add a control to a form:

1. From the DB2 Table Editor Object tool bar select the control that you want to add to the form.

2. Position the pointer on the form where you want the control to appear. The pointer changes to a hatch mark.
3. Click-and-drag the pointer to create and size the control. The control is drawn on the form.



Deleting a control from a form

You can delete a control from a form. To delete a control from a form, select the control and press the delete key on your keyboard.

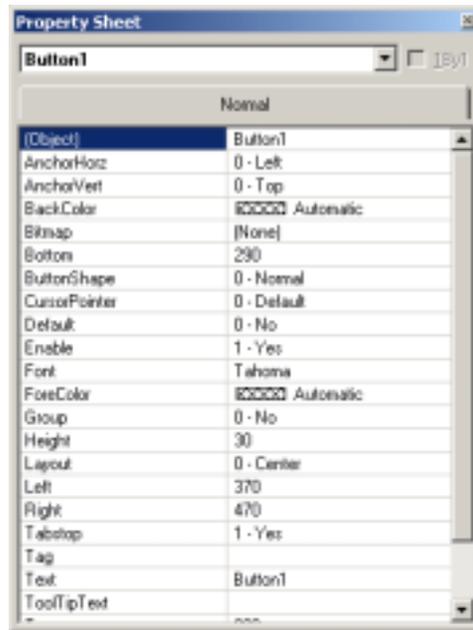
Setting control properties

You can set properties to adjust the appearance of each control on a form.

To set a control property:

1. Select a control.

2. Select Form --> Properties. The Property Sheet window opens.



3. From the table displayed, select and adjust the properties. The appearance of the selected control changes in response to the that properties you set.

Tip: For a complete list of the names and functions of each form property, see the Control Property Reference in the DB2 Table Editor Developer online help.

To change the properties for a different control, select the control from the drop-down box at the top of the form. Change the properties as desired and close the window when you are finished.

Creating a LOB control on a form

Once you have created a LOB control on a form, use the LOB Control Attributes window to set the attributes for your LOB control. You can create LOB controls that display or play text, image or multimedia LOB data.

To create a LOB control on a form:

1. From an open form, create a LOB control on your form using the LOB icon from the Object Bar.
2. Double click the newly created LOB. The LOB Control Attributes window opens.
3. On the General page, type the name of the LOB control in the Object field.
4. Select the format of the objects that will be displayed in your control from the Display menu. The choices are as follows:
 - Show images always - Specifies that objects that will be displayed in the field will always be images. Show multimedia always - Specifies that objects that will be displayed in the field will always be in multimedia format.
 - Show text always - Specifies that objects that will be displayed in the field will always be in text format.
 - Auto Recognize (or show text when format not recognized) - Specifies that DB2 Table Editor will determine the format of the LOB object and display it accordingly. When the format of the object to be displayed is not recognized by DB2 Table Editor, the object will be displayed as text. If you select Auto Recognize (or show text when format not recognized), you must set format

options for each LOB format on the Format page by scrolling through the Format pages for each LOB format using the arrows on the top right of the Format page.

5. *Optional:* Check the Allow external viewing and editing check box to enable viewing and editing of LOBs using an external program such as Microsoft Paint. When this check box is checked, in the DB2 form that you are creating you can double click the LOB control and the LOB will be displayed in an external viewer. Some LOB types can be edited using the external viewer then saved back to the database if desired.
6. *Optional:* Check the Allow menu tools check box to allow editing of text or multimedia LOB objects using the editing tools present on your machine. When editing a text object, you can select the portion of the text that you want to edit, then right click and select the appropriate editing tool from the pop-up menu.
7. On the Column page, select the LOB column that you want to have displayed in the LOB control from the list.
8. On the Format page, specify the options corresponding to the type of LOB that you selected from the Display menu on the General page.
 - If you selected Show images always, you must specify the options for image LOBs.
 - If you selected Show multimedia always, you must specify the options for multimedia LOBs.
 - If you selected Show text always, you must specify the options for text LOBs.
 - If you selected Auto Recognize (or show text when format not recognized) you must specify the options for image, multimedia and text lob, by scrolling through the format pages for each LOB type using the arrow buttons on the top right of the format screen.
9. Click OK.

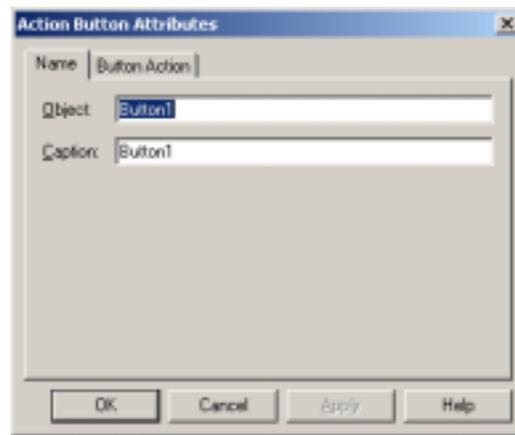
Adding an accelerator key

You can add an accelerator key to a button control.

To add an accelerator key to a button control:

1. Select a button control on your form.
2. Select Form --> Attributes. The Action Button Attributes window opens.
3. In the Caption field, type the caption that you want to appear on the button. To define a letter as the accelerator key for that control, Type an **&** in front of the letter. For example, the caption for the commonly used Close button would be

&Close.



4. Click OK. The letter that you specified is now underlined in the button caption. The user can now perform the button action by holding down the ALT key and typing the corresponding letter.

Using special command characters

DB2 Table Editor users can use special command characters in a text box or combo box control to limit the results of a search. All forms created using DB2 Table Editor Developer automatically recognize the search parameters that are associated with the special command characters listed below.

Table 8. Special Command Characters

Enter:	Result:
<X	limits the search results to those rows which are less than the value of X.
>X	limits the search results to those rows which are greater than the value of X.
[X	limits the search results to those rows which start with the value of X.
]X	limits the search results to those rows which end with the value of X.
^X	limits the search results to those rows which are not equal to the value of X.
~X	limits the search results to those rows which match the pattern of X. For example, to search for the pattern XYZ, you would enter ~%XYZ%.
%X	performs a wildcard search. For example, %ABC returns values that end with ABC, ABC% returns values that begin with ABC, and %ABC% returns values that contain ABC.
*X	performs a wildcard search. For example, *ABC returns values that end with ABC, ABC* returns values that begin with ABC, and *ABC* returns values that contain ABC.
=X	does not substitute for special characters in a search. For example, to search for the string 4*5, enter =4*5.

Working with validation rules

This section describes how to add validation rules to a form or a control. You can add validation rules individually or in groups. This section discusses these topics:

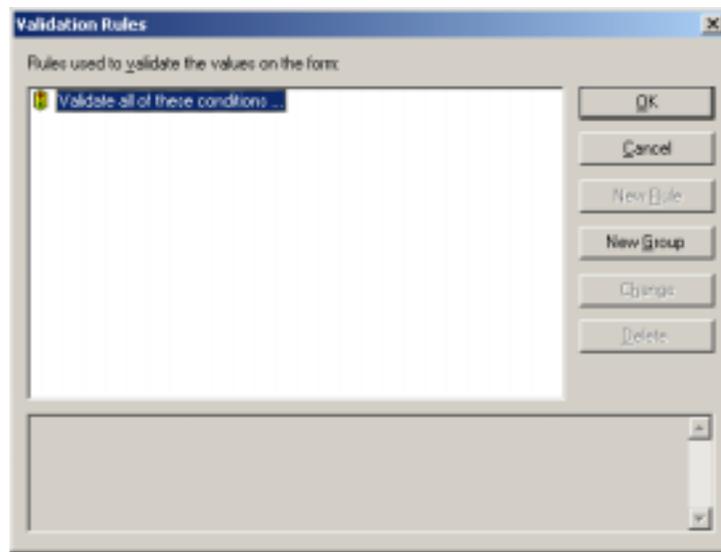
- Creating a Validation Rule
- Creating a Validation Group
- Deleting a Validation Rule
- Deleting a Validation Group

Creating a validation rule

You can create a new validation rule and add it to a control.

To create a new validation rule:

1. Select Form --> Validation Rules. The Validation Rules window opens.



2. Click the New Rule button. the Control window opens.
3. Select the control to which you want to apply the rule.
4. Click Next. The Rule Type window opens.
5. Select the rule type that you want to apply to the control and click Next. A dialog box opens asking for additional validation rule parameters.

Note: For a complete list of the name and function of each validation rule, see the Validation Rules Reference in the DB2 TableEditor Developer online help system.

6. Enter any additional parameters the validation rule requires and click **Next**. The Action window opens.
7. Specify when to apply the validation rule by checking any appropriate check box on the window, and click **Next**.The Error Message dialog box opens.
8. In the Error message field, type the error message that you want to display upon violation of the validation rule. If custom help is available, enter the name and context number of the help file to associate with the error message. Click **Next**. A summary of the validation rule opens.
9. Click **Finish** to create the validation rule.

Creating a validation group

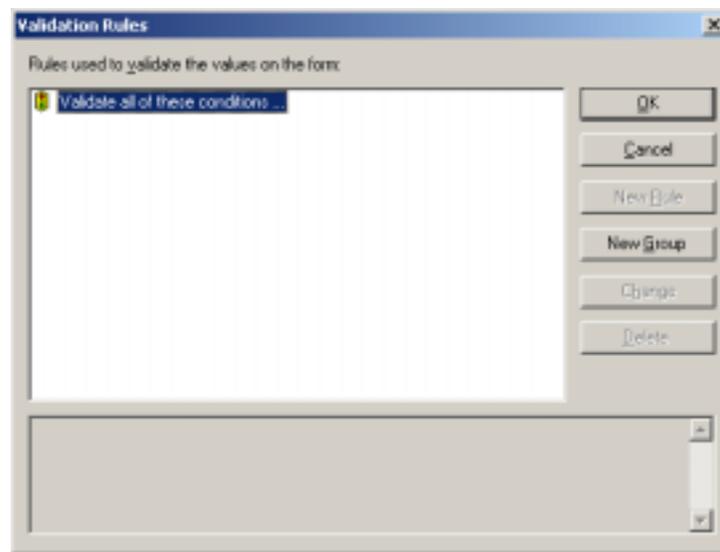
You can create a group of validation rules to validate the contents of a control. Validation groups are based on AND conditions. All of the validation groups must evaluate to true for the form contents to pass validation. The validation rules contained within groups are based on OR conditions. At least one of the rules in a group must evaluate to true for the group to pass validation.

For example:

```
Rule:F_Name field must not have a null value
AND
Rule:L_Name field must not have a null value
AND
Group:Rule:SSN field must have a 9 character entry
OR
Rule:Tax_ID field must have a 9 character entry
```

To create a new validation group:

1. Select Form --> Validation Rules. The Validation Rules window opens.



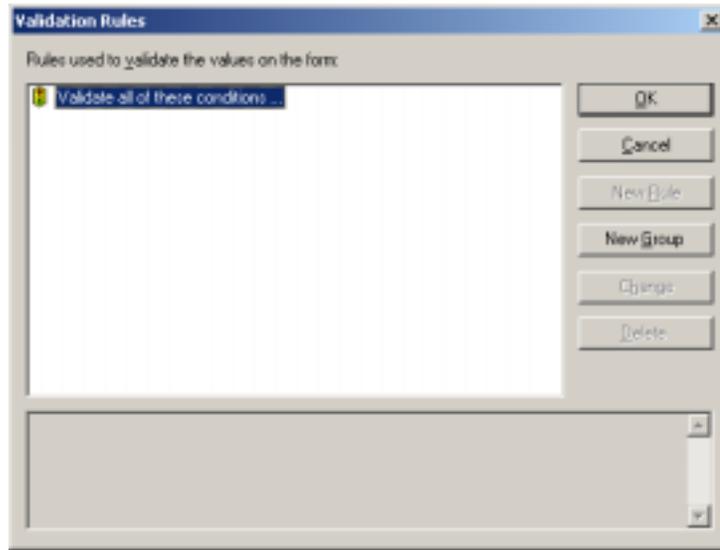
2. Click New Group. A new validation group is created.
3. Click **New Rule**, and a create new validation rules for the group. To create new rule, follow the steps in the following section: "To create a new validation rule:" on page 57

Deleting a validation rule

You can delete a validation rule from a form.

To delete a validation rule:

1. Select Form --> Validation Rules. The Validation Rules window opens.



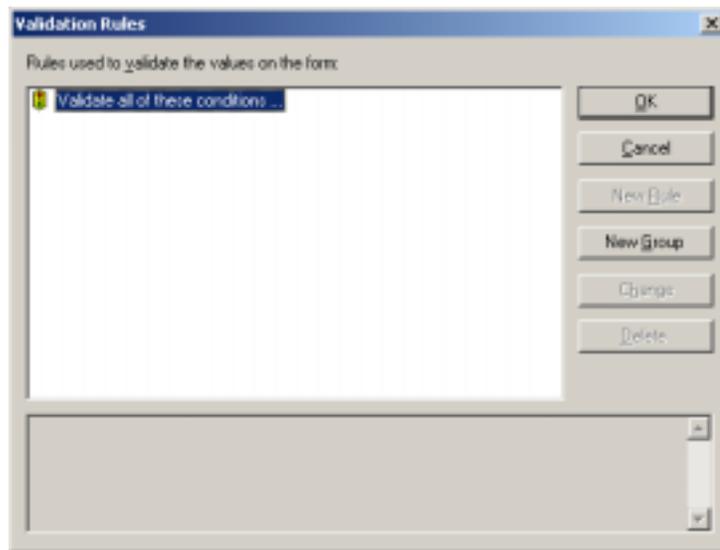
2. Select the rule you want to delete and click **Delete**. The validation rule is deleted.

Deleting a validation group

You can delete a validation group from a form.

To delete a validation group:

1. Select Form --> Validation Rules. The Validation Rules window opens.



2. Select the group that you want to delete and click **Delete**. The validation group is deleted.

Locking when Editing a Table

Locking is implemented as follows in the Edit table feature. The LOCK TABLE statement is used to control locking. The options for locking mode are either SHARE MODE or EXCLUSIVE MODE.

- In share mode, concurrent application processes are prevented from executing all but read-only operations on the table.
- In exclusive mode, concurrent application processes are prevented from executing any operations on the table. Note that exclusive mode does not prevent concurrent application processes that are running at the Uncommitted Read (UR) isolation level from executing read-only operations on the table.

If you select no lock, then DB2 Table Editor does not issue the LOCK TABLE statement, but only obtains locks as needed to perform updates, inserts or deletes. Note that if a table contains a large number of rows, and no row conditions are specified, then not all of data is automatically loaded into the grid until requested. In this case a read cursor will remain open until all of the data is loaded.

Migrating Forms to a Different Environment

Use the Forms, Servers and Tables window to migrate forms from one environment to another. For instance, you can migrate a form from a test environment to a production environment

To migrate a form to a different environment:

1. Open the Forms window by selecting File --> Migrate Forms. The Forms window opens.
2. From the Forms window, select the form or forms that you want to migrate.
3. Click Next. The Servers window opens.
4. From the Servers window, select the server from which you want to migrate the forms, from the Origin server drop-down list.
5. Select the server to which you want to migrate the forms, from the Destination server drop-down list.
6. Click Add.
7. Click Next. If you are migrating a form that contains select statements, the Statements window opens. If your form does not contain select statements, the Tables window opens.
8. For forms that contain select statements:
 - a. Select the origin and destination server pair for the form that you are migrating from the Server Link drop down.
 - b. Select the control that contains the SQL statement from the Controls drop-down list.
 - c. Modify the tables in the select statement in the Select Statement box. The tables referenced in the select statement should be changed to tables in the database to which you are migrating the form.
 - d. Click Apply to apply the changes that you made to your select statement in the Select Statement box. (To undo your changes to the SQL statement, click Rollback.)
 - e. Click Next to move to the next step in the migration process. The Tables window opens
9. From the Tables window, select the origin and destination server combination from the Server link drop-down menu.
10. From the Origin table drop-down menu, select the table in the origin database to which your form connects.

11. From the Destination table drop-down menu, select the table in the destination database to which your form will connect. You can click the browse button to browse for a table using the Object List window.
12. Click Add. The origin and destination tables appear in the Table Links list.
13. Click Finish. Your form is migrated.

Part 4. DB2 Table Editor Java Player and DB2 Table Editor User

DB2 Table Editor is a tool that allows you to access databases using a form. You can open forms, work with data on the database, and commit changes to the database - all without writing complex SQL statements to communicate with the database.

Chapter 10. Overview of the DB2 Table Editor Component and the Java Player Component

DB2 Table Editor provides robust table editing operations, including the ability to:

- Edit tables on multiple DB2 database platforms Maintain referential integrity while performing inserts, updates and deletes
- Quickly search for, and locate specific rows, filter the rows that are displayed, or immediately jump to tables of related data
- Update primary keys
- Commit accumulated data changes at the end of a session, and use find-and-replace for large-scale changes
- Save the specifications of a table editing session (such as row filtering) and call up the same table-editing layout in the future
- Edit with the standard DB2 Table Editor interface, or create prototype forms in seconds with the form building wizard
- Easily locate and launch previously saved Java-based forms, including from links in Web pages or e-mail messages (using DB2 Table Editor Java Player)
- Launch the standard DB2 Table Editor interface from the DB2 Export Facility, DB2 Web Query, or DB2 Control Center
- Access large objects in DB2 databases

Chapter 11. Installing DB2 Table Editor Java Player

The DB2 Table Editor Java Player application can be installed and run on a client machine. To establish connectivity between the Java client and the DB2 database server, install and configure the required client software and drivers, as well as the required network structure.

Installing DB2 Table Editor Java Player

DB2 Table Editor Java Player is a DB2 JDBC application. Calls to JDBC are translated to calls to DB2 CLI, through Java native methods. This dependency requires that the DB2 Client Application Enabler (CAE) component be installed at the client; DB2 Table Editor Java Player uses the CAE to communicate with DB2. A JDBC request flows through DB2 CLI to the DB2 server through the normal CAE communication flow. Your network infrastructure must be configured to support this communications model. DB2 Table Editor Product Support cannot provide support for configuring your network infrastructure.

DB2 Table Editor Java Player can be run as an application or as an applet. The primary difference between the application version and applet version is that the application requires the DB2 Client Application Enabler (CAE) to be installed on the client, and uses the CAE to communicate with DB2, while the applet depends on a Java-enabled Web browser, and does not require any DB2 code installed on the client.

Installing DB2 Table Editor Java Player as an application:

To run DB2 Table Editor Java Player as an application on a client machine, install the Java interpreter, or Java Virtual Machine (JVM), as well as the JDBC driver on the client machine. The JDBC driver is included with the DB2 Client Application Enabler. The DB2 Table Editor application does not include these components.

To install and configure DB2 Table Editor Java Player as an application on a client machine:

1. Install the Java interpreter (version 1.1 or higher).
2. Install the DB2 CAE, including the JDBC driver. The JDBC driver file name is db2java.zip.
3. Install DB2 Table Editor according to the installation instructions. For instructions on installing DB2 Table Editor, see Chapter 4, "Installing DB2 Table Editor's Graphical User Interface" on page 15.

The files required for DB2 Table Editor Java Player will be installed automatically. These files are:

- db2forms.jar - Class file for the DB2 Table Editor Java Player application
- One or more of the following license files.
 - dbehost.lic
 - dbe400.lic
 - dbework.lic
- The HtmlHelp directory - HTML help files for the DB2 Table Editor Java Player application.

Note: The HtmlHelp directory, which contains the help files for DB2 TableEditor Java Player, must be installed in the same directory as db2forms.jar

4. Create a desktop shortcut to access the application. Create a new shortcut with the following target command line:

```
"x:\<location_of_jrew.exe>\jrew.exe" -cp
"x:\<location_of_db2java.zip>\db2java.zip";"x:\
<location_of_db2forms.jar>\db2forms.jar" DB2Forms
Where x:\<location_of_db2java.zip> specifies the location of the IBM DB2
JDBC driver that was installed as part of the DB2 CAE, and
x:\<location_of_db2forms.jar> specifies the location of the DB2 Table Editor
classes that were installed with DB2 Table Editor. For example:

"C:\Program Files\SQLLIB\java\jre\bin\jrew.exe" -cp
"C:\Program Files\SQLLIB\java
\db2java.zip";"C:\DB2 Table Editor\db2forms.jar"
DB2Forms
```

Note: If you are planning to access an Informix™ database using DB2 Table Editor, you also need to add the following to the shortcut target:

```
x:\<location_of_ifxjdbc.jar>\ifxjdbc.jar
```

Where x:\<location_of_ifxjdbc.jar> specifies the location of the Informix JDBC driver on your machine.

Installing DB2 Table Editor Java Player as an Applet

The DB2 Table Editor Java Player application, as well as individual DB2 Table Editor forms, can be installed on a web server, and accessed through a web browser enabled for Java 1.1 or greater. Both the Java application, and individual forms, are downloaded to the client browser via a JAVA applet embedded in an HTML file. There is no need to install or configure any DB2 Table Editor software on the client machine. However, you must install and configure the required network structure to establish connectivity between the browser client and the web server.

Delivering a JDBC applet relies on a web browser enabled for Java 1.1 combined with the JDBC driver. The driver consists of a JDBC client and a JDBC server. When the user loads an HTML file containing the DB2 Table Editor applet tags, the JDBC client driver is downloaded to the client along with the applet. When a connection to a DB2 database is requested by the applet, the client opens a TCP/IP socket to the JDBC server on the machine where the Web server is running. After a connection is set up, the client sends each of the subsequent database access requests from the applet to the JDBC server through the TCP/IP connection. The JDBC server then makes corresponding CLI (ODBC) calls to perform the task. Upon completion, the JDBC server sends the results back to the client through the connection. Your network infrastructure must be configured to support this communications model. DB2 Table Editor Product Support cannot provide support for configuring your web server and network infrastructure.

To install DB2 Table Editor Java Player as an applet:

1. On the client machine: Install a web browser enabled for Java 1.1 or greater.
2. On the host machine, install and configure a web server, and establish connectivity between the web server and client machine via HTTP and TCP/IP protocols.
3. Install the DB2 database infrastructure on the same host machine as the web server. DB2 includes the JDBC driver; the file is named db2java.zip.
4. Install DB2 Table Editor according to the installation instructions on any DB2 Table Editor compatible machine. You will copy the DB2 Table Editor Java

Player program files from this machine to your web server. For instructions on installing DB2 Table Editor, see Chapter 4, "Installing DB2 Table Editor's Graphical User Interface" on page 15

Note: The files that are required for DB2 Table Editor Java Player will be installed automatically when you install DB2 Table Editor. These files are:

- db2forms.jar - Class file for the DB2 Table Editor Java Player application.
- The HtmlHelp directory - HTML help files for the DB2 Table Editor Java Player application.

5. Copy the following files and directories to the home directory on your web server:

- db2forms.jar
- db2java.zip
- One or more of the following license files.
 - dbehost.lic
 - dbe400.lic
 - dbework.lic

You must copy the license file that corresponds to the platform on which the DB2 database to which you want to connect resides.

- HtmlHelp Directory (Including all sub folders)

The Home Directory is the directory where any HTML files containing a DB2 Table Editor applet will reside. This is the directory where your HTTP Server serves files from. This directory structure must be followed for DB2 Table Editor Java Player to function properly as an applet.

6. To run a DB2 Table Editor form via an applet, use DB2 Table Editor Developer to create a form. Then, in DB2 Table Editor developer, automatically generate an HTML file with the embedded applet tags for the form. Or, create your own custom HTML file containing the following applet tag and parameters:

```
<applet code = "DB2FormsApplet.class" width=N height=N
align="Position_of_form" archive="db2forms.jar,db2java.zip">
<param name=HostName value='Name_of_host'>
<param name=Port value='Port_number'>
<param name=RDBName value='RDB_name'>
<param name=UserID value='ID_of_user'>
<param name>Password value='Password_of_user'>
<param name=FormOwner value='owner_of_form'>
<param name=FormName value='name_of_form'>
</applet>
```

Where:

- Width - Specifies the applet display width in the browser window, in pixels.
- Height - Specifies the applet display height in the browser window, in pixels.
- Align - Specifies the applet display alignment in the browser window, as "center", "left", or "right".
- HostName - Specifies the name of the HTTP server from which the JAVA applet will run.
- Port - Specifies the port number that the JDBC applet server is listening on.
- RDBName - Specifies the DB2 database alias name.

- UserID - Specifies the user ID with which to log in to the server (optional).
- Password - Specifies the password with which to log in to the server (optional).
- FormOwner - Specifies the owner of the form.
- FormName - Specifies the name of the form to display in the browser window.

Note: Make sure that the location that is specified in the applet for the JDBC driver, db2java.zip, is correct. If the JDBC driver file is not located in the same directory as the HTML file containing this applet, the applet attribute "archive" must reflect the correct location of db2java.zip. If necessary, edit the applet attribute "archive" to reflect the location of the JDBC driver, or copy db2java.zip to the same directory as the HTML file.

If you are planning to access Informix databases, the following parameters must also be added to the HTML file:

```
<param name=Informix_host value= '<http_server_name>' >
<param name=Informix_port value= '<port number>' >
<param name=Informix_Database value= '<database name>' >
<param name=Informix_Server value= '<server name >' >
```

Where:

- **Informix_host** – Specifies the name of the http server from which the JAVA applet will run.
- **Informix_port** - Specifies the port number that the JDBC applet server is listening on.
- **Informix_Database** - Specifies the name of the Informix database to which you want to connect.
- **Informix_Server** - Specifies the name of the Informix server from which the JDBC applet will run.

7. Install the HTML file that contains the DB2 Table Editor applet tags in the same directory as db2forms.jar, db2java.zip, the license file, and the HtmlHelp directory:

```
Home Directory
db2forms.jar
db2java.zip
name_of_file.html
<License file>.lic
HtmlHelp Directory
```

Where name_of_file is the name of the HTML file containing the DB2 Table Editor applet tags.

8. Open the URL that is used to connect to the DB2 Table Editor Java player. The URL is as follows: http://<http server>/<path>/xxx.html Where HTTP server is the name of your HTTP server, <path>/xxx.html is the path to and name of the html file that contains the applet parameters that are used to open the form. You can open the HTML file from any web browser enabled for Java 1.1, and run the DB2 Table Editor Java Player application, or DB2 Table Editor form.

Command Line Parameters

You can define settings and actions to take effect when DB2 Table Editor Java Player is started. These parameters are defined on the DB2 Table Editor Java Player command line. They can be used to preset settings, or to run unattended sessions. For help on adding command line parameters to an icon or start menu, refer to your operating system's help facility. The following parameters are available from the DB2 Table Editor Java Player command line:

Table 9. DB2 Table Editor Java Player command line parameters

Parameter	Result
/IServer:servername	The /IServer:servername parameter defines the server where the form specified on the /IFormName:formname parameter is stored
/IFormName:formname	The /IFormName:formname parameter defines the owner and name of a form stored at a database server to run after starting DB2 Table Editor User. The format for the form name is owner.formname. To use the /IFormName:formname parameter, you must also specify the /IServer:servername parameter.
/IFormFile:formfile	The /IFormFile:formfile parameter defines the location and name of a locally stored form to run after starting DB2 Table Editor User.
/IUserID:userID	The /IUserID:userID parameter defines the user ID to use when running a procedure specified with the /IFormName:formname parameter. It is used in conjunction with the /IPassword:password parameter.
/IPassword:password	The /IPassword:password parameter defines the password of the user specified with the /IUserID:userID parameter.

Note: The /IPassword parameter includes the user's password in plain text.

For example, the following string would pass the Server, the UserID, and the Password parameters to the database when DB2 Table Editor is launched:

```
jrew -cp c:\sql11ib\java\db2java.zip;"c:\Program Files\Ibm\Db2Forms\db2forms.jar"  
DB2Forms /IServer:MyServer /IUserID:demo /IPassword:demo
```

Chapter 12. Working with the DB2 Table Editor Component and the Java Player Component

Before working with DB2 Table Editor component or DB2 Table Editor Java Player component, you must define and configure the database servers to which you want to connect using DB2 Table Editor Console. For more information on this, see “Configuring DB2 Table Editor” on page 27

This chapter details some of the basic features and functions of DB2 Table Editor, including:

- Opening the application
- Elements of the main window
- Opening a form
- Saving work to the database
- Closing a form
- Closing the application
- Edit related feature

Opening the application

Instructions follow for opening your application.

Opening DB2 Table Editor Windows User

Once DB2 Table Editor had been installed on your computer you can access it through the Windows Start menu. To open the application, select **Start --> Programs --> DB2 Table Editor**.

Opening DB2 Table Editor Java Player component

DB2 Table Editor Java Player component is accessed through an icon on your desktop. Your administrator will set this up. To Start the DB2 Table Editor Java Player component, double-click the Icon on you desktop.

Connecting to an Informix Database using the DB2 Table Editor Java Player component

You can connect to an Informix Version 9.3 database using the DB2 Table Editor Java Player component.

Note: You must use the Informix JDBC driver installed on your machine in order to connect to an Informix database.

To connect to an Informix database using DB2 Table Editor Java Player component:

1. Ensure that the path to the Informix JDBC driver is added to the path of your shortcut to the DB2 Table Editor Java Player Component. For more information on this step, see “To install and configure DB2 Table Editor Java Player as an application on a client machine:” on page 67.
2. Open the Set Database window by selecting View --> Set Server.
3. Select the Informix radio button to specify that you will be accessing an Informix database server.

4. In the Database Server box, select the database server to which you want to connect.
5. In the Host name or IP address field specify the host name or IP address of the database server to which you want to connect.
6. In the Port Number field, specify the port number for the database server to which you want to connect.
7. In the Database Name field, specify the name of the database to which you want to connect.
8. Click OK.

Opening a form

You need to open a form to use DB2 Table Editor. Forms may be saved on your computer or on a server.

To open a form that is stored locally:

1. Select File --> Open. The Open window opens.
2. Locate and select the file you want to open, and click Open. The form opens within the Main window.

Note: The steps to open a local file are the same for both Windows and Java versions.

To open a form that is stored on a server using DB2 Table Editor for Windows:

1. Select File --> Open from server. The Open From Server window opens.
2. From the Server list, select the server name where your form is stored.
3. If you know the owner name, enter it in the Owner field.
4. If you know the name of the form, enter it in the Name field.
5. Click the List button. A list of forms, that match the criteria that you specified, opens in the Object List window.

Note: The percent sign (%) acts as a wildcard symbol. Enter the percent sign in the Owner and Name fields to return forms with any Owner, or with any Name.

6. From the Object List window, select the form that you want to use. Click OK. The selected form opens.

To open a form that is stored on a server from DB2 Table Editor Java Player component:

1. Select File --> Open from server. The Open From Server window opens.

Note: If the server is provided as a command line parameter, DB2 TableEditor uses this server as a default, and the Set Server Dialog Box does not display.

2. Specify the database where the form that you want to access is stored, and click **OK**. The Open From Server window opens.
3. Specify the owner and name of the form that you want to open, and click **List**.

- Note:** If you do not know the owner or name of the form, enter the wildcard symbol %, and click **List**.
4. Enter your User ID and Password if prompted, and click **OK**. The Object List dialog window opens, containing a list of forms.
 5. Select the form that you want to open, and click **OK**. The form opens.

Editing a table

One of the most powerful features in DB2 Table Editor is the ability to create forms to edit tables in the database.

When you use the Edit Table feature, you create a form to insert, update, and delete records from a table in a database (based on your particular privileges). These forms include the ability to update primary keys. When you create a form, you specify a Primary Table. the primary table for each form is that table that will be updated with insert, update and delete actions.

The specifications for these forms may be saved using the DB2 Table Editor component application for reuse by the Windows or Java application.

Editing a table

You can automatically create a new form for a table using the Edit Table feature. You can then use this form to search and edit the table.

To create a new form using the Edit Table feature:

1. Select File --> Edit Table. The Edit Tables window opens.
2. Select the server on which the table that you want to edit is stored and click Next.
3. In the Owner field, type the user ID of the owner of the table that you want to edit.
4. In the Name field, type the name of the table that you want to edit and click Search.

Tip: You can choose from a list of tables that are stored at a database server. From the Edit Table window, enter the wildcard character % in the Owner and Name fields (and in the Database Name, and Tablespace name fields, which are available if you are accessing a z/OS or OS/390 system).

5. Select the table that you want to edit, and click Next. the table that you select here is the Primary table for the form. For more information on primary tables, see "Primary Table" on page 22.
6. From the Data View box, select the data view you that you prefer.

Note: The form layout is similar to any Windows dialog box, displaying edit fields and buttons. The table layout is tabular in appearance, displaying data in a grid of rows and columns.

7. In the Available columns list, check all of the columns that you want to appear in the form.
8. In the Available buttons list, check all of the buttons that you want to appear in the form.

Note: No buttons are used if you edit the form in table layout mode.

9. Click **Next**.

10. Click the Add button that is adjacent to the Row Conditions box. The Row Condition window opens.
11. Use the fields in the Row Condition window to specify a Row Limit. Click Add. The Row Condition window closes.
12. Click the Add button that is adjacent to the Sort Conditions box. The Sort Condition window opens.
13. Select the column or columns from the list, on which you want your data sorted.
14. In the Sort Direction box, specify whether you want your data to be sorted in ascending or descending order.
15. Click Add. the Sort Condition window closes.
16. If you selected the Table Layout format in step 6, select a Save Mode from the Save Mode box.

Note: The On Leave Row save mode sends changes you make to DB2 every time you click outside a row you have made changes to. The At End save mode sends changes you make to DB2 when you are finished with all edits to the form.

17. If you selected the At End save mode in step 16, select a Lock Table option from the Lock Table box. The options are as follows:
 - Select No to edit the table without locking it.
 - Select Shared to prevent other applications from performing any but read-only operations on the table while you are editing it.
 - Select Exclusive to prevent concurrent applications from performing any operations on the table.
18. Click Finish. The form or table appears in the DB2 Table Editor component main window.

You can now use the form to search and edit the table.

Working with primary keys

If the table that you are editing contains a primary key column, that column is represented by the addition of a key icon to the column header. Any changes that you make to primary key columns are automatically cascaded to all matching key columns throughout the database.

The Edit Related feature

Edit Related is an option for full-screen edit forms when there is a primary key included on the form that has dependent tables. If this is the case, using the full-screen edit pop-up menu produces a menu that includes the Edit Related option.

When Edit Related is selected, DB2 Table Editor first checks to see if there is more than one related table. If so, it produces a list of tables to choose from. You can select the related table that they wish to edit. This dialog is not shown if there is only one related table. The Edit Table wizard is then displayed for the related table and will produce a new form for the related table when the Finish button is clicked. If the Edit Related option was invoked when either a cell or row was selected, then the Edit wizard dialog for the related table is automatically populated with row conditions for the row/cell values that were selected.

Working with the Table Editor

You can edit tables in form mode and in table layout mode. Both modes use a wizard to give you full access to the data within one table. Both traditional forms and full-screen edit forms can be saved locally and at a database server. Once saved, these forms can be opened and edited at a later time.

Form mode

Form mode uses a wizard to create a form that is designed to edit the table that you select. The wizard dynamically builds the controls and columns that are displayed in the form based on the selections that you make. Changes are committed when you click the Update, Insert, or Delete buttons.

Table Layout mode

Table layout mode uses a wizard to create a tabular representation of the table that you select. You can make edits within the table grid itself. You can choose whether changes are committed when you are done editing the table, or any time that you select another row in the editing grid.

You navigate the table layout by using the cursor keys, the Tab key, and the mouse. You edit a row by selecting the cell that you want to edit and typing directly in the cell. You can also use the options available in the Edit Menu and the table layout edit popup menus.

Working with Rows in DB2 Table Editor

You can perform actions on a row in a table by right clicking the row, then selecting one of the following options:

- Insert Row - insert a new row into the table directly below the current row.
- Insert Duplicate Row - insert a new row into the table directly below the current row, with values identical to those in the current row.
- Delete Row - remove the current row from the table.
- Copy Row - copy the selected row and place it on the Clipboard, replacing the current contents of the Clipboard.
- Find in Row - search the table for the specified information.
- Replace in Row - replace the specified value in the table with a new value.
- Update Row - update the current row with new information.
- Undo Row - undo the changes you have made.
- Edit Related - open the Edit Table wizard with the data filled in to generate a table layout form for the table related to the primary key you have selected.

Deleting a row in the Table Editor

You can delete a row from a DB2 table using the Table Editor.

Note: The Primary table for the form will be updated. For more information on primary tables, see "Primary Table" on page 22.

To delete a row from the form layout: Use the Search button to locate the row that you want to delete and click Delete. The row is deleted from the table.

To delete a row from the table layout: From an open form, right-click on the gray blocks to the left of the table and select Delete Row from the pop-up menu. The row is deleted from the table.

Inserting a row

You can insert a new row into a table from the Table Editor.

Note: The Primary table for the form will be updated. For more information on primary tables, see "Primary Table" on page 22.

To insert a row using the form layout: In an open form, enter the information that you want to appear in the new row and click Insert. The new row is inserted at the end of the table.

To insert a row using the table layout:

1. From an open form, right-click on the gray blocks to the left of the table and select Insert Row. A new blank row is created directly below the row where you clicked.
2. Enter the information that you want to appear in the new row into each cell in the table and click outside the new row. The new row is inserted into the table.

To insert a row by duplicating an existing row using the table layout:

1. From an open form in table layout, right click the gray block to the left of the row that you want to duplicate and select Insert Duplicate Row from the pop-up menu. A new row is created directly below the row where you clicked, identical to the selected row.
2. Make any desired changes to the duplicate row and click outside the new row. The new row is inserted into the table.

Updating a row

You can update a row from the Table Editor.

Note: The Primary table for the form will be updated. For more information on primary tables, see "Primary Table" on page 22.

To update a row using the form layout::

1. Use the Search button to locate the row that you want to update.
2. Update information in that row using the form.
3. Click Update. The row is updated with the new information.

To update a row using the table layout::

1. In an open form, select the cell that you want to edit.
2. Type the information that you want to appear in the row and click outside the row. The row is updated with the new information.

Working with Columns in DB2 Table Editor

You can perform actions on a column in a table by right clicking the column, then selecting one of the following options:

- Sort Ascending - sort the contents of the column in ascending order.
- Sort Descending - sort the contents of the column in descending order.
- Copy Column - copy the selected column and place it on the Clipboard, replacing the current contents of the Clipboard.
- Find in Column - search the column for the specified information.
- Replace in Column - replace the specified value in the column with a new value.
- Columns - open the Columns dialog box.
- Edit Related - open the Edit Table wizard with the data filled in to generate a table layout form for the table related to the primary key you have selected.

Working with Cells in DB2 Table Editor

You can perform actions on a cell in a table by right clicking the cell, then selecting one of the following options:

- Cut Value - remove the selected value from the cell and place it on the Clipboard, replacing the current contents of the Clipboard.
- Copy Value - copy the selected value and place it on the Clipboard, replacing the current contents of the Clipboard.
- Paste Value - insert the contents of the Clipboard at the insertion point, replacing any selection.
- Zoom - open the Zoom dialog box. From the Zoom dialog box you can view the values in a column in HEX. The values that are shown those HEX bytes are for the value as they are stored in your program, in the Windows ANSI CCSID for Windows or UTF-8 for Java. This is not necessarily the bytes as their value is stored in the database.
- Find - search the table for the specified information.
- Replace - replace the specified value in the table with a new value.
- Launch - run the value of the selected cell, if the value is a valid file or application name.
- View/Edit LOB - open the Column LOB Type Associations dialog box in order to view or edit the selected LOB.
- Edit Related - open the Edit Table wizard with the data filled in to generate a table layout form for the table related to the primary key that you have selected.

Saving work to the database

Changes and updates are automatically sent in batches to the database. When you close DB2 Table Editor any changes that you have made since the last commit to the database are automatically saved.

If your developer has designed a Commit button into a the form, this can be used to commit changes to the database immediately.

Working with LOBs (Large Objects)

When editing a table with LOB values, DB2 Table Editor first retrieves information about the LOB in the form of a LOB Locator by indicating what application should be used to open it. Once you have indicated the appropriate application, you can simply double click the LOB in order to view it. When running the Java Player component as an applet, you can use the internal viewer to view LOBs that are of basic image or character types as the applet does not have access to locally installed applications. The internal viewer supports the following types of LOB data:

- JPEG or .jpg files
- Graphics Interchange Format or .gif files
- Bitmap or .bmp files
- text files

DB2 Table Editor supports LOBs for DB2 connection types that are implemented to handle LOBs. DRDA connections connecting to DB2 for Multiplatforms cannot handle LOBs, however CLI connections can. For DB2 for OS/390 LOBs are supported for both DRDA and CLI connections.

Note: When you are using DB2 TableEditor Java Player component or the DB2 Table Editor Control Center Plug-in, Long VarChar data is shown and treated as CLOB.

Indicating which application will be used to open Large Objects (LOBs)

You can view and edit large objects (LOBs) directly from an open form. You can automatically launch an appropriate application to work with the LOB that you have selected, with the LOB open for editing.

To indicate which application will be used to open LOB:

1. From an open form, right click the cell that contains the reference to the LOB that you want to work with. A drop-down menu opens.
2. Click View/Edit LOB. The Column LOB Type Associations dialog box opens.
3. There are two ways to indicate which application will be used to view and edit the LOB.
 - To indicate which application will be used to view and edit the LOB via its file extension, select the File Extension radio button. Type the file extension of the LOB in the File Extension field and click the Browse button and browse until you find the executable file for the application with which you want to view and edit the LOB.
 - To indicate which application will be used to view and edit the LOB via the mapping column, select the Mapping Column radio button. Select the desired mapping column from the drop-down list then click the Mapping button. The File Extension Mapping dialog box opens. Select the file extensions for the formats listed in the column from the LOB Type Associations window. This option is only available for forms that contain a mapping column.
4. *Optional:* Save the options on the form by selecting File --> Save at Server.

Note: The column LOB associations information, including mapping if it has been used, is saved when the form is saved. The next time the form is opened this information will be used.

5. Click OK. The appropriate application launches with the selected LOB open for viewing or editing.

Note: Once you have indicated the appropriate application, on subsequent attempts to view the LOB you can simply double click the LOB in order to view it. If you want to change the application that is used to open the LOB, Hold down the shift key, then right click the LOB that you want to view and select View/Edit LOB. The Column LOB Type Associations dialog box opens and you can change your selection.

Part 5. DB2 Table Editor Control Center Plug-in

The DB2 Table Editor Control Center plug-in adds DB2 Table Editor menu items to the DB2 Table pop-up menu and adds a button that starts DB2 Table Editor as an add-in tool to the DB2 for Multiplatforms toolbar.

Chapter 13. Overview of the DB2 Table Editor Control Center plug-in

The DB2 Table Editor Control Center Plug-in is an extension to the DB2 Control Center. For more information about the Control Center, refer to the IBM DB2 Administration Guide or The IBM DB2 Control Center online help. The DB2 Table Editor Control Center plug-in adds DB2 Table Editor menu items to the table popup menu and adds a tool bar button to the Control Center tool bar that starts DB2 Table Editor as an add in tool.

Using the DB2 Table Editor Control Center Plug-in

There are two ways to use the DB2 Table Editor Control Center plug-in. From the DB2 Control Center, you can click the DB2 Table Editor icon in the tool bar and DB2 Table Editor will open. It is also possible to open DB2 Table Editor right to the table that you want to edit. From the DB2 Control Center, you can select the table that you want to edit, and right-click that table. From the table pop-up menu that appears, you can choose from three table editor options, all of which will open DB2 Table Editor ready to edit the table that you have selected. The options are as follows:

- Edit Tabular - This option opens DB2 Table Editor, ready to edit the table that you selected with the table data displayed in a table format.
- Edit Columnar - This option opens DB2 Table Editor, ready to edit the table that you selected with the table data displayed in columns.
- Edit Wizard - This option opens DB2 Table Editor, ready to edit the table that you selected using the DB2 Table Editor's, Edit Table wizard.

Chapter 14. Installing the DB2 Table Editor Control Center Plug-in

The DB2 Table Editor Control Center plug-in adds DB2 Table Editor menu items to the table pop-up menu and adds a tool bar button to start DB2 Table Editor as an add-in tool.

Installing the DB2 Table Editor Control Center Plug-in:

The DB2 Table Editor Control Center Plug-in is an extension to the DB2 Control Center. For more information about the Control Center, refer to the IBM DB2 Administration Guide. The DB2 Table Editor Control Center plug-in adds DB2 Table Editor menu items to the table popup menu and adds a tool bar button to start DB2 Table Editor as an add in tool.

To install the DB2 Table Editor Control Center Plug-in:

1. Copy the following files and directories to the *SQLLIB\cc*:
 - db2forms.jar
 - One or more of the following license files.
 - dbehost.lic
 - dbe400.lic
 - dbework.lic

You must copy the license file that corresponds to the platform on which the DB2 database to which you want to connect resides.

- HtmlHelp Directory (Including all sub folders)

By default this directory is *SQLLIB\cc*. The Control Center requires that these files be located in the Java classpath so that they can be loaded when the Control Center is started.

2. Copy the version of *db2plug.zip* that corresponds to the version of DB2 that you are using, into *SQLLIB\cc* directory. There are two version of *db2plug.zip*: *db2plug.v6zip* and *db2plug.v7zip*. If you are using DB2 Version 6, then copy *db2plug.v6zip*, if you are using DB2 V7, then copy *db2plug.v7zip*.
3. Rename the release specific version that you copied to: *db2plug.zip*

Note: If there is already another version of *db2plug.zip* in *SQLLIB\cc*, you must combine the contents of the existing *db2plug.zip* and the DB2 Table Editor *db2plug.zip*. For Example, if you are using the Zip utility, use the following command:

```
zip -r0 db2plug.zip com\companyname\myplugin\*.class
```

Where the plug-in package name is: *com.companyname.myplugin*

4. Locate the *db2cc* file in the *SQLLIB\bin*. If you are using Windows, the file is named *db2cc.bat*. If you are using the Unix or Linux, the file is named *db2cc*.
5. Update *db2cc* to include both "*db2plug.zip*" and "*db2forms.jar*". The file names must follow a *-c* option. With newer releases of DB2, you must add the *-c* option. If you are using an older version of DB2 which has the *-c* option specified in *db2cc*, you can append the values to the end of the existing *-c* option. The following is an example of *db2cc* after you have added the files if you are working with DB2 V6:

```

IF "%1" == "wait" GOTO WAIT
db2javit -j:"CC" -d:"CC" -c:"db2plug.zip;db2forms.jar"
-o:"-mx128m -ms32m" -a:"%1 %2 %3 %4 %5 %6 %7 %8"
GOTO END

:WAIT
db2javit -j:"CC" -d:"CC" -c:"db2plug.zip;db2forms.jar" -w:
-o:"-mx128m -ms32m" -a:"%2 %3 %4 %5 %6 %7 %8 %9"
GOTO END

:end

```

The following is an example of *db2cc* after you have added the files if you are working with DB2 V7:

```

IF "%1" == "wait" GOTO WAIT
db2javit -j:"CC" -d:"CC" -c:"db2forms.jar"
-o:"-mx128m -ms32m" -a:"%1 %2 %3 %4 %5 %6 %7 %8"
GOTO END

:WAIT
db2javit -j:"CC" -d:"CC" -c:"db2forms.jar"
-w:-o:"-mx128m -ms32m" -a:"%2 %3 %4 %5 %6 %7 %8 %9"
GOTO END

:end

```

6. If you are running the Control Center as a Java applet, complete the following steps:
 - a. Copy the *db2forms.jar* file where the <codebase> tag points to in *db2cc.htm*.
 - b. Update *db2cc.htm* to include "*db2plug.zip*" and "*db2forms.jar*" in the archive list.

You can now start Table Editor from within Control Center by selecting a DB2 Table Editor menu item from the Table popup menu or the new tool bar button.

Part 6. DB2 Table Editor for the ISPF interface

There are three client versions of DB2 Table Editor for z/OS that are used to access z/OS or OS/390 servers: a Windows version, a Java version, and an ISPF version. This Part of the User's Guide addresses the ISPF version.

This part is intended for database administrators and developers who want to use DB2 Table Editor to edit tables and modify data in an ISPF environment. This document assumes that you are familiar with ISPF and basic DB2 concepts.

Chapter 15. Overview of DB2 Table Editor, ISPF Component

DB2 Table Editor for ISPF allows you to work with DB2 tables in an ISPF environment using a quick and easy interface. You can access, edit, and search data stored on databases that are linked to DB2 Table Editor. You can also perform Inserts, Updates, and Deletes without writing SQL.

Chapter 16. Customizing DB2 Table Editor for ISPF

To install DB2 Table Editor, complete the installation instructions that are found in the program directory. In order to install DB2 Table Editor, you must have DBADM authority.

In order to use DB2 Table Editor you must have also installed DB2 Change Accumulation Tool CTC Common Code program number 5655-F5501.

Once you have installed DB2 Table Editor you must customize it before you use it.

Licensing

The DB2 Table Editor ISPF component has one license use management (LUM) key associated with it. The LUM key allows you to install DB2 Table Editor ISPF component on z/OS machines and to connect to other z/OS machines using Aliases. For information about the licensing required to connect to other DB2 subsystems, contact your IBM representative.

Customization summary

Complete the installation instructions can be found in the Program Directory. After you install DB2 Table Editor, you must tailor some files for your system. The following steps configure your system.

Table 10. Overview of steps for configuring DB2 Table Editor

Step	Description	Job
1	Create the control file.	<i>ETICNTFL</i>
2	Bind the packages	<i>ETI#BND1</i>
3	Bind the plans	<i>ETI#BND2</i>
4	Bind the packages and plans for all other subsystems to which you want to connect	<i>ETI#BND1 ETI#BND2</i>
5	Edit SETISAMP members ETICLIST and ETI	<i>ETICLIST and ETI</i>

Customizing DB2 Table Editor

After you have installed DB2 Table Editor, follow these steps to tailor some files for your system.

Step 1. Create the control file

1. Edit SETISAMP member *ETICNTFL*.
2. Add the appropriate job card to *ETICNTFL*.
3. Follow the directions contained in *ETICNTFL* to make the other required changes to *ETICNTFL*.
4. Run *ETICNTFL*.

Step 2. Bind the packages

1. Edit SETISAMP member *ETI#BND1*
2. Add the appropriate job card to *ETI#BND1*.
3. Specify a valid STEPLIB name.

4. Specify a valid DBRMLIB name.
5. Change the DSN SYSTEM name to the appropriate name for your environment.
6. *Optional:* Rename your plan with any name that is appropriate for your site. The default plan name is ETIPLAN1.
7. Follow the directions contained in *ETI#BND1* to make the other necessary changes to EIT#BND1.
8. Run EIT#BND1.

Step 3. Bind the plans

1. Edit SETISAMP member *ETI#BND2*.
2. Add the appropriate job card to *ETI#BND2*.
3. Specify a valid STEPLIB name.
4. Specify a valid DBRMLIB name.
5. Change the DSN SYSTEM name to the appropriate name for your environment.
6. *Optional:* Rename your plan with any name that is appropriate for your site. The default plan name is *ETI#BND2*.
7. Follow the directions contained in *ETI#BND2* to make the other necessary changes to EIT#BND2.
8. Run *ETI#BND2*.

Step 4. Bind the packages and plans for all other sub systems to which you want to connect

Repeat steps 2 and 3 for each subsystem that you plan to run against. Use a unique subsystem identifier each time, SYSTEM(XXXX), where XXXX is the subsystem identifier. You can use the same plan name for all subsystems.

Step 5. Edit SETISAMP members ETI and ETICLIST

Edit SETISAMP member ETICLIST by changing the variables to meet your environment's requirements according to the instructions in the member.

1. Edit SETISAMP member ETICLIST.

In ETICLIST:

- Specify the location of the control file
- Specify the locations of the libraries that are used at your site
- Specify the high-level qualifier.
- Change:
 - '&CTCLEVEL..SETIMENU') UNCOND STACK to '&CTCLEVEL..SCTCMENU') UNCOND STACK
 - '&CTCLEVEL..SETIPENU') UNCOND STACK to '&CTCLEVEL..SCTCPENU') UNCOND STACK
 - '&CTCLEVEL..SETILOAD') UNCOND STACK to '&CTCLEVEL..SCTCLOAD') UNCOND STACK

The following is an example of an ETICLIST member. There are notes within the text of this sample file (in lower case) indicating the parts of the file that need to be modified as part of step five.

```
PROC 0 DIRECT(NO) CNAME(NONE) TNAME(NONE) SNAME(NONE) +
  WARNB(YES) WARNB(YES) LOCBRSE(YES) + <if yes Table Editor will
lock in browse mode>
  ETILEVEL(HILEVEL.ETI.PDS) + < the prefix of the ETI libraries>
  CTCLEVEL(HILEVEL.CTC.PDS) + < the prefix of the CTC libraries>
  DB2CNTFL(ETI.DB2.CONTROL) <the control file>

ISPEXEC VPUT (DIRECT CNAME TNAME SNAME) ASIS
ISPEXEC VPUT (DB2CNTFL WARNB WARNB LOCKBRSE) ASIS

CONTROL NOMSG
FREE FILE(ETIPRINT)
CONTROL MSG

ISPEXEC LIBDEF ISPLIB DATASET ID('&ETILEVEL..SETIMENU' +
                                '&CTCLEVEL..SETIMENU') UNCOND STACK
<change the above line to '&CTCLEVEL..SCTCMENU') UNCOND STACK>
ISPEXEC LIBDEF ISPLIB DATASET ID('&ETILEVEL..SETIPENU' +
                                '&CTCLEVEL..SETIPENU') UNCOND STACK
<change the above line to '&CTCLEVEL..SCTCPENU') UNCOND STACK>
ISPEXEC LIBDEF ISPLIB DATASET ID('&ETILEVEL..SETILOAD' +
                                '&CTCLEVEL..SETILOAD') UNCOND STACK
<change the above line to '&CTCLEVEL..SCTCLOAD') UNCOND STACK>

ALLOC FILE(ETIPRINT) SYSOUT
ETI$MAIN

FREE FILE(ETIPRINT)
```

2. You must customize the CLISTS for ISPF dialog access to DB2 Table Editor. Edit SETISAMP member ETI. Change the PROC0 statement to match your installation's high level qualifiers for the ETI and CTC libraries by replacing ETILVL with the high level qualifier of the ETI libraries and CTCLVL with the high level qualifier of the CTC libraries. Edit SETISAMP member ETI and specify the library where CLIST is located.

The following is an example of an ETI member. There is a note within the text of this sample file (in lower case) indicating the part of the file that needs to be modified as part of step four.

```
PROC 0 DIRECT(NO) CNAME(NONE) TNAME(NONE) SNAME(NONE) +
      WARNA(YES) WARNB(YES) LOCKBRSE(YES) +
      CLISTLIB( LOCATION OF PDS )

CONTROL NOMSG
FREE FILE(ETICLIST)
CONTROL MSG

ALLOC FILE(ETICLIST) DATASET('&CLISTLIB') SHR REU <The PDS where ETI
is currently a member.>

ALTLIB ACTIVATE APPLICATION(CLIST) FILE(ETICLIST)

ISPEXEC SELECT CMD(ETICLIST DIRECT('&DIRECT') -
      NAME('&CNAME') TNAME('&TNAME') SNAME('&SNAME') -
      WARNA('&WARNA')<column selection warning
-- a warning appears when the value is YES and columns
are excluded
WARNB('&WARNB') - <lock table warning
-- a warning appears
when the value is YES and a LOCK TABLE command is
issued from the DB2 Table Editor main panel>
NEWAPPL(ETI) PASSLIB

ALTLIB DEACTIVATE APPLICATION(CLIST)
```

The libraries listed above should match the names that you allocated during installation. Include the load library as part of your authorized list. (Some of the programs in the load library must be APF authorized in order to run.)

Note: Examine each member to ensure that it is complete and that the data set names are correct and review all statements before submitting each job.

Step 6. Optional: Add DB2 Table Editor to the DB2 Administration Tool Launchpad

DB2 Administration Tool, program number 5655-E64, provides a central location to launch one or more DB2 tools. You can add DB2 Table Editor to the DB2 Administration Tool Launchpad. This step is optional. For detailed information on how to enable an IBM DB2 tool for the launchpad, see DB2 Administration User's Guide (SC27-0974), available on the Web at : <http://www.ibm.com/software/data/db2imstools/details/>. DB2 Administration Tool, program number 5655-E64, adds launchpad support for installed IBM DB2 tools that have an ISPF interface. It provides a central location to launch one or more DB2 tools. You must have applied APAR PQ45029 and created the Launchpad table.

To add DB2 Table Editor to the DB2 Administration Tool launchpad:

1. Locate the sample install CLIST in the following library: (xxx.xxxx(ETIADBI). Where xxx.xxx is the library where DB2 Table Editor is stored.

2. Modify the sample CLIST (ETIADBI) by following the instructions contained in the CLIST.
3. Run the ETIADBI CLIST and DB2 Table Editor will be added to the launchpad.

Completing customization

To complete the installation of DB2 Table Editor on your system, DB2 Table Editor must be configured for your system and default values must be set.

Table 11. Overview of steps for configuring DB2 Table Editor

Step	Description	Job/Panel
1	Start DB2 Table Editor.	<i>ETI</i>
2	Set up the configuration file for DB2 Table Editor.	Enter DB2 System Parameters panel
3	Set the DB2 subsystem parameters	Update Parameters for DB2 Subsystem panel
4	Repeat steps 2 and 3 for each DB2Subsystem to which you want to connect.	Enter DB2 System Parameters panel and Update Parameters for DB2 Subsystem panel

Step 1. Start DB2 Table Editor

Start DB2 Table Editor by running the SETISAMP library member ETI. For example, from the ISPF TSO command panel, issue this command:

```
EX &CLIST(ETI)
```

Where &CLIST is the name of the DB2 Table Editor SETISAMP library that is installed.

The primary panel for DB2 Table Editor opens. (For more information on the DB2 Table Editor main panel, see the section called “The DB2 Table Editor main panel” on page 99.)

First Paragraph

Step 2. Setup the configuration file for DB2 Table Editor

1. From the DB2 Table Editor Selection Menu, type **S** on the command line to set up the configuration file for DB2 Table Editor.

The Enter DB2 System Parameters panel opens.

```

-----Enter DB2 System Parameters-----
Command ==>

GDG Base Model DSN      ==> PDRJS.JUNK
DB2 Control Dataset    ==> RSC.DB2.CONTROL
(Pre-allocated)

Enter DB2 Subsystem Info:

DB2 Subsystem ID       ==>          (1-4 Character Subsystem ID)

Valid command selection values are:
 1: ZPARM, BSDS, and Load Library Information
 2: DB2 Table Editor

```

2. In the DB2 Subsystem ID field, type the subsystem identifier (SSID) for the DB2 subsystem that you want to configure. This field has a four-character maximum.

Step 3. Set the DB2 subsystem parameters

1. Type **1** on the Command line and press the Enter key. The Update Parameters for DB2 Subsystem panel opens.

```

----- Update Parameters for DB2 Subsystem R61A
Command ==>

Enter or Update Specific DB2 Parameters :

DB2 ZPARMs Member      ==>
DB2 Bootstrap DSN #01  ==>
DB2 Bootstrap DSN #02  ==>
DB2 Loadlib1           ==>
DB2 Loadlib2           ==>
DB2 Loadlib3           ==>
DB2 Loadlib4           ==>
DB2 Loadlib5           ==>

```

2. In the Update Parameters for DB2 Subsystem panel, enter information specific to a particular DB2 subsystem, including ZPARM, BSDS, and load library.
 - a. In the DB2 ZPARMs Member field, type the eight-character ZPARM load module member name generated for this DB2 subsystem.
 - b. In the DB2 Bootstrap DSN #01 and the DB2 Bootstrap DSN #02 fields type the full data set names of the two bootstrap data sets that are being used by this DB2 Subsystem.
 - c. In the DB2 Loadlib1 to DB2 Loadlib5 fields, type the names of the data sets that comprise the current loadlib concatenation for DB2. The loadlib usually consists of:
 - a subsystem-specific DSNEEXIT library
 - the base DSNEEXIT library for the current DB2 version
 - the base DSNLOAD library for the current DB2 version

Note: There are two extra Loadlib fields. They can be left blank or used to enter any other libraries contained in the loadlib.
 - d. Press Enter to submit your changes. A message appears on the screen to indicate that your changes have been saved.

- e. Press PF3 the Update Parameters for DB2 Subsystem panel closes. A message stating that a record has been added appears. Press PF3 to return to the previous screen.

Note: For information on setting up a data sharing environment, see the section: “Setting up a data sharing environment”.

3. Type **2** on the Command line. The second Update Parameters for DB2 Subsystem panel opens.

```
-----DB2 Table Editor - Update Parameters for DB2 Subsystem R61A
Command ==>

Enter or Update Specific DB2 Parameters:

Plan Name      ==> ETIPLAN1
```

4. In the second Update Parameters for DB2 Subsystem panel, type information specific to the DB2 Table Editor parameters.
 - a. In the Plan Name field, type the eight character plan name used for DB2 Table Editor. This is the same plan name that was used in member ETI#BIND in the SETISAMP library when the plan for DB2 Table Editor was bound.
 - b. Press Enter to submit your changes. A message appears on the screen to indicate that your changes have been saved.
 - c. Press PF3 to return to the previous screen.
5. Press Enter to submit your changes. A message appears on the screen to indicate that your changes have been saved.

Step 4. Setup the configuration file and set the DB2 subsystem parameters for all other DB2 Subsystems to which you want to connect

Repeat steps 2 and 3 for all DB2 subsystems to which you want to connect.

Setting up a data sharing environment

When setting up a data sharing environment, first, identify the subsystem ID (SSID). This will be used as the group name. For any required data sets, such as the bootstrap data set (BSDS), you can use any one of the member’s BSDS names. You only have to specify one member’s BSDS name because each member’s BSDS name contains the necessary information for the entire group.

Note: You can configure for each subsystem within the group, though the entire data sharing environment will still be used. You will spend less time setting up and save effort if you configure for a single group ID.

Permissions

In order to use DB2 Table Editor, the user must have EXECUTE permission on the DB2 Table Editor plan and on the load lib. You must also have read permission on the control file. The Administrator must also have UPDATE authority on the control file.

Chapter 17. Working with DB2 Table Editor for ISPF

This chapter contains information on using DB2 Table Editor for ISPF. It describes the basic elements of DB2 Table Editor and contains instructions for opening, viewing, and editing tables.

Getting started

To use DB2 Table Editor, select the table that you want to view or edit and make choices about the way that you want to view or edit it. You can make these choices from the DB2 Table Editor main panel.

The DB2 Table Editor main panel

The DB2 Table Editor main panel is the primary menu from which you navigate through DB2 Table Editor. Use this panel to make choices about the tables that you want to view or edit.

```
ETI$MAIN  V4R3 ----- IBM DB2 Table Editor ----- 2002/06/20 16:05:55
Option ==> _____
DB2 Subsystem ID          Current          User:
-----
Table Creator Like      _____ (DB2 Like Criteria for Selection List)
Table Name Like        _____

Fetch Limit            0_____ (Zero has no limit)
Max Char Display       30_____ (1-32768)

Lock Table             N          (Shared / Exclusive / No)

NULL default value     Y          (Yes / No)
Show separator         Y          (Yes / No)
Separator character    ,
Decimal character      .

Save options           Y          (Yes / No)

Enter END command to return to ISPF, or 'S' to go to the setup screen.
```

Field	Description
DB2 Subsystem ID	Type the four character DB2 Subsystem ID (SSID) for the subsystem that contains the table you want to edit or view.
Current SQLID	Type the SQLID for the User under whom you want to run all SQL.
User	Displays the current User.
Table Creator Like	Type the table creator name for which you want to search. You can use wildcards in your search. Use the * character or the % character as the wildcard character. Typing an * returns all table creators. You can also type a LIKE clause here, if desired.
Table Name Like	Type the table name for which you want to search. You can use wildcards in your search. Use the * character or the % character as the wildcard character. Typing an * returns all table names.
Fetch Limit	Type the maximum number of rows that you want the query to return. Type 0 to indicate no limit.

Field	Description
Max Char Display	Type the maximum number of characters that you want to be displayed in each column. the default is 30 characters. If a field exceeds this maximum it will appear in white and you must use the zoom command to see all of the data. For more information of using the zoom command, see the section: "To edit a long cell:" on page 106.
Lock Table	Type N (No) to edit the table without locking it. Type S (Shared) to prevent other applications from performing anything but read-only operations on the table while you edit it. Type E (Exclusive) to prevent concurrent applications from performing any operations on the table while you edit it.
NULL default value	Type Y to have empty cells filled with nulls when you insert a blank row. Type N to have empty cells left empty when you insert a blank row. Note: Rows that are marked as not null will not be filled with nulls, even if Y is selected. If you have selected N , but later edit a cell to insert a null value, DB2 Table Editor will change your selection from N to Y
Show separator	Type Y to have thousands separated by a specified character when numbers greater than 999 are shown. Type N to show no thousands separator character when numbers greater than 999 are shown.
Separator character	Type the character to be used to separate the thousands in when numbers greater than 999 are shown.
Decimal character	Type the character to be used to represent a decimal when numbers are shown.
Save options	Type Y to save the options that you have specified in this panel. Type N if you do not want to save these options.

Opening a table

You can use DB2 Table Editor to view or edit table data.

To open a table:

1. From the DB2 Table Editor main panel, in the DB2 Subsystem ID field, type the ID for the DB2 Subsystem where the table that you want to open is located.
2. In the Table Creator Like field, type the name of the creator of the table that you want to open.
You can use wildcards or a LIKE clause to search for a table. To see a list of all tables, type * in the Table Creator Like field.
3. In the Fetch Limit field, type the maximum number of rows that you want the query to return. Type **0** to indicate no limit.
4. In the Max Char Display field, type the maximum number of characters that you want to be displayed in each column. If a field exceeds this maximum it will appear in white and you must use the **zoom** command to see all of the data.
5. In the Lock Table field, type **N** (No) to edit the table without locking it. Type **S** (Shared) to prevent other applications from performing anything but read-only

operations on the table while you edit it. Type **E** (Exclusive) to prevent concurrent applications from performing any operations on the table while you edit it.

- In the NULL default value field, type **Y** to have empty cells filled with null when you insert a blank row. Type **N** to have empty cells left empty when you insert a blank row.

Note: Rows that have the not null attribute associated with them will not be filled with nulls, even if **Y** is selected.

If you have selected **N**, but later edit a cell to insert a null value, DB2 Table Editor will change your selection from **N** to **Y**.

- In the Save Options field, type **Y** to save the options that you have entered on this panel, or, type **N** if you do not want to save these options.
- Press the Enter key. A list of tables that match the criteria that you entered is displayed in the Table Selection.

```

ETI$DPTB V4R3 ----- Table Selection ----- 2002/06/25 12:15:30
Option ==> _____ Scroll ==> PAGE
-----+>
                                         Row 1 of 39
Creator Like USERID__                      DB2 Subsystem: SUBSYS
Tables Like *_____

-----
Cmd  Table Name      Creator  Database  Tablespace  OBID
-   TABLE1         USERID  DBNAME1   TSNAME      01036  2001-06-13-11.5
-   TABLE2         USERID  DBNAME2   TSNAME1     00003  2001-05-29-16.1
-   TABLE3         USERID  DBNAME1   TSNAME      01033  2001-06-08-17.3
-   TABLE4         USERID  DBNAME2   TSNAME1     01030  2001-06-12-22.0
-   TABLE5         USERID  DBNAME4   TSNAME2     00004  2001-05-29-16.1

Valid Line Commands: (Select table, Count rows)
  
```

- If the table the you want to open is not listed you can search further as follows: In the Table Name Like field, type the name of the table that you want to open. You can use wildcards or a LIKE clause to search for a table. To see a list of all tables, type ***** in the Table Name Like field.
- Type **S** in the Command (cmd) column next to the table that you want to open. Press the Enter key.

The Select Columns panel opens.

```

ETI$DPSC V4R3 ----- Select Columns ----- 2002/06/20 16:23:45
Option ==> _____ Scroll ==> PAGE
-----
And/Or on Where Clause      A          Creator ==> USERID
Long or Short Data Types    L (L or S) Table ==> TABLENAME
-----
Select  Ord  Srt  Frz  Type          Column Name      Where Clause
1      ___  A   N   DECIMAL(5,2)  COL1
2      ___  A   N   DECIMAL(5,2)  COL1
3      ___  A   N   DECIMAL(5,2)  COL2
4      ___  A   N   DECIMAL(5,2)  COL3
***** Bottom of Data *****
  
```

Field	Description
And/Or on Where Clause	Type A to connect all WHERE clause statements with AND. Type O to connect all WHERE clause statements with OR.

Field	Description
Long or Short Data Types	Type L to specify that you want to view the long name for data type names shown in the table (as shown in the screen capture above). Type S to have abbreviated data type names shown in the table.
Table	Displays the name of the current table.
Creator	Displays the creator of the current table.
Select	The Select column contains a number that indicates the order in which the corresponding column will be selected. You can change the numbering in the columns if desired. Type N , or leave the Select column blank to exclude the corresponding column from the generated table. Note: To mark all rows to be included in the generated table, (rows with a number in the Select column will be included) type include all on the command line, this will re-number all rows. To mark all rows to be excluded from the generated table, (marked N in the Select column) type exclude all on the command line.
Ord	Use this option to indicate by which columns to order the data. Type 1 next to the column that you want to order by first. To order by additional columns, type numbers next to the desired columns in the order that you want them to be sorted.
Srt	Type A to sort the corresponding column in ascending order. Type D to sort the column in descending order.
Frz	Type Y in the Frz column to freeze the associated column when the table is being edited. When the table is open in the Edit Table Rows panel, the column appears in the table, it will be left justified. When you scroll through the table, the frozen column will always be on the screen.
Type	Displays the data type of the corresponding column.
Column Name	Displays the column name of each column in the selected table.
Where Clause	Type a WHERE clause to apply to the corresponding column. For example, > 100.

11. *Optional:* In the Where Clause column, type **A** to connect all WHERE clause statements with AND. Type **O** to connect all Where Clause statements with OR.
12. *Optional:* In the Select column, you can modify the number that is associated with the row in the table. The number in the Select column indicates the order in which the corresponding column will be selected. To exclude the corresponding column from the generated table, type **N**, or leave the Select column blank.

Note: To mark all rows to be included in the generated table, (rows with a number in the Select column will be included) type **include all** on the command line, this will re-number all rows. To mark all rows to be excluded from the generated table, (marked **N** in the Select column) type **exclude all** on the command line.
13. *Optional:* In the Ord field, type **1** next to the column that you want to order by first. To order by additional columns, type numbers next to the desired columns in the order that you want them to be sorted.

14. *Optional:* In the Srt field, type **A** to sort the corresponding column in ascending order. Type **D** to sort the column in descending order.
15. *Optional:* in the Frz column type **Y** to freeze a column, so that when it appears on the Edit Table Rows panel, it will remain on the screen when you scroll left and right through the table. Frozen columns are left justified on the edit table rows panel.
16. *Optional:* If you want to edit or view the generated SQL for your table, Type SQL on the command line and press the Enter key. The Generate Select Statement panel opens. Make the changes that you want to make to the SQL and press PF3.
17. *Optional:* Type **COUNT** on the command line. A popup box is displayed indicating the number of rows that will be returned by the your SQL statement.
18. *Optional:* Type **E** on the command line to edit the table.
19. *Optional:* Type **B** on the command line to browse the contents of the table.
20. Press the Enter key. The table will be displayed according to your specifications.

Editing a table

You can use DB2 Table Editor for z/OS to perform inserts, updates, and deletes to your DB2 tables. DB2 Table Editor will commit the changes that you make automatically when you finish editing the table.

Locking

Locking is implemented as follows in the Edit table feature. The LOCK TABLE statement is used to control locking. The options for locking mode are either SHARE MODE or EXCLUSIVE MODE.

- In share mode, concurrent application processes are prevented from executing all but read-only operations on the table.
- In exclusive mode, concurrent application processes are prevented from executing any operations on the table. Exclusive mode does not prevent concurrent application processes that are running at the Uncommitted Read (UR) isolation level from executing read-only operations on the table.

If you select no lock, then DB2 Table Editor does not issue the LOCK TABLE statement, but only obtains locks as needed to perform updates, inserts or deletes. Note that if a table contains a large number of rows, and no row conditions are specified, then not all of data is automatically loaded into the grid until requested. In this case a read cursor will remain open until all of the data is loaded.

Updating

When DB2 Table Editor, ISPF component, is updating a row, it looks for the row in one of two ways.

- If the table is locked, DB2 Table Editor uses a unique index (if one exists) to find the row that is to be updated.
- If the table is not locked, or it does not have a unique index, DB2 Table Editor examines all columns in the row, with the exception of some types of columns, to determine whether it has found the correct row. The following columns are not examined:
 - LOB columns
 - A string that is too long for a LIKE clause. Strings up to 4000 characters in length can be used in a like clause in z/OS.

Note: If a string is greater than 4000, and no duplicate rows exist it can be used, if duplicate rows exist, then the long string must be replaced with a short string. If a string is less than 255 characters in length, an equal (=) operator is used.

- User-defined data type (UDT) columns.

Updating or Deleting Duplicate rows

After the ISPF component of DB2 Table Editor has completed an update or a delete, it checks to see whether more than one row has been updated or deleted. If so, DB2 Table Editor rolls back the transaction, opens a cursor to the first occurrence of the row, and updates or deletes that row.

Inserting a blank row

You can insert a blank row into a table using the Edit Table Rows panel. You can insert a row into a table that contains data or into an empty table.

To insert a blank row:

1. In the Select Columns panel, ensure that there is a number in the Select column of the rows that you want to edit. If you do not want to have a particular column selected type N in the Select column or leave it blank.
2. Type E on the command line and press Enter. The Edit Table Rows panel opens.

```

ETI$EDIT V4R3 ----- Edit Table Rows ----- 2002/06/25 12:19:50
Option ==>> _____ Scroll ==>> PAGE _____
----->
Table ==> TABLENAME          Creator ==> USERID
-----
Cmd S N ONE      N COLNAME          N COLNAME2
___ N      1 N CONTENT          N CONTENT2
___ N      2 N DATA            N DATA2
___ N      3 N INFO              N INFO2
___ Y      N NAME                N NAME2
***** Bottom of Data *****

Valid Line Commands: Delete, Insert, Repeat, Undo, Copy, Before, After, Form

```

3. In the Edit table rows panel, type I in the Command (cmd) column of the row below which you want the new row to appear.

Note: To remove the row that you inserted and exit the Edit Table Rows panel without committing your changes, type CAnCel on the Command line.

4. Press the Enter key.
A blank row appears below the row that you selected.
5. Enter data in any fields required in that row. For example, if one of the cells must have a value that is not null, you must enter a value in that cell. The changes that you have made will be committed in DB2 when you exit the table that you are editing.
6. To cancel the action that you have taken before you exit the table, enter the command to cancel (CAnCel) on the command line and press Enter. You will exit the panel without committing the changes.

To insert more than one blank row.

To insert more than one row into the table that you are editing, type **I**, followed by the number of new rows that you want to insert in the Command column of the row below which you want the new rows to appear.

To insert a row into an empty table:

1. From the DB2 Table Editor Main panel, open an empty table.
2. In the Edit Table Rows panel, type **Insert** on the command line. A blank row is inserted into the table.

Repeating rows

You can repeat one row or a range of rows using the Edit Table Rows panel.

To repeat a row:

In an open table, type **R** in the Command (cmd) column of the row that you want to repeat. Press the Enter key.

The changes that you have made will be committed in DB2 when you exit the table that you are editing.

To repeat a row more than once:

In an open table, type **R** followed by the number of times that you want the row repeated, in the Command (cmd) column of the row that you want to repeat. Press the Enter key. For example **R5** would repeat a row five times.

The changes that you have made will be committed in DB2 when you exit the table that you are editing.

To repeat a range of rows:

In an open table, type **RR** in the Command (cmd) columns of 2 rows, the row at the beginning and the row at the end of the range of rows that you want to repeat. Press the Enter key.

The changes that you have made will be committed in DB2 when you exit the table that you are editing.

Copying rows

You can copy rows using the copy command from the Edit Table Rows panel.

To copy a row:

1. Type **C** in the Command column of the row that you want to copy.
2. Type **A** in the Command column of the row above which you want the row to be inserted, or, type **B** in the Command column of the row below which you want the row to be inserted.
3. Press the Enter key.

To copy more than one row:

1. Type **C**, followed by the number of rows that you would like to copy in the Command column of the first row that you want to copy.

For Example, to copy five rows, type **C5** in the Command column of the first row in the sequence of rows that you want to copy. This will result in that row and the four rows below it being copied.

2. Type **A** in the Command column of the row above which you want the rows to be inserted, or, type **B** in the Command column of the row below which you want the rows to be inserted.
3. Press the Enter key.

To copy a range of rows:

1. In an open table, type **CC** in the Command (cmd) column of two rows, the row at the beginning and the row at the end of the range of rows that you want to copy.
2. Type **A** in the Command column of the row above which you want the rows to be inserted, or, type **B** in the Command column of the row below which you want the rows to be inserted.
3. Press the Enter key.

Editing a row

You can use the Edit Table Rows panel to edit a row in a table. When editing a row you can choose to view the row alone (one row is shown at a time) in form mode or you can view the row in tabular mode where you will see more than one row displayed at once.

To edit a row:

1. In an open table, move the cursor to the cell in the row that you want to edit and type the necessary changes.

Note: If the data in the cell is too long to appear on the screen the cell will appear in white. To view or edit the to the entire contents of the cell, use the **zoom** Command line. For instructions on using the zoom command, see the section: "Editing a long cell".

To edit rows one row at a time, select the row that you want to edit and type **F** on the Cmd column corresponding to that row. Press Enter. The row appears in form format. In form mode, you can browse through the rows in a table using PF10 to go to the previous row and PF11 to go to the next row. To return to tabular format press PF3.

2. When you have completed your changes, press the Enter key.
DB2 Table Editor marks the row as updated and highlights the changes in yellow. The changes that you have made will be committed in DB2 when you exit the table that you are editing.

Editing a long cell

Cells that are too long to be displayed on the screen appear in white. You can view or edit the entire contents of a long cell by using the **zoom** command. The zoom command can be used in two ways: In an open table, type **zoom** on the command line, then move the cursor to the cell that you want to edit; or, place the cursor on the cell that you want to edit and press PF4

When making changes to VARCHAR columns, DB2 Table Editor will automatically calculate the length of the column based on the input value and trailing pad characters.

To edit a long cell:

1. In an open table, place the cursor on the cell that you want to edit.
2. Press PF4.
The Column Editor panel opens.

```

ETI$EXPL V4R3 ----- Column Editor ----- 2002/06/20 17:50:16
Option ==> _____ Scroll ==> PAGE
-----
Table ==> TABLENAME      Creator ==> USERID
Column ==> COLNAME        Type ==> DATATYPE
Length ==> 00000005       Max ==> 00000016
Pad ==> 40                Null ==> N
-----

***** ***** Top of Data *****
***** -+-----1-----2-----3-----4-----5-----6-----7--
..... COLUMN CONTENTS
***** ***** Bottom of Data *****

```

Note: The fields that appear in this panel vary based on the data type of the selected cell.

Field	Description
Table	The name of the table in which the selected cell exists.
Creator	The creator of the table in which the selected cell exists.
Column	The name of the column in which the selected cell exists.
Type	The data type of the column in which the selected cell exists.
Length	The length in characters of the data in the selected cell.
Max	The maximum length in characters of the data in the selected cell.
Pad	Contains the character that the contents of the cell will be padded with when the contents of the selected cell are shorter than the value in the Max field.
Null	This value is N if the selected cell does not have a value of null. The Value is Y if the selected cell does have a value of null.

3. View or edit the data.

Note: When you are editing a VARCHAR column in zoom mode, the length is determined automatically by removing the PAD character. This means that when you are editing a column of a specified length, if you type a value that is longer than the specified length, the length of the column will be changed by DB2 TableEditor ISPF component by removing the necessary number of pad characters. The length cannot exceed the maximum length specified by the data type.

4. When you are finished viewing or editing the data, press PF3.

If you have made changes to the cell, DB2 Table Editor updates the cell with the information that you entered and returns to the Edit Table Rows panel.

Viewing the contents of a cell as hex

You can view the contents of a cell with hex in the zoom panel.

To edit a cell with a hex editor: In the Zoom panel, type **hex on**, on the Command (cmd) line and press the Enter key. The hex editor is turned on. To turn the hex editor off, type **hex off**. The hex editor is turned off.

Getting information about a column heading

You can get information about a column heading in the Edit Table Rows panel.

To get information about a column heading:

Place the cursor over a column heading and press the PF4 key. Information about that column will appear.

Deleting rows

You can delete a row or many rows using the Edit Table Rows panel.

To delete a row:

In an open table, type **D** in the Command (cmd) column of the row that you want to delete. Press the Enter key.

DB2 Table Editor marks the selected row for deletion. The row will appear in green. The delete will be committed to the database when you exit the table. For information on committing data, see the section: “Committing changes to DB2” on page 109.

To delete more than one row:

In an open table, type **D** followed by the number of rows that you want to delete in the Command (cmd) column of the first row that you want to delete. For Example **D5** would delete the row next to which you typed the command and the four following rows. Press the Enter key.

DB2 Table Editor marks the selected rows for deletion. The rows will appear in green. The deletes will be committed to the database when you exit the table. For information on committing data, see the section: “Committing changes to DB2” on page 109.

To delete a range of rows:

In an open table, type **DD** in the Command (cmd) column of two rows, the row at the beginning and the row at the end of the range of rows that you want to delete. Press the Enter key.

DB2 Table Editor marks the selected rows for deletion. The rows will appear in green. The deletes will be committed to the database when you exit the table. For information on committing data, see the section: “Committing changes to DB2” on page 109.

Undoing an action

You can undo an action before it is committed. You can undelete a row before committing the delete to the database, remove an inserted row before the insert has been committed to the database, and undo changes that have been made.

To undo an action:

In an open table, type **U** in the Command (cmd) column of the row that contains the result of the action that you want to undo. For example, if you wanted to undo an insert, you would type **U** in the command column of the row that was inserted. Press the Enter key.

To undo an action in more than one consecutive row:

In an open table, type **U** followed by the number of rows that you want to undo in the Command (cmd) column of the first row. For Example **U5** would undo the actions done to the row next to which you typed the command and the four following rows. Press the Enter key.

To undo actions in a range of rows:

In an open table, type **UU** in the Command (cmd) column of two rows, the row at the beginning and the row at the end of the range of rows where you want to undo actions. Press the Enter key.

Committing changes to DB2

DB2 Table Editor commits the changes that you have made automatically when you exit a table. When you exit a table that you have made changes to, a confirmation screen appears.

```
1 1 Save changes
  2 Discard changes

2 1 Abort on error
  2 Return to edit on error

COMMIT every 1 ____ (0-32767) rows
```

Field	Description
Save changes	Type 1 to commit the changes that you have made to DB2.
Discard changes	Type 2 to prevent the changes that you have made from being committed to DB2.
Abort on error	The DB2 transaction will be aborted if an error is encountered.
Return to edit on error	If an error is encountered, the editor will open and all uncommitted rows that contain a change will be displayed.
Commit everyx rows	Type the number of rows after which you would like DB2 to commit your changes. For example, if you type 5 in the Commit every x rows field, your changes will be committed every 5 rows. If you type 0, your changes will be committed at the end of the table.

Searching a table

You can search a table using the **find** command.

To search a table:

Find

►►—F—*text string*—◄◄

Note: to search for a text string that contains spaces, enclose the text string in single quote marks. (See example below).

1. On the command line, in an open table, type **F** followed by the text string that you want to find. If you are searching for a text string that includes spaces, enclose the string in single quotes.

For example, if you want to find the text "JENNIFER SMITH", type F 'JENNIFER SMITH' on the command line.

2. Press the Enter key.

DB2 Table Editor moves the cursor to the first occurrence of the text you typed and highlights the text in yellow.

Note: To find the next occurrence of the text that you specified press PF5.

Tip: Searching is case sensitive, so be sure to use the correct case when typing the text string that you want to search for.

To search a column in a table:

Find

►►—F—*text string*—'*column name or number*'——————►►

Note: to search for a text string that contains spaces, enclose the text string in single quote marks. (See example below).

1. On the command line, type **F** followed by the text string that you want to find followed by the name or number of the column in which you want to search. If you are searching for a text string that includes spaces, enclose the string in single quotes.

For example, if you want to find the text "JENNIFER SMITH" in the employee column, type F 'JENNIFER SMITH' employee on the command line.

Tip: Searching is case sensitive, so be sure to use the correct case when typing the text string that you want to search for.

2. Press the Enter key.
DB2 Table Editor moves the cursor to the first occurrence of the text that you typed and highlights the text in yellow.

Note: To find the next occurrence of the text that you specified press **PF5**.

Searching a table and changing the contents of cells in the table

You can search a table and replace the contents of a cell using the **change** command.

Change

►►—C—'*text string A*'⁽¹⁾—'*text string B*'⁽²⁾——————►►

Notes:

- 1 Where text string A is the text that you are searching for.
- 2 Where text string B is the text that will replace text string A.

You can change all occurrences of one text string to another text string with the **change all** command.

Change all

►►—C—'*text string A*'⁽¹⁾—'*text string B*'⁽²⁾—ALL——————►►

Notes:

- 1 Where text string A is the text that you are searching for.
- 2 Where text string B is the text that will replace text string A.

To search a table and change the contents of a cell:

1. On the command line, type **C**, followed by the text that you want to change, followed by the text that you want to change it to. If either text string contains spaces, enclose the text string in single quotes.

For example, if you want to find the text "JENNIFER SMITH" and change it to "JENNIFER JONES", type `C 'JENNIFER SMITH' 'JENNIFER JONES'` on the command line.

Tip: Searching is case sensitive, so be sure to use the correct case when typing the text string that you want to search for.

2. Press the Enter key.

DB2 Table Editor changes the first occurrence of the text that you specified and highlights the changed row in yellow.

Note: To find and change the next occurrence of the text that you specified, press PF6.

To search a table and change the contents of all cells that match the search criteria:

1. On the command line, type: **C**, followed by the text that you want to change, followed by the text that you want to change it to followed by **ALL**. If either text string contains spaces, enclose the text string in single quotes.

For example, if you want to find all of the instances of the text "4500" and change them to "5600", type `C '4500' '5600' ALL` on the command line.

Tip: Searching is case sensitive, so be sure to use the correct case when typing the text string that you want to search for.

2. Press the Enter key.

DB2 Table Editor changes all of the occurrences of the first text string that you specified to the second text string that you specified and highlights the changed rows in yellow.

Note: If a match is found in a truncated cell, (a cell that contains data that is too long to appear on the screen) the row will be highlighted and the cursor will be positioned at the beginning of the row. You can edit the cell by using the **zoom** command. For more information about using the zoom command see the following section: "Editing a long cell" on page 106.

To search a table and change the contents of all cells within a column that match the search criteria:

You can search for and replace the text that matches your search criteria within a specified column using the **change all in column** command.

Change all

►►—C—'*text string A*'⁽¹⁾—'*text string B*'⁽²⁾—ALL—'*column name or number*'—►►

Notes:

- 1 Where text string A is the text that you are searching for.
- 2 Where text string B is the text that will replace text string A.
1. On the command line, type: **C**, followed by the text that you want to change, followed by the text that you want to change it to followed by **ALL** followed by

the name or number of the column in which you want the changes to be made. If either text string contains spaces, enclose the text string in single quotes.

For example, if you want to find all of the instances of the text "4500" and change them to "5600" that occur in column 14, type C '4500' '5600' ALL 14 on the command line.

2. Press the Enter key.

DB2 Table Editor changes all of the occurrences of the first text string that you specified to the second text string that you specified in the specified column and highlights the changed rows in yellow.

Note: If a match is found in a truncated cell, (a cell that contains data that is too long to appear on the screen) the row will be highlighted and the cursor will be positioned at the beginning of the row. You can edit the cell by using the **zoom** command. For more information about using the zoom command see the following section: "Editing a long cell" on page 106.

Part 7. Appendixes

Appendix A. Troubleshooting

This section contains information on problems that can occur while using DB2 Table Editor and their solutions.

Troubleshooting for DB2 Table Editor Administrative Console Component, DB2 Table Editor Component and Java Player Component.

This section contains information on Frequently asked questions and common problems when using the DB2 Table Editor graphical user interface.

SDF (server definition file) does not appear in Server List

If your server does not appear in your server list, ensure that the correct .ini file is set as the default.

To change the default server definition file:

1. From the DB2 Table Editor component or the Java Player component main page, select View --> Options. The Options page opens.
2. From the Options page, click the Advanced button. The advanced version of the options page opens.
3. From the General page on the Options window, type the file name and location for the desired .ini file, or browse to the file by clicking the Browse (...) button.

Note: The default location for the server definition file on Windows NT and Windows 2000 is the WINNT directory.

4. Click OK. The advanced version of the Options window closes.
5. Click OK. The Options window closes.
6. The server defined in the .ini file now appears in your server list.

Installing DB2 Table Editor Java player as an applet

For information on installing DB2 table Editor Java Player as an applet, see "Installing DB2 Table Editor Java Player as an Applet" on page 68.

Appendix B. DB2 Table Editor ISPF Component Commands

Table 12. DB2 Table Editor Commands

Command	Function	Panel where available
EX &CLIST(ETI)	To start DB2 Table Editor, run the SETISAMP library member ETI.	N/A
S	To set up the configuration file, type S on the command line.	Main Panel
ABOUT	To display the IBM copyright notice, type ABOUT on the command line. This information is also displayed when you start the product the first time.	Main Panel
PF3	To return to the previous screen, press the PF3 key.	Main Panel
INCLUDE ALL	To select and re number all columns type INCLUDE ALL on the command line and the Select column will be renumbered. If two ore more columns has the same number, the order in which they appear on the screen will prevail.	Select Columns panel
EXCLUDE ALL	To have all columns excluded from the generated table, type EXCLUDE ALL on the command line and an N will appear in the Select column next to all rows.	Select Columns panel
E	Type E on the command line to edit the table in the Edit Table Rows panel.	Select Columns panel and Generated Select Statement panel
B	Type B on the command line to browse the table in the Edit Table Rows panel.	Select Columns panel and Generated Select Statement panel
F	Type F on the Cmd line of the row that you want to enter in the Edit Table Rows panel to edit the row in form format.	Edit Table Rows panel
Form	Type Form on the command line of the Edit Table Rows panel to edit the first row listed on the panel in form format. This allows you to edit rows one at a time.	Edit Table Rows panel
SQL	Type SQL on the command line of the Select Columns window to open the Generated Select Statement panel. From this panel you can edit the SQL	Select Columns panel
COUNT	Type COUNT on the command line to display a popup box indicating the number of rows that will be returned by your SQL statement.	Select Columns panel and Generated Select Statement panel
ZOOM, PF4, or HEX	To view or edit entire contents of a long cell, and additional information about the cell, type ZOOM or HEX on the command line or press the PF4 key. For more information on using this command see "Editing a long cell" on page 106	Edit Table Rows panel

Table 12. DB2 Table Editor Commands (continued)

Command	Function	Panel where available
HEX ON	In the Zoom Panel, to edit a row with a hex editor, type HEX ON on the command line.	ZOOM panel
HEX OFF	In the Zoom Panel, to turn the hex editor off, type HEX OFF on the command line.	ZOOM panel
CAPS ON	To force all changes to capital letters, type CAPS ON. on the command line.	Edit Table Rows panel and ZOOM panel
CAPS OFF	To resume entering data using uppercase and lowercase letters, type CAPS OFF on the command line.	Edit Table Rows panel and ZOOM panel
CHECK ON	To turn off checking (checking checks the screen for edit errors), type CHECK ON on the command line.	Edit Table Rows panel
CHECK OFF	To turn on checking (checking checks the screen for edit errors), type CHECK OFF on the command line.	Edit Table Rows panel
I	To insert a new row into the table that you are editing, type I in the Command column of the row below which you want to insert a new row. For more information on using this command, see "Inserting a blank row" on page 104	Edit Table Rows panel and Generated Select Statement panel
I<nn>	To insert more than one row into the table that you are editing, type I , followed by the number of new rows that you want to insert in the Command column of the row below which you want the new rows to appear.	Edit Table Rows panel and Generated Select Statement panel
R	To repeat a row, type R in the Command column of the row that you want to repeat."Repeating rows" on page 105	Edit Table Rows panel and Generated Select Statement panel
R<nn>	To insert a repeated row more than once, type R in the Command column of the row that you want to repeat, followed by the number of times that you want to repeat that row.	Edit Table Rows panel and Generated Select Statement panel
RR	To repeat a range of rows between two specified rows, type RR in the Command columns of two different rows.	Edit Table Rows panel and Generated Select Statement panel
C	To copy a row, type C in the Command column of the row that you want to copy, then type A in the Command column of the row above which you want the row to be inserted, or, type B in the Command column of the row below which you want the row to be inserted.	Edit Table Rows panel

Table 12. DB2 Table Editor Commands (continued)

Command	Function	Panel where available
C<nn>	To copy more than one row at a time, type C in the Command column of the row that you want to copy, followed by the number of rows that you want to copy, then type A in the Command column of the row above which you want the rows to be inserted, or, type B in the Command column of the rows below which you want the row to be inserted.	Edit Table Rows panel
CC	To copy all rows between two specified rows, type CC in the Command columns of two different rows, then type A in the Command column of the row above which you want the rows to be inserted, or, type B in the Command column of the row below which you want the rows to be inserted.	Edit Table Rows panel
D	To delete a row, type D in the Command column of the row that you want to delete. For more information on using this command, see "Deleting rows" on page 108	Edit Table Rows panel and Generated Select Statement panel
D<nn>	To delete more than one row, type D in the Command column of the row that you want to delete, followed by the number of rows that you want to delete.	Edit Table Rows panel and Generated Select Statement panel
DD	To delete all rows between two specified rows, type DD in the Command columns of two different rows to delete the range of rows in between those two rows.	Edit Table Rows panel and Generated Select Statement panel
U	To undo an action previously performed on a row, type U in the Command column. For more information on using this command, see "Undoing an action" on page 108	Edit Table Rows panel
U<nn>	Type U in the Command column of the row for which you want to undo an action, followed by the number of rows for which you want to undo actions.	Edit Table Rows panel
UU	Type UU in the Command columns of two different rows to undo the previous actions for the range of rows in between those two rows.	Edit Table Rows panel
F <text string>	Type F followed by the text that you want to search for, enclosed in quotes. For more information on using this command, see "Searching a table" on page 109	Edit Table Rows panel
F <text string> <column name or number>	To search for a text string within a specified column, type F , followed by the text string for which you want to search, followed by the column name or number in which you want to search.	Edit Table Rows panel
PF5	To find the next occurrence of the text string that you were searching for using the Find command, press the PF5 key.	Edit Table Rows panel

Table 12. DB2 Table Editor Commands (continued)

Command	Function	Panel where available
C <text string> <new text string>	To change one text string to a new value, type C followed by the text string that you want to replace, followed by the text string that you want to replace it with. For more information on using this command, see “Searching a table and changing the contents of cells in the table” on page 110	Edit Table Rows panel
PF6	To change the next occurrence of the text string that you were searching for using the Change command, press the PF6 key.	Edit Table Rows panel
C <text string> <new text string> ALL	To change all matching text strings to a new value, type C , followed by the text string that you want to replace, followed by the text string that you want to replace it with. For more information on using this command, see “To search a table and change the contents of all cells that match the search criteria.” on page 111.	Edit Table Rows panel
C <text string> <new text string> ALL <column name or number>	To change all matching text strings that are found in a specified column to a new value, type C followed by the text string that you want to replace, followed by the text string that you want to replace it with followed by ALL followed by the name or number of the column in which you want the changes to be made.	Edit Table Rows panel
CANcel	Type CANcel on the Command line to exit the Edit Table Rows panel without committing the changes that were made.	Edit Table Rows panel

Appendix C. The use of color in DB2 Table Editor for ISPF

Color is used in DB2 table editor to indicate a number of the state of rows and columns as outlined below:

Table 13. The use of color in DB2 Table Editor

Color	Meaning
Red	Indicates an error
Yellow	Indicates one of three things: <ul style="list-style-type: none">• The value has been updated• The row has been inserted• The row has been found using the found command.
Green	The row is marked for deletion
White	The column is truncated. (Truncated columns can be edited using the Zoom command (or pressing PF4))

Appendix D. Data Type Abbreviations Used in DB2 Table Editor for ISPF

To save space in DB2 Table Editor, abbreviations are used to represent data types. The following table contains the abbreviations and the long names for all of the data types used in DB2 Table Editor.

Data Type	Abbreviation
INTEGER	INT
SMALLINT	SMA
CHAR	CHR
DECIMAL	DEC
FLOAT	FLT
VARCHAR	VCH
LONGVAR	LVC
GRAPHIC	GRA
VARGRAPHIC	VGR
LONGVARG	LVG
DATE	DTE
TIME	TME
TIMESTAMP	TST
BLOB	BLB
CLOB	CLB
DBCLOB	DBC
ROWID	ROW
DISTINCT	DST

Appendix E. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10594-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is as your own risk.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

International Business Machines Corporation

Department J46A/G4
555 Bailey Avenue
San Jose, CA
U.S.A.
95141-1003

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information soft copy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX	iSeries
AS/400	NUMA-Q
DataJoiner	MVS
DB2	OS/2
DB2 Connect	OS/390
Distributed Relational Database Architecture	Redbooks
DRDA	SAA
IBM	SQL/DS
Informix	z/OS

Microsoft and WindowsNT are registered trademarks in the United State, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Index

A

- abbreviations, data types 123
- ABOUT, command 117
- accelerator keys, adding 55
- adding a control to a form 52
- adding, accelerator key 55
- advanced installation 15
- applet installation, DB2 Table Editor Java Player 68
- assigning users to a resource limits group 36
- attributes 21

B

- binding the DB2 Table Editor packages 29
- blank row, insert 104
- buttons, adding an accelerator key to 55

C

- C, command 118, 120
- Call Level Interface (CLI) 3
- capital letters, force changes to 118
- CAPS OFF, command 118
- CAPS ON, command 118
- cells in DB2 Table Editor, working with 79
- change all, command 110
- change, command 120
- changes, committing 109
- changing, password 36
- CHECK OFF, command 118
- CHECK ON, command 118
- checking, turn off 118
- checking, turn on 118
- CLI (Call Level Interface) 3
- CLI environment, configuring 9
- client installation 19
- closing a form 52
- color in DB2 Table Editor, use of 121
- Column Editor, panel 106
- columns in DB2 Table Editor, working with 78
- columns, deselect 117
- columns, select all 117
- command line parameters, DB2 Table Editor Java Player 71
- command, ABOUT 117
- command, C 118, 120
- command, CAPS OFF 118
- command, CAPS ON 118
- command, change 120
- command, change all 110
- command, CHECK OFF 118
- command, CHECK ON 118
- command, copy 118
- command, COUNT 117
- command, D 119
- command, delete 119
- command, EX & CLIST(ETI) 117

- command, exclude all 102
- command, EXCLUDE ALL 117
- command, F 119
- command, FIND 119
- command, hex off 107
- command, HEX OFF 118
- command, hex on 107
- command, HEX ON 118
- command, I 118
- command, include all 102
- command, INCLUDE ALL 117
- command, insert 118
- command, PF3 117
- command, PF4 117
- command, R 118
- command, repeat row 118
- command, S 117
- command, U 119
- command, undelete 119
- command, zoom 106
- command, ZOOM 117
- commands, DB2 Table Editor 117
- comments, how to send vii
- commit scope, specifying 51
- committing, changes 109
- communications 4
- concepts 21
- configuration file, setup 117
- configuring a server that has bound packages associated with it 37
- configuring DB2 Table Editor 27
- configuring DB2 Table Editor Java Player on the client machine 67
- configuring your CLI environment 9
- configuring your network environment 7
- configuring your SNA environment 7
- configuring your TCP/IP environment 8
- connection to the database server, testing 29
- connectivity, DB2 Table Editor 3
- console component, DB2 Table Editor 3
- console component, overview 25
- console component, starting 27
- Control Center plug in, overview 83
- Control Center plug in, using 83
- Control Center Plug-in, installing 85
- controls 21
- controls, adding to a form 52
- controls, deleting from a form 53
- copy, command 118
- copy, row 105, 118, 120
- copyright notice, displaying 117
- COUNT, command 117
- CPI-C and LU 6.2 4
- creating a validation group 58
- creating a validation rule 57
- creating, form by importing a panel from DBEDIT 48
- creating, forms 43
- creating, forms manually 45

- creating, forms using the Edit Table feature 75
- creating, forms using the Forms wizard 43
- creating, resource limits group 35
- creating, sample tables 26, 36
- creating, server 27
- customization summary, DB2 Table Editor 91
- customizing, DB2 Table Editor 91

D

- D, command 119
- data 21
- data sharing environment, setting up 97
- data types, abbreviations 123
- database 21
- database, saving to 79
- DB2 Table Editor Commands 117
- DB2 Table Editor component 3
- DB2 Table Editor component, opening 73
- DB2 Table Editor component, overview 65
- DB2 Table Editor component, working with 73
- DB2 Table Editor connectivity 3
- DB2 Table Editor Console component 3
- DB2 Table Editor Console component, overview 25
- DB2 Table Editor Console component, starting 27
- DB2 Table Editor Console component, using 27
- DB2 Table Editor Control Center plug in, overview 83
- DB2 Table Editor Control Center plug in, using 83
- DB2 Table Editor Control Center Plug-in, installing 85
- DB2 Table Editor Developer 3
- DB2 Table Editor Developer, overview 41
- DB2 Table Editor Developer, working with 43
- DB2 Table Editor Java Player 3
- DB2 Table Editor Java Player applet, installing 68
- DB2 Table Editor java player application, running 68
- DB2 Table Editor Java Player component, opening 73
- DB2 Table Editor Java Player component, working with 73
- DB2 Table Editor Java Player, installing 67
- DB2 Table Editor Java Player, installing and configuring on a client machine 67
- DB2 Table Editor, configuring 27
- DB2 Table Editor, customization summary 91
- DB2 Table Editor, customizing 91
- DB2 Table Editor, installing 15
- DB2 Table Editor, main panel 99
- DB2 Table Editor, overview 3
- DB2 Table Editor, working with 99
- DBEDIT, importing a panel 48
- delete, command 119
- delete, row 108, 119
- deleting a form 52
- deleting a validation group 59
- deleting a validation rule 58
- deleting, control from a form 53
- deleting, rows 77
- deselect, columns 117
- developer, DB2 Table Editor 3
- developer, overview 41
- developer, working with 43
- display, number of rows to be returned 103, 117

- displaying, IBM copyright notice 117
- Distributed Relational Database Architecture (DRDA) 4
- DRDA (Distributed Relational Database Architecture) 4

E

- Edit Related 76
- Edit Table feature, creating forms using 75
- edit table in ISPF component, locking in 103
- edit table Java and Windows components, locking in 59
- edit, contents of a long cell 117
- edit, long cell 106
- edit, row 106
- edit, row with hex editor 118
- edit, table 75, 103
- edit, with hex editor 107
- editing, table 75
- empty table, inserting a row into 105
- enabling DB2 Table Editor governing capabilities 35
- Enter DB2 System Parameters, panel 95
- EX &CLIST(ETI), command 117
- exclude all, command 102
- EXCLUDE ALL, command 117

F

- F, command 119
- file server, installation 18
- find, command 119
- form attributes, setting 50
- form mode 77
- form, creating by importing a panel from DBEDIT 48
- forms 21
- forms, adding controls 52
- forms, closing 52
- forms, creating 43
- forms, creating manually 45
- forms, creating using the Edit Table feature 75
- forms, deleting 52
- forms, deleting controls 53
- forms, full screen 21
- forms, opening 49, 74
- forms, printing 52
- forms, saving 51
- forms, working with 49
- full-screen forms 21

G

- governing, enabling 35
- governing, FAQ 32
- governing, profile use 33
- governing, resource limits group 33
- governing, resources controlled 32
- governing, schedule 34
- governing, setting up 32

H

hex editor, edit row with 118
hex editor, edit with 107
hex editor, turning off 118
hex off, command 107
HEX OFF, command 118
hex on, command 107
HEX ON, command 118

I

I, command 118
IBM copyright notice, displaying 117
IBM DB2 Table Editor, overview 3
include all, command 102
INCLUDE ALL, command 117
initial action, specifying 50
Initial focus, specifying 50
insert, blank row 104
insert, command 118
insert, row 118
inserting a row into an empty table 105
inserting, rows 77
installing and configuring DB2 Table Editor Java Player
on a client machine 67
installing DB2 Table Editor 15
installing DB2 Table Editor Java Player 67
installing, advanced 15
installing, client 19
installing, Control Center Plug-in 85
installing, DB2 Table Editor Control Center Plug-in 85
installing, file server 18

J

Java Player application, running 68
Java Player component, opening 73
Java Player component, working with 73
Java Player, applet installing 68
Java Player, DB2 Table Editor 3
Java Player, installing 67
Java Player, installing and configuring on a client
machine 67
Java Player, overview 65
join 21

L

large objects, working with 79
license use management keys, DB2 Table Editor ISPF
component 91
licensing, DB2 Table Editor ISPF component 91
LOBs, working with 79
locking in edit table, ISPF component 103
locking in edit table, Java and Windows
components 59
long cell, edit 106
long cell, edit or view contents 117
long cell, view or edit contents 117
LU 6.2 and CPI-C 4

LUM keys, DB2 Table Editor ISPF component 91

M

maintaining packages 25

N

network environment, configuring 7
Notices 125
number of rows to be returned, display 103, 117

O

open and restricted enrollment 34
opening, DB2 Table Editor for Windows 73
opening, DB2 Table Editor Java Player component 73
opening, form 49
opening, forms 74
opening, table 100
overview, DB2 Table Editor Console component 25
overview, DB2 Table Editor Control Center plug-in 83
overview, DB2 Table Editor Developer 41
overview, IBM DB2 Table Editor 3
overview, Java Player component and DB2 Table Editor
component 65

P

packages, binding 29
panel, Column Editor 106
panel, DB2 Table Editor main panel 99
panel, Enter DB2 System Parameters 95
panel, Select Columns 101
panel, Table Selection 101
panel, Update Parameters for DB2 Subsystem (option
1) 96
panel, Update Parameters for DB2 Subsystem (option
2) 97
password, changing 36
PF3, command 117
PF4, command 117
predefining settings, DB2 Table Editor Java Player 71
primary keys, working with 76
primary table 43
printing, form 52
properties 22

R

R, command 118
repeat row, command 118
repeat, row 105, 118
resource limits group 33, 34
resource limits group schedules, creating 36
resource limits group, assigning users 36
resource limits group, creating 35
restricted and open enrollment 34
return to the previous screen 117
roe, copy 105

- row, copy 118, 120
- row, delete 108, 119
- row, edit 106
- row, insert 118
- row, repeat 105, 118
- row, undelete 108, 119
- rows in DB2 Table Editor, working with 77
- rows to be returned, display number 103, 117
- rows, deleting 77
- rows, inserting 77
- rows, updating 78
- running DB2 Table Editor Java Player application 68
- running unattended sessions, DB2 Table Editor Java Player 71

S

- S, command 117
- sample tables, creating 26, 36
- saving, forms 51
- saving, to the database 79
- schedule 34
- schedules, resource limits group 36
- search and change, in a table 110
- search, a table 109
- select all, columns 117
- Select Columns, panel 101
- sending comments to IBM vii
- server, configuring with bound packages associated with it 37
- server, creating 27
- servers 25
- set up, configuration file 117
- setting general form attributes 50
- setting up governing 32
- SNA environment, configuring 7
- SNA networks, using 27
- special command characters, using 56
- specifying, commit scope 51
- specifying, initial action 50
- specifying, initial focus 50
- starting DB2 Table Editor 117
- starting, DB2 Table Editor Console component 27

T

- table layout mode 77
- Table Selection, panel 101
- table, edit 103
- table, editing 75
- table, opening 100
- table, search and change in 110
- table, searching a 109
- TCP/IP and WinSock 4
- TCP/IP environment, configuring 8
- testing connection to database server 29
- trademarks 127
- turn off, checking 118
- turn on, checking 118

U

- U, command 119
- undelete, command 119
- undelete, row 108, 119
- Update Parameters for DB2 Subsystem (option 1), panel 96
- Update Parameters for DB2 Subsystem (option 2), panel 97
- updating, rows 78
- using DB2 Table Editor Console component 27
- using SNA networks 27
- using, Control Center Plug-in 83
- using, DB2 Table Editor Control Center Plug-in 83

V

- validation group, creating 58
- validation group, deleting 59
- validation rules 22
- validation rules, creating 57
- validation rules, deleting 58
- validation rules, working with 57
- view contents of a long cell 117

W

- WinSock and TCP/IP 4
- working with cells in DB2 Table Editor 79
- working with columns in DB2 Table Editor 78
- working with DB2 Table Editor Developer 43
- working with forms 49
- working with LOBs 79
- working with primary keys 76
- working with rows in DB2 Table Editor 77
- working with the DB2 Table Editor component and the Java Player component 73
- working with validation rules 57
- working with, DB2 Table Editor 99

Z

- zoom, command 106
- ZOOM, command 117

Readers' Comments — We'd Like to Hear from You

IBM DB2 Table Editor
User's Guide
Version 4 Release 3

Publication No. SC27-1616-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

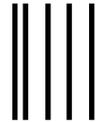
Phone No.



Fold and Tape

Please do not staple

Fold and Tape



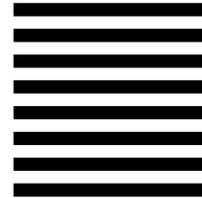
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
090/H150
555 Bailey Avenue
San Jose, CA
U.S.A.
95141-9989



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5724-B33

Printed in U.S.A.

SC27-1616-01

