OpenEdge Development:
Basic Database Tools
## Contents

Preface .............................................................................. Preface–1

1. **Data Dictionary Basics.** ........................................ 1–1
   Starting the Data Dictionary ..................................... 1–2
   Data Dictionary menu bar ..................................... 1–3

2. **Database Tasks** ......................................................... 2–1
   Creating a database .............................................. 2–2
   Connecting to a database ...................................... 2–3
   Switching to another connected database ............. 2–4
   Disconnecting a database ...................................... 2–5
   Viewing session information ................................ 2–6
   Dumping data definitions ...................................... 2–7
   Creating an incremental data definition file .......... 2–8
   Creating a parameter file ...................................... 2–9
   Editing a parameter file ........................................ 2–10
   Loading data definitions ...................................... 2–11

3. **Table Tasks** .............................................................. 3–1
   Creating a table ..................................................... 3–2
   Specifying validation criteria ............................... 3–2
   Specifying triggers .............................................. 3–2
   Specifying string attributes ................................. 3–3
   Modifying a table .................................................. 3–4
   Deleting a table ..................................................... 3–5
   Freezing or unfreezing a table .............................. 3–6
   Generating a table report ..................................... 3–7
   Generating a Detailed Table Report ...................... 3–7
   Generating a Quick Table Report ......................... 3–9
   Generating a Table Relations Report .................. 3–10
   Dumping table contents ....................................... 3–11
   Loading table contents ...................................... 3–13
   Reconstructing bad load records ......................... 3–14
## Contents

4. Field Tasks ................................................................. 4–1  
   Adding a field ......................................................... 4–2  
   Modifying a field ..................................................... 4–3  
   Changing a field name globally .................................... 4–5  
   Copying a field ......................................................... 4–6  
   Deleting a field ......................................................... 4–7  
   Changing field order .................................................. 4–8  
   Generating a Quick Field Report ................................. 4–9  
   Adjusting the Field Width ......................................... 4–10

5. Index Tasks ............................................................... 5–1  
   Creating an index ...................................................... 5–2  
   Modifying an index ..................................................... 5–4  
     Changing the primary index for a table ........................ 5–4  
     Renaming an index ................................................ 5–4  
     Deactivating an index ............................................. 5–5  
   Deleting an index ...................................................... 5–6  
   Generating a Quick Index Report ................................ 5–7

6. Sequence Tasks .......................................................... 6–1  
   Support for 64-bit sequences ....................................... 6–2  
   Creating a sequence .................................................. 6–3  
   Modifying a sequence ................................................ 6–4  
   Deleting a sequence ................................................... 6–5  
   Dumping sequence definitions ..................................... 6–6  
   Dumping sequence current values .................................. 6–7  
   Loading sequence current values ................................... 6–8  
   Generating a sequence report ....................................... 6–9

7. Trigger Tasks ............................................................... 7–1  
   Creating a trigger ....................................................... 7–2  
   Modifying a trigger .................................................... 7–4  
   Deleting a trigger ....................................................... 7–5  
   Generating a trigger report ......................................... 7–6

8. Database Menu Reference ................................................ 8–1  
   Database menu .......................................................... 8–2  
   Select Working Database option .................................... 8–3  
   Create option ............................................................ 8–4  
   Connect option .......................................................... 8–5  
   Disconnect option ....................................................... 8–7  
   Reports option .......................................................... 8–8  
     Detailed Table Report option ..................................... 8–9  
     Quick Table Report option ......................................... 8–10  
     Quick Field Report option ......................................... 8–10  
     Quick Index Report .................................................. 8–11  
     Sequence Report option ............................................ 8–12  
     Trigger Report option ............................................... 8–12
9. **Schema Menu Reference** ................................................................. 9–1
   Schema menu ........................................................................... 9–2
   Modify Table option ................................................................. 9–3
      Validation option .................................................................... 9–4
      Table Triggers option ............................................................ 9–5
      String Attributes option ......................................................... 9–7
   Add New Table option ............................................................... 9–9
   Delete Table(s) option ............................................................... 9–10
   Field Editor option .................................................................. 9–11
      Add ........................................................................................... 9–12
      Field Triggers ......................................................................... 9–25
      View As ................................................................................... 9–26
      GoIndex option ...................................................................... 9–27
   Reorder Fields option ............................................................... 9–28
   Global Field Name Change option ........................................... 9–29
   Index Editor option ................................................................... 9–30
   Sequence Editor option ............................................................ 9–34
   Adjust Field Width option ......................................................... 9–36

10. **Admin Menu Reference** .............................................................. 10–1
    Admin menu ............................................................................ 10–2
    Dump Data and Definitions option ........................................... 10–3
       Data Definitions (.df file) option ............................................ 10–4
       Table Contents (.d file) option .............................................. 10–4
       SQL Views option ................................................................ 10–5
       User Table Contents option ................................................ 10–5
       Security Authentication Records option ............................... 10–5
       Security Permissions option .................................................. 10–6
       Auto-Connect Records Only option ...................................... 10–7
       Collation Tables option .......................................................... 10–7
       Sequences Definitions option ............................................... 10–7
       Sequences Current Values option ......................................... 10–7
       Database Identification Properties option ............................ 10–8
       Database Options ................................................................. 10–8
       Audit Policies option ............................................................. 10–9
       Audit Data option ................................................................. 10–10
       Create Incremental .df File option ........................................ 10–12
    Load Data and Definitions option ............................................. 10–14
       Data Definitions (.df file) option ............................................ 10–15
       Table Contents (.d file) option .............................................. 10–16
       SQL Views option ................................................................ 10–17
       User Table Contents option ................................................ 10–17
       Security Authentication Records option ............................... 10–17
       Security Permissions option .................................................. 10–18
       Sequences Current Values option ......................................... 10–18
       Reconstruct Bad Load Records option .................................. 10–18
       Database Identification Properties option ............................ 10–19
<table>
<thead>
<tr>
<th>Figures</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1–1:</td>
<td>Data Dictionary window</td>
<td>1–2</td>
</tr>
<tr>
<td>Figure 1–2:</td>
<td>Data Dictionary menu bar</td>
<td>1–3</td>
</tr>
<tr>
<td>Figure 8–1:</td>
<td>Select Working Database dialog box</td>
<td>8–3</td>
</tr>
<tr>
<td>Figure 8–2:</td>
<td>Create Database dialog box</td>
<td>8–4</td>
</tr>
<tr>
<td>Figure 8–3:</td>
<td>Connect Database dialog box</td>
<td>8–5</td>
</tr>
<tr>
<td>Figure 8–4:</td>
<td>Report Options dialog box</td>
<td>8–9</td>
</tr>
<tr>
<td>Figure 8–5:</td>
<td>Detailed Table Report dialog box</td>
<td>8–9</td>
</tr>
<tr>
<td>Figure 8–6:</td>
<td>Quick Table Report dialog box</td>
<td>8–10</td>
</tr>
<tr>
<td>Figure 8–7:</td>
<td>Quick Field Report dialog box</td>
<td>8–10</td>
</tr>
<tr>
<td>Figure 8–8:</td>
<td>Quick Index Report dialog box</td>
<td>8–11</td>
</tr>
<tr>
<td>Figure 8–9:</td>
<td>Sequence Report dialog box</td>
<td>8–12</td>
</tr>
<tr>
<td>Figure 8–10:</td>
<td>Trigger Report dialog box</td>
<td>8–12</td>
</tr>
<tr>
<td>Figure 8–11:</td>
<td>User Report dialog box</td>
<td>8–13</td>
</tr>
<tr>
<td>Figure 8–12:</td>
<td>Table Relations Report dialog box</td>
<td>8–13</td>
</tr>
<tr>
<td>Figure 8–13:</td>
<td>Quick Storage Area Report dialog box</td>
<td>8–14</td>
</tr>
<tr>
<td>Figure 8–14:</td>
<td>Data Width Report dialog box</td>
<td>8–14</td>
</tr>
<tr>
<td>Figure 8–15:</td>
<td>Quick Alternate Buffer Pool Report</td>
<td>8–15</td>
</tr>
<tr>
<td>Figure 8–16:</td>
<td>Quick Encryption Policy Report dialog box</td>
<td>8–24</td>
</tr>
<tr>
<td>Figure 8–17:</td>
<td>Detailed Encryption Policy Report dialog box</td>
<td>8–25</td>
</tr>
<tr>
<td>Figure 9–1:</td>
<td>Modify Table window</td>
<td>9–3</td>
</tr>
<tr>
<td>Figure 9–2:</td>
<td>Table Validation dialog box</td>
<td>9–4</td>
</tr>
<tr>
<td>Figure 9–3:</td>
<td>Table Triggers dialog box</td>
<td>9–5</td>
</tr>
<tr>
<td>Figure 9–4:</td>
<td>Example of a delete trigger procedure</td>
<td>9–6</td>
</tr>
<tr>
<td>Figure 9–5:</td>
<td>Table String Attributes dialog box</td>
<td>9–7</td>
</tr>
<tr>
<td>Figure 9–6:</td>
<td>Add New Table window</td>
<td>9–9</td>
</tr>
<tr>
<td>Figure 9–7:</td>
<td>Field Editor window</td>
<td>9–11</td>
</tr>
<tr>
<td>Figure 9–8:</td>
<td>Field Editor Add option window</td>
<td>9–12</td>
</tr>
<tr>
<td>Figure 9–9:</td>
<td>Blob Field Attributes dialog box</td>
<td>9–22</td>
</tr>
<tr>
<td>Figure 9–10:</td>
<td>Clob Field Attributes dialog box</td>
<td>9–23</td>
</tr>
<tr>
<td>Figure 9–11:</td>
<td>Field Triggers dialog box</td>
<td>9–25</td>
</tr>
<tr>
<td>Figure 9–12:</td>
<td>View-As Phrase dialog box</td>
<td>9–26</td>
</tr>
<tr>
<td>Figure 9–13:</td>
<td>GoIndex option window</td>
<td>9–27</td>
</tr>
<tr>
<td>Figure 9–14:</td>
<td>Resequence Order of Fields dialog box</td>
<td>9–28</td>
</tr>
<tr>
<td>Figure 9–15:</td>
<td>Global Field Name Change dialog box</td>
<td>9–29</td>
</tr>
<tr>
<td>Figure 9–16:</td>
<td>Index Editor window</td>
<td>9–30</td>
</tr>
<tr>
<td>Figure 9–17:</td>
<td>Adding Index dialog box</td>
<td>9–33</td>
</tr>
<tr>
<td>Figure 9–18:</td>
<td>Sample Sequence Editor window</td>
<td>9–34</td>
</tr>
<tr>
<td>Figure 10–1:</td>
<td>Dump Data Contents for Some Tables dialog box</td>
<td>10–4</td>
</tr>
<tr>
<td>Figure 10–2:</td>
<td>Create Incremental .df File window</td>
<td>10–12</td>
</tr>
<tr>
<td>Figure 10–3:</td>
<td>Load Data Definitions dialog box</td>
<td>10–15</td>
</tr>
<tr>
<td>Figure 10–4:</td>
<td>Load Data Contents for Some Tables dialog box</td>
<td>10–16</td>
</tr>
<tr>
<td>Figure 10–5:</td>
<td>Reconstruct Bad Load Records dialog box</td>
<td>10–18</td>
</tr>
<tr>
<td>Figure 10–6:</td>
<td>Edit User List dialog box</td>
<td>10–25</td>
</tr>
<tr>
<td>Figure 10–7:</td>
<td>Edit Data Security dialog box</td>
<td>10–26</td>
</tr>
<tr>
<td>Figure 10–8:</td>
<td>Security Administrators dialog box</td>
<td>10–29</td>
</tr>
<tr>
<td>Figure 10–9:</td>
<td>User Report dialog box</td>
<td>10–30</td>
</tr>
<tr>
<td>Figure 10–10:</td>
<td>Export Text dialog box</td>
<td>10–39</td>
</tr>
<tr>
<td>Figure 10–11:</td>
<td>Output Record Formatting dialog box</td>
<td>10–40</td>
</tr>
<tr>
<td>Figure 10–12:</td>
<td>Import DIF dialog box</td>
<td>10–42</td>
</tr>
<tr>
<td>Figure 10–13:</td>
<td>Import Fixed-Length dialog box</td>
<td>10–43</td>
</tr>
<tr>
<td>Figure 10–14:</td>
<td>Fixed-Length Field Columns dialog box</td>
<td>10–44</td>
</tr>
<tr>
<td>Figure 10–15:</td>
<td>Import dBASE II, III, III+ and IV Definitions dialog box</td>
<td>10–44</td>
</tr>
<tr>
<td>Figure 10–16:</td>
<td>Alternate Buffer Pool Maintenance dialog box</td>
<td>10–49</td>
</tr>
<tr>
<td>Figure 10–17:</td>
<td>Alternate Buffer Pool Maintenance Copy Current Setting To dialog box</td>
<td>10–49</td>
</tr>
<tr>
<td>Figure 11–1:</td>
<td>Edit Parameter File dialog box</td>
<td>11–3</td>
</tr>
<tr>
<td>Figure 11–2:</td>
<td>Parameter File Comments dialog box</td>
<td>11–4</td>
</tr>
<tr>
<td>Figure 11–3:</td>
<td>Quote Entire Lines dialog box</td>
<td>11–5</td>
</tr>
<tr>
<td>Figure 11–4:</td>
<td>Quote By Delimiter dialog box</td>
<td>11–5</td>
</tr>
<tr>
<td>Figure 11–5:</td>
<td>Quote By Column Ranges dialog box</td>
<td>11–6</td>
</tr>
<tr>
<td>Figure 11–6:</td>
<td>ASSIGN statement example</td>
<td>11–7</td>
</tr>
<tr>
<td>Figure 11–7:</td>
<td>Generate FORM for “Customer” dialog box</td>
<td>11–7</td>
</tr>
<tr>
<td>Figure 11–8:</td>
<td>FORM statement example</td>
<td>11–9</td>
</tr>
<tr>
<td>Figure 11–9:</td>
<td>DEFINE WORK-TABLE statement example</td>
<td>11–9</td>
</tr>
<tr>
<td>Figure 11–10:</td>
<td>Edit OpenEdge Auto-Connect List dialog box</td>
<td>11–10</td>
</tr>
<tr>
<td>Figure 11–11:</td>
<td>Freeze/Unfreeze Table dialog box</td>
<td>11–11</td>
</tr>
<tr>
<td>Figure 11–12:</td>
<td>Index Deactivation Warning dialog box</td>
<td>11–12</td>
</tr>
<tr>
<td>Figure 11–13:</td>
<td>View Report dialog box</td>
<td>11–13</td>
</tr>
<tr>
<td>Figure 11–14:</td>
<td>Dump CREATE VIEW of All Views dialog box</td>
<td>11–15</td>
</tr>
<tr>
<td>Figure 11–15:</td>
<td>Dump CREATE TABLE dialog box</td>
<td>11–16</td>
</tr>
</tbody>
</table>
## Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1–1:</td>
<td>Data Dictionary menu</td>
<td>1–3</td>
</tr>
<tr>
<td>Table 1–2:</td>
<td>Options available without a connected database</td>
<td>1–4</td>
</tr>
<tr>
<td>Table 8–1:</td>
<td>Database menu</td>
<td>8–2</td>
</tr>
<tr>
<td>Table 8–2:</td>
<td>Reports menu</td>
<td>8–8</td>
</tr>
<tr>
<td>Table 8–3:</td>
<td>Auditing reports</td>
<td>8–16</td>
</tr>
<tr>
<td>Table 8–4:</td>
<td>Encryption Policy reports</td>
<td>8–23</td>
</tr>
<tr>
<td>Table 9–1:</td>
<td>Schema menu</td>
<td>9–2</td>
</tr>
<tr>
<td>Table 9–2:</td>
<td>Table Trigger events</td>
<td>9–5</td>
</tr>
<tr>
<td>Table 9–3:</td>
<td>String attribute examples</td>
<td>9–8</td>
</tr>
<tr>
<td>Table 9–4:</td>
<td>Field Editor options</td>
<td>9–11</td>
</tr>
<tr>
<td>Table 9–5:</td>
<td>Character format symbols</td>
<td>9–13</td>
</tr>
<tr>
<td>Table 9–6:</td>
<td>Character display format examples</td>
<td>9–13</td>
</tr>
<tr>
<td>Table 9–7:</td>
<td>Decimal and integer format characters</td>
<td>9–14</td>
</tr>
<tr>
<td>Table 9–8:</td>
<td>Numeric display format examples</td>
<td>9–15</td>
</tr>
<tr>
<td>Table 9–9:</td>
<td>Date display format examples</td>
<td>9–17</td>
</tr>
<tr>
<td>Table 9–10:</td>
<td>DATETIME display format examples</td>
<td>9–17</td>
</tr>
<tr>
<td>Table 9–11:</td>
<td>DATETIME-TZ display format example</td>
<td>9–17</td>
</tr>
<tr>
<td>Table 9–12:</td>
<td>Field label examples</td>
<td>9–18</td>
</tr>
<tr>
<td>Table 9–13:</td>
<td>Logical display format examples</td>
<td>9–18</td>
</tr>
<tr>
<td>Table 9–14:</td>
<td>Field initial values</td>
<td>9–19</td>
</tr>
<tr>
<td>Table 9–15:</td>
<td>Data types</td>
<td>9–20</td>
</tr>
<tr>
<td>Table 9–16:</td>
<td>View As widgets</td>
<td>9–27</td>
</tr>
<tr>
<td>Table 9–17:</td>
<td>Index Editor options</td>
<td>9–32</td>
</tr>
<tr>
<td>Table 9–18:</td>
<td>Sequence Editor options</td>
<td>9–35</td>
</tr>
<tr>
<td>Table 10–1:</td>
<td>Admin menu</td>
<td>10–2</td>
</tr>
<tr>
<td>Table 10–2:</td>
<td>Dump Data and Definitions menu</td>
<td>10–3</td>
</tr>
<tr>
<td>Table 10–3:</td>
<td>Load Data and Definitions menu</td>
<td>10–14</td>
</tr>
<tr>
<td>Table 10–4:</td>
<td>Database Identification menu</td>
<td>10–22</td>
</tr>
<tr>
<td>Table 10–5:</td>
<td>Security menu</td>
<td>10–24</td>
</tr>
<tr>
<td>Table 10–6:</td>
<td>Edit User List dialog box options</td>
<td>10–25</td>
</tr>
<tr>
<td>Table 10–7:</td>
<td>Access restrictions for tables</td>
<td>10–27</td>
</tr>
<tr>
<td>Table 10–8:</td>
<td>Edit Data Security dialog box options</td>
<td>10–27</td>
</tr>
<tr>
<td>Table 10–9:</td>
<td>Authentication Systems dialog box hot keys</td>
<td>10–31</td>
</tr>
<tr>
<td>Table 10–10:</td>
<td>Authentication System Domains dialog box hot keys</td>
<td>10–33</td>
</tr>
<tr>
<td>Table 10–11:</td>
<td>Export Data menu</td>
<td>10–39</td>
</tr>
<tr>
<td>Table 10–12:</td>
<td>Export format file extensions</td>
<td>10–40</td>
</tr>
<tr>
<td>Table 10–13:</td>
<td>Import Data menu</td>
<td>10–42</td>
</tr>
<tr>
<td>Table 11–1:</td>
<td>DataServer menu</td>
<td>11–2</td>
</tr>
<tr>
<td>Table 11–2:</td>
<td>Utilities menu</td>
<td>11–3</td>
</tr>
<tr>
<td>Table 11–3:</td>
<td>Quoter Functions menu</td>
<td>11–4</td>
</tr>
<tr>
<td>Table 11–4:</td>
<td>Generate Include Files menu</td>
<td>11–6</td>
</tr>
<tr>
<td>Table 11–5:</td>
<td>Edit OpenEdge Auto-Connect List options</td>
<td>11–11</td>
</tr>
<tr>
<td>Table 11–6:</td>
<td>PRO/SQL menu</td>
<td>11–13</td>
</tr>
</tbody>
</table>
Preface

This Preface contains the following sections:

• Purpose
• Audience
• Organization
• Using this manual
• Typographical conventions
• Examples of syntax descriptions
• OpenEdge messages
• Third party acknowledgements
Purpose

This book is a user guide for the OpenEdge® RDBMS basic database administration toolset for character-mode.

Audience

This book is intended for developers who want to use the character-mode Data Dictionary to manage the OpenEdge RDBMS.

Organization

Chapter 1, “Data Dictionary Basics”
Describes how to access the Data Dictionary and its menu options.

Chapter 2, “Database Tasks”
Describes how to perform database tasks.

Chapter 3, “Table Tasks”
Describes how to perform administrative tasks for tables and files.

Chapter 4, “Field Tasks”
Describes how to perform tasks to create and maintain fields.

Chapter 5, “Index Tasks”
Describes how to perform tasks to create and maintain indexes.

Chapter 6, “Sequence Tasks”
Describes how to perform tasks to create and maintain sequences.

Chapter 7, “Trigger Tasks”
Describes how to perform tasks to create and maintain schema triggers.
Chapter 8, “Database Menu Reference”
Describes the Database menu options and dialog boxes.

Chapter 9, “Schema Menu Reference”
Describes the Schema menu options and dialog boxes.

Chapter 10, “Admin Menu Reference”
Describes the Admin menu options and dialog boxes.

Chapter 11, “Advanced Menus Reference”
Describes the menu options and dialog boxes for the Utilities and PRO/SQL menus.

Using this manual

OpenEdge provides a special purpose programming language for building business applications. In the documentation, the formal name for this language is ABL (Advanced Business Language). With few exceptions, all keywords of the language appear in all UPPERCASE, using a font that is appropriate to the context. All other alphabetic language content appears in mixed case.

For the latest documentation updates see the OpenEdge Product Documentation Overview page on PSDN: http://communities.progress.com/pcom/docs/DOC-16074.

References to ABL compiler and run-time features

ABL is both a compiled and an interpreted language that executes in a run-time engine. The documentation refers to this run-time engine as the ABL Virtual Machine (AVM). When the documentation refers to ABL source code compilation, it specifies ABL or the compiler as the actor that manages compile-time features of the language. When the documentation refers to run-time behavior in an executing ABL program, it specifies the AVM as the actor that manages the specified run-time behavior in the program.

For example, these sentences refer to the ABL compiler’s allowance for parameter passing and the AVM’s possible response to that parameter passing at run time: “ABL allows you to pass a dynamic temp-table handle as a static temp-table parameter of a method. However, if at run time the passed dynamic temp-table schema does not match the schema of the static temp-table parameter, the AVM raises an error.” The following sentence refers to run-time actions that the AVM can perform using a particular ABL feature: “The ABL socket object handle allows the AVM to connect with other ABL and non-ABL sessions using TCP/IP sockets.”
Abril 11, 2008

Preface

References to ABL data types

ABL provides built-in data types, built-in class data types, and user-defined class data types. References to built-in data types follow these rules:

- Like most other keywords, references to specific built-in data types appear in all UPPERCASE, using a font that is appropriate to the context. No uppercase reference ever includes or implies any data type other than itself.
- Wherever integer appears, this is a reference to the INTEGER or INT64 data type.
- Wherever character appears, this is a reference to the CHARACTER, LONGCHAR, or CLOB data type.
- Wherever decimal appears, this is a reference to the DECIMAL data type.
- Wherever numeric appears, this is a reference to the INTEGER, INT64, or DECIMAL data type.

References to built-in class data types appear in mixed case with initial caps, for example, Progress.Lang.Object. References to user-defined class data types appear in mixed case, as specified for a given application example.

Typographical conventions

This manual uses the following typographical conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold</strong></td>
<td>Bold typeface indicates commands or characters the user types, provides emphasis, or the names of user interface elements.</td>
</tr>
<tr>
<td><em>Italic</em></td>
<td>Italic typeface indicates the title of a document, or signifies new terms.</td>
</tr>
<tr>
<td>SMALL, BOLD CAPITAL LETTERS</td>
<td>Small, bold capital letters indicate OpenEdge key functions and generic keyboard keys; for example, GET and CTRL.</td>
</tr>
<tr>
<td>KEY1+KEY2</td>
<td>A plus sign between key names indicates a simultaneous key sequence: you press and hold down the first key while pressing the second key. For example, CTRL+X.</td>
</tr>
<tr>
<td>KEY1 KEY2</td>
<td>A space between key names indicates a sequential key sequence: you press and release the first key, then press another key. For example, ESCAPE H.</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
</tr>
<tr>
<td>Fixed width</td>
<td>A fixed-width font is used in syntax statements, code examples, system output, and filenames.</td>
</tr>
<tr>
<td>Fixed-width italics</td>
<td>Fixed-width italics indicate variables in syntax statements.</td>
</tr>
<tr>
<td>Fixed-width bold</td>
<td>Fixed-width bold indicates variables with special emphasis.</td>
</tr>
</tbody>
</table>
### Examples of syntax descriptions

In this example, ACCUM is a keyword, and `aggregate` and `expression` are variables:

**Syntax**

```
ACCUM aggregate expression
```

FOR is one of the statements that can end with either a period or a colon, as in this example:

```
FOR EACH Customer NO-LOCK:
    DISPLAY Customer.Name.
END.
```

In this example, STREAM `stream`, UNLESS-HIDDEN, and NO-ERROR are optional:

**Syntax**

```
DISPLAY [ STREAM stream ] [ UNLESS-HIDDEN ] [ NO-ERROR ]
```
In this example, the outer (small) brackets are part of the language, and the inner (large) brackets denote an optional item:

**Syntax**

```
INITIAL [ constant [ , constant ] ]
```

A called external procedure must use braces when referencing compile-time arguments passed by a calling procedure, as shown in this example:

**Syntax**

```
{ &argument-name }
```

In this example, EACH, FIRST, and LAST are optional, but you can choose only one of them:

**Syntax**

```
PRESELECT [ EACH | FIRST | LAST ] record-phrase
```

In this example, you must include two expressions, and optionally you can include more. Multiple expressions are separated by commas:

**Syntax**

```
MAXIMUM ( expression , expression [ , expression ] ... )
```

In this example, you must specify MESSAGE and at least one expression or SKIP [ (n) ], and any number of additional expression or SKIP [ ( n ) ] is allowed:

**Syntax**

```
MESSAGE { expression | SKIP [ ( n ) ] } ... 
```

In this example, you must specify {include-file, then optionally any number of argument or &argument-name = "argument-value", and then terminate with }:

**Syntax**

```
{ include-file
  [ argument | &argument-name = "argument-value" ] ... }
```
Long syntax descriptions split across lines

Some syntax descriptions are too long to fit on one line. When syntax descriptions are split across multiple lines, groups of optional and groups of required items are kept together in the required order.

In this example, WITH is followed by six optional items:

Syntax

WITH [ ACCUM max-length ] [ expression DOWN ]
[ CENTERED ] [ n COLUMNS ] [ SIDE-LABELS ]
[ STREAM-IO ]

Complex syntax descriptions with both required and optional elements

Some syntax descriptions are too complex to distinguish required and optional elements by bracketing only the optional elements. For such syntax, the descriptions include both braces (for required elements) and brackets (for optional elements).

In this example, ASSIGN requires either one or more field entries or one record. Options available with field or record are grouped with braces and brackets:

Syntax

ASSIGN  { [ FRAME frame ] { field [ = expression ] } }
       [ WHEN expression ]} ... 
       | { record [ EXCEPT field ... ] }

OpenEdge messages

OpenEdge displays several types of messages to inform you of routine and unusual occurrences:

- **Execution messages** inform you of errors encountered while OpenEdge is running a procedure; for example, if OpenEdge cannot find a record with a specified index field value.
- **Compile messages** inform you of errors found while OpenEdge is reading and analyzing a procedure before running it; for example, if a procedure references a table name that is not defined in the database.
- **Startup messages** inform you of unusual conditions detected while OpenEdge is getting ready to execute; for example, if you entered an invalid startup parameter.
After displaying a message, OpenEdge proceeds in one of several ways:

- Continues execution, subject to the error-processing actions that you specify or that are assumed as part of the procedure. This is the most common action taken after execution messages.
- Returns to the Procedure Editor, so you can correct an error in a procedure. This is the usual action taken after compiler messages.
- Halts processing of a procedure and returns immediately to the Procedure Editor. This does not happen often.
- Terminates the current session.

OpenEdge messages end with a message number in parentheses. In this example, the message number is 200:

```
** Unknown table name table. (200)
```

If you encounter an error that terminates OpenEdge, note the message number before restarting.

**Obtaining more information about OpenEdge messages**

In Windows platforms, use OpenEdge online help to obtain more information about OpenEdge messages. Many OpenEdge tools include the following Help menu options to provide information about messages:

- Choose Help → Recent Messages to display detailed descriptions of the most recent OpenEdge message and all other messages returned in the current session.
- Choose Help → Messages and then type the message number to display a description of a specific OpenEdge message.
- In the Procedure Editor, press the HELP key or F1.

On UNIX platforms, use the OpenEdge pro command to start a single-user mode character OpenEdge client session and view a brief description of a message by providing its number.

To use the pro command to obtain a message description by message number:

1. Start the Procedure Editor:

```
OpenEdge-install-dir/bin/pro
```

2. Press F3 to access the menu bar, then choose Help → Messages.
3. Type the message number and press ENTER. Details about that message number appear.
4. Press F4 to close the message, press F3 to access the Procedure Editor menu, and choose File → Exit.
Third party acknowledgements

OpenEdge includes AdventNet - Agent Toolkit licensed from AdventNet, Inc.
http://www.adventnet.com. All rights to such copyright material rest with AdventNet.

OpenEdge includes ANTLR (Another Tool for Language Recognition) software Copyright © 2003-2006, Terence Parr All rights reserved. Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

OpenEdge includes software developed by the Apache Software Foundation (http://www.apache.org/). Copyright © 1999 The Apache Software Foundation. All rights reserved (Xerces C++ Parser (XML) and Xerces2 Java Parser (XML)); Copyright © 1999-2002 The Apache Software Foundation. All rights reserved (Xerces Parser (XML); and Copyright © 2000-2003 The Apache Software Foundation. All rights reserved (Ant). The names “Apache,” “Xerces,” “ANT,” and “Apache Software Foundation” must not be used to endorse or promote products derived from this software without prior written permission. Products derived from this software may not be called “Apache”, nor may “Apache” appear in their name, without prior written permission of the Apache Software Foundation. For written permission, please contact apache@apache.org. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

OpenEdge includes Concurrent Java software Copyright 1994-2000 Sun Microsystems, Inc. All Rights Reserved. -Neither the name of or trademarks of Sun may be used to endorse or promote products including or derived from the Java Software technology without specific prior written permission; and Redistributions of source or binary code must contain the above copyright notice, this notice and the following disclaimers: This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MICROSYSTEMS, INC. AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN MICROSYSTEMS, INC. OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN MICROSYSTEMS, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

OpenEdge includes DataDirect software Copyright © 1991-2007 Progress Software Corporation and/or its subsidiaries or affiliates. All Rights Reserved. (DataDirect Connect for JDBC Type 4 driver); Copyright © 1993-2009 Progress Software Corporation and/or its subsidiaries or affiliates. All Rights Reserved. (DataDirect Connect for JDBC); Copyright © 1988-2007 Progress Software Corporation and/or its subsidiaries or affiliates. All Rights Reserved. (DataDirect Connect for ODBC); and Copyright © 1988-2007 Progress Software
Preface

Corporation and/or its subsidiaries or affiliates. All Rights Reserved. (DataDirect Connect64 for ODBC).

OpenEdge includes DataDirect Connect for ODBC and DataDirect Connect64 for ODBC software, which include ICU software 1.8 and later - Copyright © 1995-2003 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

OpenEdge includes DataDirect Connect for ODBC and DataDirect Connect64 for ODBC software, which include software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/). Copyright © 1998-2006 The OpenSSL Project. All rights reserved. And Copyright © 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved.

OpenEdge includes DataDirect products for the Microsoft SQL Server database which contain a licensed implementation of the Microsoft TDS Protocol.

OpenEdge includes software authored by David M. Gay. Copyright © 1991, 2000, 2001 by Lucent Technologies (dtoa.c); Copyright © 1991, 1996 by Lucent Technologies (g_fmt.c); and Copyright © 1991 by Lucent Technologies (rnd_prod.s). Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

OpenEdge includes software authored by David M. Gay. Copyright © 1998-2001 by Lucent Technologies All Rights Reserved (decsrttod.c; strtodg.c); Copyright © 1998, 2000 by Lucent Technologies All Rights Reserved (decsrttodf.c; strtord.c); Copyright © 1998 by Lucent Technologies All Rights Reserved (dmisc.c; gdtoa.h; gethex.c; gmisc.c; sum.c); Copyright © 1998, 1999 by Lucent Technologies All Rights Reserved (gdtoa.c; misc.c; smisc.c; ulp.c); Copyright © 1998-2000 by Lucent Technologies All Rights Reserved (gdtoaimp.h); Copyright © 2000 by Lucent Technologies All Rights Reserved (hd_init.c). Full copies of these licenses can be found in the installation directory, in the c:/OpenEdge/licenses folder. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
OpenEdge includes http package software developed by the World Wide Web Consortium. Copyright © 1994-2002 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All rights reserved. This work is distributed under the W3C® Software License [http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231] in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

OpenEdge includes ICU software 1.8 and later - Copyright © 1995-2003 International Business Machines Corporation and others All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.


OpenEdge includes Infragistics NetAdvantage for .NET v2009 Vol 2 Copyright © 1996-2009 Infragistics, Inc. All rights reserved.

OpenEdge includes JSTL software Copyright 1994-2006 Sun Microsystems, Inc. All Rights Reserved. Software distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

OpenEdge includes OpenSSL software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/). Copyright © 1998-2007 The OpenSSL Project. All rights reserved. This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com). Copyright © 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project. Software distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

OpenEdge includes Quartz Enterprise Job Scheduler software Copyright © 2001-2003 James House. All rights reserved. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product. This product uses and includes within its distribution, software developed by the Apache Software Foundation (http://www.apache.org/).

OpenEdge includes code licensed from RSA Security, Inc. Some portions licensed from IBM are available at http://oss.software.ibm.com/icu4j/.

OpenEdge includes the RSA Data Security, Inc. MD5 Message-Digest Algorithm. Copyright ©1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.
OpenEdge includes Sonic software, which includes software developed by Apache Software Foundation (http://www.apache.org/). Copyright © 1999-2000 The Apache Software Foundation. All rights reserved. The names “Ant”, “Axis”, “Xalan,” “FOP,” “The Jakarta Project”, “Tomcat”, “Xerces” and/or “Apache Software Foundation” must not be used to endorse or promote products derived from the Product without prior written permission. Any product derived from the Product may not be called “Apache”, nor may “Apache” appear in their name, without prior written permission. For written permission, please contact apache@apache.org.

OpenEdge includes Sonic software, which includes software Copyright © 1999 CERN - European Organization for Nuclear Research. Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. CERN makes no representations about the suitability of this software for any purpose. It is provided "as is" without expressed or implied warranty.

OpenEdge includes Sonic software, which includes software developed by ExoLab Project (http://www.exolab.org/). Copyright © 2000 Intalio Inc. All rights reserved. The names “Castor” and/or “ExoLab” must not be used to endorse or promote products derived from the Products without prior written permission. For written permission, please contact info@exolab.org. Exolab, Castor and Intalio are trademarks of Intalio Inc.

OpenEdge includes Sonic software, which includes software developed by IBM. Copyright © 1995-2003 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. Software distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

OpenEdge includes Sonic software, which includes the JMX Technology from Sun Microsystems, Inc. Use and Distribution is subject to the Sun Community Source License available at http://sun.com/software/communitysource.

OpenEdge includes Sonic software, which includes software developed by the ModelObjects Group (http://www.modelobjects.com). Copyright © 2000-2001 ModelObjects Group. All rights reserved. The name “ModelObjects” must not be used to endorse or promote products derived from this software without prior written permission. Products derived from this software may not be called “ModelObjects”, nor may “ModelObjects” appear in their name, without prior written permission. For written permission, please contact djacobs@modelobjects.com.

OpenEdge includes Sonic software, which includes code licensed from Mort Bay Consulting Pty. Ltd. The Jetty Package is Copyright © 1998 Mort Bay Consulting Pty. Ltd. (Australia) and others.
OpenEdge includes Sonic software, which includes files that are subject to the Netscape Public License Version 1.1 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/NPL/. Software distributed under the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License. The Original Code is Mozilla Communicator client code, released March 31, 1998. The Initial Developer of the Original Code is Netscape Communications Corporation. Portions created by Netscape are Copyright 1998-1999 Netscape Communications Corporation. All Rights Reserved.

OpenEdge includes Sonic software, which includes software developed by the University Corporation for Advanced Internet Development http://www.ucaid.edu Internet2 Project. Copyright © 2002 University Corporation for Advanced Internet Development, Inc. All rights reserved. Neither the name of OpenSAML nor the names of its contributors, nor Internet2, nor the University Corporation for Advanced Internet Development, Inc., nor UCAID may be used to endorse or promote products derived from this software and products derived from this software may not be called OpenSAML, Internet2, UCAID, or the University Corporation for Advanced Internet Development, nor may OpenSAML appear in their name without prior written permission of the University Corporation for Advanced Internet Development. For written permission, please contact opensaml@opensaml.org.

OpenEdge includes the UnixWare platform of Perl Runtime authored by Kiem-Phong Vo and David Korn. Copyright © 1991, 1996 by AT&T Labs. Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. THIS SOFTWARE IS BEING PROVIDED “AS IS”, WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHORS NOR AT&T LABS MAKE ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

OpenEdge includes Vermont Views Terminal Handling Package software developed by Vermont Creative Software. Copyright © 1988-1991 by Vermont Creative Software.

OpenEdge includes XML Tools, which includes versions 8.9 of the Saxon XSLT and XQuery Processor from Saxonica Limited (http://www.saxonica.com/) which are available from SourceForge (http://sourceforge.net/projects/saxon/). The Original Code of Saxon comprises all those components which are not explicitly attributed to other parties. The Initial Developer of the Original Code is Michael Kay. Until February 2001 Michael Kay was an employee of International Computers Limited (now part of Fujitsu Limited), and original code developed during that time was released under this license by permission from International Computers Limited. From February 2001 until February 2004 Michael Kay was an employee of Software AG, and code developed during that time was released under this license by permission from Software AG, acting as a "Contributor". Subsequent code has been developed by Saxonica Limited, of which Michael Kay is a Director, again acting as a "Contributor". A small number of modules, or enhancements to modules, have been developed by other individuals (either written especially for Saxon, or incorporated into Saxon having initially been released as part of another open source product). Such contributions are acknowledged individually in comments attached to the relevant code modules. All Rights Reserved. The contents of the Saxon files are subject to the Mozilla Public License Version 1.0 (the "License"); you may not use these files except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/ and a copy of the license can also be found in the
installation directory, in the c:/OpenEdge/licenses folder. Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

OpenEdge includes XML Tools, which includes Xs3P v1.1.3. The contents of this file are subject to the DSTC Public License (DPL) Version 1.1 (the "License"); you may not use this file except in compliance with the License. A copy of the license can be found in the installation directory, in the c:/OpenEdge/licenses folder. Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License. The Original Code is xs3p. The Initial Developer of the Original Code is DSTC. Portions created by DSTC are Copyright © 2001, 2002 DSTC Pty Ltd. All rights reserved.

OpenEdge includes YAJL software Copyright 2007, Lloyd Hilaiel. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of Lloyd Hilaiel nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Data Dictionary Basics

This chapter introduces the Data Dictionary and its menu options in the following sections:

• Starting the Data Dictionary
• Data Dictionary menu bar

Note: You cannot use the Data Dictionary to manipulate data from previous version databases.
Starting the Data Dictionary

Start the Data Dictionary in one of the following ways:

- At the command line, enter the following command:
  
  `pro -p dict.p`

- From any of the Application Development tools, choose **Tools**→**Data Dictionary**.

- In the Procedure Editor, type `dict`. Then choose **Compile**→**Run** or press **GO**.

The Procedure Editor opens the Data Dictionary, as shown in Figure 1–1.

![Data Dictionary window](image)

**Figure 1–1:** Data Dictionary window

Once open, the Data Dictionary displays the name of the working database in the lower-left corner of the window (if you have at least one connected database). You can have multiple databases connected at any time, but only one is the current, working database. To select an existing working database, choose **Database**→**Select Working Database** from the menu bar.
Figure 1–2 shows the Data Dictionary menu bar.

Table 1–1 describes the pull-down menus and options available from the menu bar in the Data Dictionary.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Manages database creation, connection, and disconnection, and generates reports on schema, auditing, alternate buffer pool, and transparent data encryption; also exits the Data Dictionary</td>
</tr>
<tr>
<td>Schema</td>
<td>Modifies database structure, including defining tables, fields, indexes, sequences, and triggers</td>
</tr>
<tr>
<td>Admin</td>
<td>Dumps and loads data and definitions; defines security options for database access, auditing, and encryption; imports and exports data; allows enabling of large key entries; and defines Alternate Buffer Pool assignments</td>
</tr>
<tr>
<td>DataServer</td>
<td>Accesses utilities to view and maintain the schema images for non-OpenEdge databases</td>
</tr>
<tr>
<td>Utilities</td>
<td>Creates parameter files, freezes/unfreezes tables, prepares data for dumping and loading, and generates include files</td>
</tr>
<tr>
<td>PRO/SQL</td>
<td>Performs SQL-related tasks</td>
</tr>
<tr>
<td>Tools</td>
<td>Accesses the Procedure Editor, OS Shell, and Application Compiler</td>
</tr>
</tbody>
</table>
The menu bar provides access to all the available options. However, if you do not have a database connected, you can only access the options listed in Table 1–2.

**Table 1–2: Options available without a connected database**

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Pull-down menu option</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
<td>Create...</td>
<td>Creates a new database</td>
</tr>
<tr>
<td></td>
<td>Connect...</td>
<td>Connects to an existing database</td>
</tr>
<tr>
<td></td>
<td>Exit</td>
<td>Exits the Data Dictionary</td>
</tr>
<tr>
<td><strong>Utilities</strong></td>
<td>Editor for Parameter Files...</td>
<td>Creates or edits a parameter file</td>
</tr>
<tr>
<td></td>
<td>Quoter Functions</td>
<td>Formats data files so they can read the data into an OpenEdge® database</td>
</tr>
<tr>
<td></td>
<td>Information...</td>
<td>Displays session information</td>
</tr>
<tr>
<td><strong>Tools</strong></td>
<td>All options</td>
<td>Accesses the Procedure Editor, OS Shell, and Application Compiler</td>
</tr>
</tbody>
</table>
This chapter describes the following database tasks can be performed using the Data Dictionary:

- Creating a database
- Connecting to a database
- Switching to another connected database
- Disconnecting a database
- Viewing session information
- Dumping data definitions
- Creating an incremental data definition file
- Creating a parameter file
- Editing a parameter file
- Loading data definitions
Creating a database

There are many ways to create an OpenEdge database. For a complete discussion about database creation, see *OpenEdge Data Management: Database Administration*.

To create a database using the Data Dictionary:

1. Choose **Database**→**Create**. The **Create Database** dialog box appears.

2. Enter the name of the new database.

3. Specify whether you want to create an empty database, a copy of the SPORTS database, a copy of the SPORTS2000 database, or a copy of an existing database.

4. Select **Replace If Exists** to specify whether you want to overwrite an existing database with the same name.

   **Note:** If you do not choose the **Replace If Exists** button, and a database already exists with the same name, the Data Dictionary returns an error.

5. Select **New Instance** if you are making a copy of a database and want the Data Dictionary to create a unique DB identifier for the copied database.

   If you do not select **New Instance**, the Data Dictionary assigns the source database’s DB Identifier to the copy.

   **Note:** If you want to create a copy of a database for backup purposes, do not select **New Instance**. If, however, the copy is of an audit-enabled database and will continue to store new audit data, select **New Instance**. All audit-enabled databases need unique DB Identifiers. For more information about audit-enabled databases, see *OpenEdge Getting Started: Core Business Services*.

If no database with that name exists, the Data Dictionary creates the database, or if one does and you chose the **Replace If Exists** button, the Data Dictionary overwrites the existing database and the **Database Connect** dialog box appears. Define the database attributes, as described in the “Connecting to a database” section on page 2–3, then choose **OK**.

If more than one database is connected, the **Select Working Database** dialog box appears. After you select a working database, the Data Dictionary main window reappears.
Connecting to a database

There are many ways to connect to an OpenEdge database. For detailed information about connecting to a database, see *OpenEdge Data Management: Database Administration*.

To connect to a database using the Data Dictionary:

1. Choose **Database**→**Connect**. The **Connect Database** dialog box appears:

   ![Connect Database Dialog Box](image)

   - **Physical Name**: ________________
   - **Logical Name**: ________________
   - **Database Type**: PROGRESS
   - **User ID**: ________________
   - **Password**: ________________
   - **Trigger Location**: ________________
   - **Parameter File**: ________________
   - **Other CONNECT Statement Parameters**: ________________

2. Specify the database name and any necessary database information, then choose **OK**. See the “Connect option” section on page 8–5 for a complete description of the database attributes you can specify in the **Connect Database** dialog box.

**Note**: You must select the **Multiple Users** toggle-box if you are running multi-user and you are trying to connect to a self-service client.
Switching to another connected database

The Data Dictionary allows you to switch to a new working database.

To switch to another connected database:

1. Choose **Database**→**Select Working Database**. The Data Dictionary displays a list of the connected databases.

2. Highlight the database you want for the working database, then press **RETURN**.
Disconnecting a database

There are many ways to disconnect a database. For detailed information of all disconnection methods, see OpenEdge Data Management: Database Administration.

To disconnect a database using the Data Dictionary:

1. Choose Database → Disconnect. If you have more than one database connected, the Data Dictionary displays the Select Database to Disconnect dialog box.

2. Select the database you want to disconnect from the database selection list, then press RETURN. The Data Dictionary displays an alert box prompting you to verify the disconnect.

3. Choose Yes to disconnect the selected database.

If you have more than one database connected, the Select Working Database dialog box appears and lets you select another database as the working database.

**Note:** If you disconnect a schema holder, the Data Dictionary also disconnects all of the schema holder’s corresponding non-OpenEdge databases.
Viewing session information

To display current database session information, choose Utilities → Information. The Session Information dialog box appears, which contains information such as how many databases are connected, the physical and logical names of the working database, the database type and version, and whether 64-bit sequences and large key entries are enabled for the currently selected database. (You might need to scroll down to see all the information.)
Dumping data definitions

When you dump your database’s data definitions, a data definition file (.df) is created. That file contains definitions for tables, fields, and indexes.

**Note:** The Data Dictionary allows you to dump data definition files (.df) greater than 2GB on platforms where ABL supports large files.

---

To dump data definitions:

1. Choose **Admin → Dump Data and Definitions → Data Definitions (.df file)**. The Data Dictionary alphabetically lists all the tables defined for your database.

2. You can select one or more tables as follows:
   - To select one table, highlight the table name and choose **OK**.
   - To select more than one table, choose **Select Some** and then choose **OK**. The Select Tables by Pattern Match dialog box appears. Type the table name, use the wildcard patterns, or type **ALL**. Then choose **OK**.

   If you type **ALL**, the .df file will also contain the sequence and auto-connect record definitions, but not the collation/conversion table. The Data Dictionary displays a default name for the output file into which you can dump your data definitions (hidden tables are not dumped). This default file is the name of the dump file you specified when you created the table or, for multiple tables, the database name with the .df extension.

   **Note:** When you dump only one table, the table’s dump name becomes the default for its corresponding contents dump file. For example, if you select the OrderLine table, the default filename for its corresponding contents dump file is `orderline.df`. If you choose to dump all the tables, the default name for the file contents is `db-name.df`. See the “Dumping table contents” section on page 3–11 for information about how to dump table contents.

3. Specify the filename or use the default value.

4. Specify the code page or accept the default, which is the value specified by the SESSION:STREAM or SESSION:CHARSET attribute, then choose **OK**. For more information on the SESSION:STREAM or SESSION:CHARSET attributes, see OpenEdge Deployment: Managing ABL Applications.

5. Select the **Include POSITION for .r / Binary Load Compatibility** option if you want to maintain the sequential numbering of the fields in the file after you dump the table. Activating this option ensures that you are not required to recompile .r code against any new schema that is created by loading a new *.df file, and that data dumped with PROUTIL (binary dump) can be loaded into the database.

6. Choose **OK**. The Data Dictionary displays each object name as it writes its definition to the .df file. The Data Dictionary displays a status message and prompts you to continue.
Creating an incremental data definition file

An incremental data definition file contains the differences in schema when you contrast OpenEdge databases.

Note that the Create Incremental .df file utility detects and allows fields of INTEGER data type to be changed to INT64 without requiring a dump and load of the field’s data. All other changes in data types will cause a generation of the delta.df file that will delete the existing field and recreate it with the new data type, requiring you to perform a dump and load. For more information about INT64 and other data types, see Table 9–15.

Note: This utility can only be used with OpenEdge databases.

To create an incremental data definition file:

1. Choose Database→Select Working Database, then select the new database as the working database.

2. Choose Admin→Dump Data and Definitions→Create Incremental .df File. The Data Dictionary lists all your connected databases.

3. Select the database you want to compare the working database against (that is, the older version of the database). The Data Dictionary then prompts you for the filename to which you want to write the differences.

4. Enter the filename to which you want to write the differences. The default filename is delta.df.

5. Specify a code page.

6. Choose OK. The Data Dictionary displays the table, field, index, and sequence names as it compares the databases and prompts you to specify what you want to do with any differences it finds. The Data Dictionary displays a status message when it has completed dumping the table definitions.

7. Choose OK to return to the menu bar.
Creating a parameter file

Parameter files are useful if you regularly use the same parameters when starting the OpenEdge RDBMS, or if you want to specify more parameters than can fit on the command line.

To create a parameter file:

1. Choose Utilities→Editor for Parameter Files. The Data Dictionary prompts you for the file you want to create.

2. Specify the filename. The Data Dictionary alphabetically lists all available startup parameters.

3. Specify which of the parameters you want in the file by entering yes, no, or the appropriate value, then press GO. The Data Dictionary prompts you for comments.

4. Enter any comments you want to include in the parameter file, and choose OK. The Data Dictionary window reappears.
Editing a parameter file

The Data Dictionary allows you to edit parameter files.

To edit a parameter file:

1. Choose Utilities → Editor for Parameter Files. The Data Dictionary prompts you for the file you want to edit.

2. Specify the name of an existing parameter file. The Data Dictionary alphabetically lists all available startup parameters with the values for the specified file.

3. Modify the file by entering yes, no, or the appropriate value, then press GO. The Data Dictionary prompts you for comments.

4. Enter any comments you want to include in the parameter file, and choose OK. The Data Dictionary window reappears.
Loading data definitions

To load definitions into a database, you use a data definition (.df) file. The .df file contains definitions for tables, fields, and indexes. Based on the contents of the .df file, you can load either all the data definitions for a table or only those that have changed.

**Note:** The Data Dictionary allows you to load data definition files (.df) greater than 2GB on platforms where ABL supports large files.

**To load data definitions into a table:**

1. Choose **Admin → Load Data and Definitions → Data Definitions (.df file)**. The Data Dictionary prompts you for the name of the file that contains the data definitions you want to load into your current working database.

2. Specify a filename or accept the default. The default filename is the working physical database name with a .df extension.

3. Specify whether you want to stop the load in the event a bad definition statement is encountered.

4. Specify if you prefer error message to be output as files.

5. Specify if you want to add database objects while online.

6. Specify whether to commit data definition changes, even if doing so will create an error message.

7. Specify if you prefer error messages output to your screen instead of a file.

8. Choose **OK**.

If the data definitions file does not specify a code page, the Data Dictionary prompts you for one.

The Data Dictionary displays each item as it loads the definitions for that object. It then displays a status message and prompts you to continue.

If the definition file contains encryption policies, you must be authorized to load the file.
This chapter describes procedures for using the Data Dictionary to perform the following administrative tasks for tables and files:

- Creating a table
- Modifying a table
- Deleting a table
- Freezing or unfreezing a table
- Generating a table report
- Dumping table contents
- Loading table contents
- Binary dump and load
- Creating a bulk loader description file
- Generating an include file
Creating a table

The Data Dictionary allows you to add a new table to existing storage areas.

To create a table:

1. Choose Schema→ Add New Table. The Add New Table dialog box appears.

2. Define the table attributes. For more information on the Add New Table dialog box, see the “Add New Table option” section on page 9–9.

Specifying validation criteria

You use validation criteria to control which records can be deleted from the table.

To specify table validation criteria:

1. Choose the Validation button. The Data Dictionary displays the Table Validation dialog box:

   ![Table Validation dialog box]

   1. Specify the message you want the Data Dictionary to display when a user tries an invalid deletion. There is no limit to the number of characters you can enter. A message up to 63 characters long can fit onto the two message lines without scrolling.

   2. Specify the criteria for valid deletions. For example, in the customer table, to restrict deleting customers to those customers who have no outstanding orders, enter the following criteria:

   ```
   NOT (CAN-FIND(FIRST order OF customer))
   ```

   4. Choose OK to return to the Add New Table window.

Specifying triggers

See the “Creating a trigger” section on page 7–2 for more information about creating triggers.
Specifying string attributes

By specifying string attributes, you control how much space is allocated within an r-code text segment for a string.

To specify table string attributes:

1. Choose the **String Attributes** button. The Data Dictionary displays the **Table String Attributes** dialog box:

   ![Table String Attributes dialog box]

   **Table String Attributes**
   
   String attribute options are: T, R, L, C, U and # of characters.
   
   ![Label: ]
   ![Validation Message: ]
   
   ![OK] ![Cancel]

2. Specify the attribute characters for the table label and the table validation message, as follows:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>Specifies to remove trailing spaces during string display. The Data Dictionary does not physically remove the spaces from the string, but replaces them with null values. The Dictionary does not display null values.</td>
</tr>
<tr>
<td>R</td>
<td>Specifies to right-justify the string.</td>
</tr>
<tr>
<td>L</td>
<td>Specifies to left-justify the string.</td>
</tr>
<tr>
<td>C</td>
<td>Specifies to center the string.</td>
</tr>
<tr>
<td>U</td>
<td>Indicates that the Translation Manager and the compiler should not translate the string.</td>
</tr>
<tr>
<td>max-length</td>
<td>Specifies the amount of space allocated for the text segment. You can specify a number from 1 to 999.</td>
</tr>
</tbody>
</table>

You can specify the characters in any order but can only use each character once. Also, you cannot use any of the letters in conjunction with each other, except for **U**. If you do not specify the maximum length, the Dictionary uses the length of the original string.

If you do not specify justification, the Compiler left-justifies the string by default, unless the string is a column label. If the string is a column label for a decimal or integer column or a side label, it is right-justified.

3. Choose **OK** to return to the **Add New Table** window.
Modifying a table

When you modify a field, trigger, or index structure, you modify the table.

To modify a table definition:

1. Choose Schema → Modify Table. The Data Dictionary alphabetically lists all the tables defined for your database.
2. Select the table you want to modify. The Data Dictionary displays the Modify Table dialog box.
3. Modify the table definition.

**Note:** See the “Creating a table” section on page 3–2 for information on specifying table validation and string attributes. See the “Modifying a trigger” section on page 7–4 for information about modifying triggers.

4. Choose OK. A list of options appears.
5. Choose A to apply the changes to your database.
Deleting a table

Before deleting a table, make sure that it is no longer needed.

To delete a table:

1. Choose Schema→Delete Table(s) from the menu bar. The Data Dictionary alphabetically lists all the tables defined for your database.

2. Select the table you want to delete or type ALL to delete all tables, then press GO. The Data Dictionary prompts you to verify that you want to delete the selected table.

3. Choose Yes. The Data Dictionary deletes the table from the database.
Freezing or unfreezing a table

The Data Dictionary’s Freeze/Unfreeze utility helps you administer your database.

To freeze or unfreeze a table:

1. Choose Utilities → Freeze/Unfreeze. The Data Dictionary alphabetically lists all the tables defined for your database.

2. Select the table you want to freeze or unfreeze. The Data Dictionary displays the Freeze/Unfreeze Table dialog box, as shown:

   ![Freeze/Unfreeze Table dialog box](image)

3. Specify yes or no in the Frozen field.

4. Choose OK to return to the Data Dictionary window.
Generating a table report

This section describes how to generate the following table reports:

- Detailed
- Quick
- Relations

Generating a Detailed Table Report

A Detailed Table Report gives the following information for each table in the database:

- Name
- Description
- Dump name
- Flags
- Field count
- Index count
- Labels
- Storage area name
- Trigger events
- Trigger procedure filenames

- A summary of each field, including:
  - Name
  - Order
  - Data type
  - Flags
  - Format
  - Initial value
  - Field label
  - Column label
A summary of each index, including:
- Name
- Flags
- Count
- Field name
- Storage area

Details about each field, including:
- Validation message
- Validation expression
- Description
- Help text
- Triggers
- Trigger procedure filenames
- Whether the field is overridable
- Whether to check the CRC

To generate a Detailed Table Report:

1. Choose Database → Reports → Detailed Table. The Data Dictionary alphabetically lists all the tables defined for your database.

2. Select a table or type ALL. The Report Options dialog box appears:

   ![Report Options dialog box]

3. Specify the report destination. You can display the report on your terminal, print it, or save it to a file. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.
4. Specify the page length.

5. Specify whether to order the fields by order number or alphabetically, then choose OK.

If you specified a file or printer, the Data Dictionary generates the report to the specified output destination. The Data Dictionary main window then reappears.

If you sent the report to the terminal, the Detailed Table Report dialog box appears. You can choose Change Field Order to reorder the fields, choose Print to print the report or write it to a file, or choose OK to close the Report Options dialog box.

Generating a Quick Table Report

A Quick Table Report lists the name of each table in the database. The Quick Table Report also lists:

- Each table’s area number
- The name of each table’s dump file
- Any flags (such as f if the table is frozen or s for an SQL table)
- The number of fields
- The number of indexes
- The table’s label, if applicable

To generate a Quick Table Report:

1. Choose Database → Reports → Quick Table. The Quick Table Report dialog box appears and displays information about all the tables in the working database.

2. Choose Print if you want to print the report or to write it to a file. The Print Options dialog box appears.

3. Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.

4. Specify the page length and choose OK. The Data Dictionary sends the report to the specified output destination and returns you to the Quick Table Report dialog box.

5. Choose OK to return to the main Data Dictionary window.
Generating a Table Relations Report

The Table Relations Report details the relations between tables.

To generate a Table Relations Report:

1. Choose **Database**→**Reports**→**Table Relations**. The Data Dictionary alphabetically lists all the tables defined for your database.

2. Select a table or type **ALL**. The Data Dictionary generates the report.

3. Choose **Print** if you want to print the report or write it to a file. The **Print Options** dialog box appears.

4. Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.

5. Specify the page length and choose **OK**. The Data Dictionary sends the report to the specified output destination and returns you to the **Table Relations Report** dialog box.

6. Choose **Switch Tables** to generate a table relations report for another table, or choose **OK** to return to the Data Dictionary window.
Dumping table contents

When you dump table data, the Data Dictionary creates a contents file that contains data for each table you specify. Each file has a .d extension. The Data Dictionary uses contents files to load data into your database. The Data Dictionary creates a separate contents file for every table you dump.

**Note:** OpenEdge allows you to dump .d files exceeding 2GB when your operating system supports large files.

Binary dumps can be a more efficient way of dumping table contents. See the “Binary dump and load” section on page 3–15 for more information about performing a binary dump and load.

To dump table contents:

1. Choose Admin → Dump Data and Definitions → Table Contents (.d files) from the menu bar. The Data Dictionary alphabetically lists all the tables defined for your database (hidden tables do not display).

2. Select the table names whose contents you want to dump. Use the Select Some and Deselect Some buttons to select or deselect groups of tables.

3. Press GO.

If you selected a single table, the Data Dictionary displays a default name for the file that you can dump your table contents into. This default file name is always the dump name of the table definition file, with a .d extension. If you want to specify a file name other than the default file, type the name in the Output File field.

If you selected more than one table, the Data Dictionary prompts you for a directory name to write the contents files. If you do not specify a directory, the Data Dictionary creates the files in the current directory. The Data Dictionary names each contents file with the table’s corresponding dump name.

4. Accept the default or enter a different name.

5. If your table contains large objects, accept the Include LOB default of yes; otherwise, choose no.

6. If you chose yes in Step 5, type in the LOB directory field the name of the directory that contains the LOB.

7. If you want to use character mapping, enter the character mapping, then choose OK. See OpenEdge Development: Internationalizing Applications for information about character mapping, PROTERMCAP, and national language support.
8. Specify the code page or accept the default.

9. Choose **OK** to dump the data contents file(s). The Data Dictionary displays each table name as it writes the table contents to the table contents file. After dumping the table contents, the Data Dictionary displays a status message. Choose **OK** to return to the Data Dictionary window.

---

**Note:** You must dump your sequence values separately.

---

**To dump user table contents:**

1. Choose **Admin → Dump Data and Definitions → User Table Contents.** The Data Dictionary prompts you for the file to which to write the user file contents. The default filename is _user.d_.

2. Specify the filename or accept the default. After dumping the user file contents to the specified file, the Data Dictionary displays a status message and prompts you to continue.

3. Choose **OK** to return to the Data Dictionary main window.
Loading table contents

The Data Dictionary uses a contents (.d) file that you specify to load table data into your database. A contents file contains data for a table.

**Note:** OpenEdge allows you to load table contents files exceeding 2GB.

Remember the following when loading table contents:

- Table definitions must be in the database before you can load table contents.
- You must load your sequence values separately.

There are two ways to load the table contents (.d) files:

- Use the Data Dictionary.
- Use the Bulk Loader (PROUTIL BULKLOAD) utility. For information about using the Bulk Loader, see the chapter on administration utilities in *OpenEdge Data Management: Database Administration*.

For information about performing a binary dump and load, see the “Binary dump and load” section on page 3–15.

---

**To load table contents into a database:**

1. Make sure that the working database is the target database where you want to load the table contents.

2. Choose Admin → Load Data and Definitions → Table Contents (.d files) from the menu bar. The Data Dictionary alphabetically lists all the tables defined for your database.

3. Select the table names whose contents you want to load. Use the Select Some and Deselect Some buttons to select or deselect groups of tables.

   The Data Dictionary prompts you for the name of the contents file or (if you choose multiple files) the directory that contains the contents files you want to load into your current database.

4. Specify the directory name, filename, or use the default value.

   When you load the contents of more than one table, enter a directory from which you want to load the contents files. If you do not specify a directory, the Data Dictionary loads the files from the current directory. The Data Dictionary loads each table from the corresponding table-dumpname.d filename.

5. If your load contains large objects, accept the Include LOB default of yes; otherwise, choose no.

6. If you chose yes in Step 5, type in the LOB directory field the name of the directory that contains the LOB.
7. Specify the error rate.

As the Data Dictionary loads records from any files you designate, it might encounter data that cannot be loaded. The Data Dictionary does not load the record. It places all erroneous records in a `table.e` file (the name of the table with a `.e` extension). If you specify an error rate of 10 percent, the Data Dictionary must successfully load 90 records from every set of 100 records loaded.

8. Select **Output Errors to Screen** to view error messages generated while loading table data. The default output for errors is to a file.

9. Choose **OK**. After loading the table contents, the Data Dictionary displays a status message and prompts you to continue.

10. Choose **OK** to return to the Data Dictionary main window.

### To load user table contents:

1. Choose **Admin → Load Data and Definitions → User Table Contents**. The Data Dictionary prompts you for the file from which you want to read the user file contents. The default filename is `_user.d`.

2. Specify the filename or accept the default. After loading the user file contents to the specified file, the Data Dictionary displays a status message and prompts you to continue.

3. Choose **OK** to return to the Data Dictionary main window.

### Reconstructing bad load records

If the Data Dictionary encounters errors while loading a data file, it creates an error file. You can use this error file and the original data file to build a new data file for bad records.

### To reconstruct bad load records:

1. Choose **Admin → Load Data and Definitions → Reconstruct Bad Load Records**. The **Reconstruct Bad Load Records** dialog box appears:

   ![Reconstruct Bad Load Records dialog box](image)

   When a data load encounters errors, an error file is produced. This utility takes the error file and the original data file and produces a new data file with only the bad records. This new file can then be edited and reloaded.

   - **Original Data File:**  
   - **Error File:**  
   - **Output Data File:** `error.d`  

2. Specify the filename of the original data file, the error file, and the new data file. The default filename for the new data file is `error.d`.

3. Choose **OK**. The Data Dictionary displays a status message and prompts you to continue.
Binary dump and load

The OpenEdge database utility PROUTIL allows you to move and reorganize data by performing a binary dump and load instead of the Data Dictionary ASCII dump and load. The binary method improves the dump and load performance significantly because the data remains in binary format during the process. Also, if security is a concern, for an encryption-enabled database you can encrypt the output of your binary dump.

Performing the binary dump

Use the following PROUTIL command to perform a binary dump:

```
proutil dbname -C dump tablename directory [ -index num ]
```

**dbname**

Specifies the database from which you want to dump. You must completely define the path.

**tablename**

Specifies the name of the table containing the data you want to dump.

**directory**

Specifies the name of the target directory into which the data will be dumped.

**-index num**

Specifies an index with which to dump the table’s contents. If you choose not to use this option, the command uses the primary index to dump the table.

**Note:** Additional parameters to PROUTIL DUMP are available for security (encryption) and performance. For a complete description and syntax, see *OpenEdge Data Management: Database Administration*.

The dump command writes the data to a dump file named `table.bd`, where `table` is the name of the table that was dumped. For example, if you dump data from the table named “customer,” the procedure creates a dump file named `customer.bd` in the directory you specify in the command.
Each binary dump file contains a header and a description for each record in the table. The dump file appears in the following format:

<table>
<thead>
<tr>
<th>Header</th>
<th>Record length</th>
<th>Table number</th>
<th>Binary record</th>
<th>Record CRC</th>
</tr>
</thead>
</table>

The file header contains information that appears in the following order:

- Version number
- Date and time the file was created
- Name of the table being dumped
- Number of the table being dumped
- CRC of the table being dumped
- Number of fields in the table
- Name of the database in which the table resides
- Section number
- Number of the first record

When the dump procedure finishes, a screen message reports the number of records and the number of kilobytes of data that were dumped.

**Performing the binary load**

Use the following PROUTIL command to perform a binary load:

```
proutil dbname -C load filename
```

**dbname**

Specifies the database into which you want to load the data. You must completely define the path.

**filename**

Specifies the binary dump file that you want to load. You must completely define the path.

**Note:** Additional parameters to PROUTIL LOAD are available for security (encryption) and performance. For a complete description and syntax, see *OpenEdge Data Management: Database Administration*.

When the load procedure finishes, a screen message reports the number of records that were loaded.
Creating a bulk loader description file

You use the bulk loader description file with the Bulk Loader utility to load data (.d) files into an OpenEdge database quicker than you can using the Data Dictionary Load utility.

You must create a bulk loader description file before you can run the Bulk Loader utility. For more information about bulk loader description files, see the chapter on dumping and loading in *OpenEdge Data Management: Database Administration*.

To create a bulk loader description file:

1. Choose Admin→Create Bulk Loader Description File. The Data Dictionary alphabetically lists all the tables defined for your database.

2. Select the tables for which you want to create bulk loader description files and press GO. The Data Dictionary prompts you for the bulk load filename.

3. Specify a bulk load filename. The default filename is `table-name.fd`.

4. If your load contains large objects, accept the Include LOB default of yes; otherwise, choose no.

5. If you choose yes in Step 4, type in the LOB directory field the name of the directory that contains the LOB.

6. Specify the code page.

7. Choose OK. After dumping the table contents to the specified file, the Data Dictionary displays a status message and prompts you to continue.
Generating an include file

You can generate include files for the following statements:

- ASSIGN
- FORM
- DEFINE WORK-TABLE

To create an ASSIGN statement include file:

1. Choose Utilities → Generate Include Files → ASSIGN Statement. The Data Dictionary alphabetically lists all the tables defined for your database.
2. Select the table for which you want to create the ASSIGN statement. The Data Dictionary prompts you for the filename of the output file.
3. Specify the filename. The default filename is table-name.i.
4. Choose OK. The Data Dictionary displays a status message and prompts you to continue.

To create a FORM statement include file:

1. Choose Utilities → Generate Include Files → FORM Statement. The Data Dictionary alphabetically lists all the tables defined for your database.
2. Select the table for which you want to create the FORM statement. The Generate FORM for "selected-table-name" dialog box appears:

   Output File: customer.f ............................................ (Files...)
   Fully Qualify Names?: No
   Fully Expand Arrays?: Yes
   Frame Type: Normal/Diag/Overlay/Op-Only
   Formatting: Prop Dictionary/Explicit
   Validation: None
   Labeling: Prop Dictionary/Explicit/None
   Labels: Side Side/Top/None

   NOTE: Only Non-LOB fields will be included in the output file

3. Enter the appropriate information in the fill-in fields, then choose OK. The Data Dictionary displays a status message and prompts you to continue.
To create a DEFINE WORK-TABLE statement include file:

1. Choose **Utilities**→ **Generate Include Files**→ **DEFINE WORK-TABLE statement**. The Data Dictionary alphabetically lists all the tables defined for your database.

2. Select the table for which you want to create the DEFINE WORK-TABLE statement. The Data Dictionary prompts you for the filename of the output file.

3. Specify the filename. The default filename is `table-name.i`.

4. Choose **OK**. The Data Dictionary displays a status message and prompts you to continue.
<table>
<thead>
<tr>
<th>Table Tasks</th>
</tr>
</thead>
</table>
This chapter describes the following tasks involved in creating and maintaining fields:

- Adding a field
- Modifying a field
- Changing a field name globally
- Copying a field
- Deleting a field
- Changing field order
- Generating a Quick Field Report
- Adjusting the Field Width
Adding a field

When you add a field, you modify the database’s schema.

To add a field to a table:

1. Choose Schema → Field Editor. The Data Dictionary’s Field Editor alphabetically lists all the tables defined for your database.

2. Select the table to which you want to add a new field. The Field Editor lists all the fields defined for the selected table.

3. Choose the Add option from the horizontal menu at the bottom of the window. A window similar to the following appears:

4. Define the field attributes. See the “Field Editor option” section on page 9–11 for more information about the fields in this window.

5. Press RETURN or GO.

6. Choose Exit to apply the change. The Field Editor displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

Modifying a field

When you edit a field definition, you modify the database’s schema. Changing a field’s data type from INTEGER to INT64 is a special case (for information, see the “Changing a field data type from integer to INT64” section on page 4–3).

To modify a field definition:

1. Choose Schema → Field Editor. The Data Dictionary’s Field Editor alphabetically lists all the tables defined for your database.
2. Select the table that contains the field you want to modify. The Field Editor lists all the fields defined for the selected table.
3. Choose Modify.
4. Choose the field you want to modify. The Field Editor displays the field definition in the window. See the “Field Editor option” section on page 9–11 for information about the fields in this window.
5. Edit the field definition.
6. Press RETURN or GO.
7. Choose Exit to apply the change. An alert box appears, asking if you want to apply or undo the changes to the database, or if you want to make other changes.

Changing a field data type from integer to INT64

You can change the data type of a field from integer to INT64. No dump and load of the field’s data is required.

Note: To change a field’s data type from INT64 back to integer, you must perform a dump and load of the field’s data.

To change a field’s data type from integer to INT64:

1. Choose Schema → Field Editor. The Data Dictionary’s Field Editor alphabetically lists all the tables defined for your database.
2. Select the table that contains the field you want to modify. The Field Editor lists all the fields defined for the selected table.
3. Choose Modify.
4. Choose the field you want to modify. The Field Editor displays the field definition in the window. See the “Field Editor option” section on page 9–11 for information about the fields on this window.
5. Change the data type from integer to int64, and press **GO**. The **Confirmation of data type change** window appears:

![Confirmation of data type change](image)

If you change this field’s data type to int64, you will not be able to change the data type back to integer after you commit the changes. To revert this change back, you will need to dump/load this table. Do you really want to change this field’s data type?

[ ] Don’t show me this again (for this session)

Yes  No

6. Select the **Don't show me this again (for this session)** option to stop confirmation messages regarding changing the data type from appearing for the rest of the session.

7. Choose **Yes** to make the change, and then choose **A** to apply the changes.
Changing a field name globally

The Data Dictionary allows you to rename a field throughout all tables.

To change a field name globally:

1. Choose **Schema → Global Field Name Change**. The **Global Field Name Change** dialog box appears.
2. Specify the current field name. Press **F5** to display a list of all field names.
3. Specify the new name.
4. Choose **OK**. The Data Dictionary displays an alert box prompting for verification of the global field name change.
5. Choose **Yes** to apply the change. The main menu of the Data Dictionary appears.
Copy a field

You can copy a field from one table to another table in the working database.

To copy a field from one table to another:

1. Choose Schema → Field Editor. The Field Editor alphabetically lists all the tables defined for your database.
2. Select the table to which you want to copy the field. The Field Editor lists all the fields defined for the selected table.
3. Choose the Copy option from the menu at the bottom of the window.
4. Select the table from which you want to copy the field. The Field Editor alphabetically lists all the fields defined for that table.
5. Select the fields you want to copy, then press GO. The Field Editor copies the specified fields into the table you selected. The main menu of the Field Editor reappears.
6. Choose Exit to apply the change. The Data Dictionary displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.
Deleting a field

When you delete a field, you modify the database’s schema.

**To delete a field:**

1. Choose **Schema → Field Editor**. The **Field Editor** alphabetically lists all the tables defined for your database.

2. Select the table that contains the field you want to delete. The **Field Editor** lists all the fields defined for the selected table.

3. Choose **Delete**.

4. Choose the field you want to delete, then press **RETURN**. The **Field Editor** prompts you to verify that you want to delete the field.

5. Choose **Yes**. The **Field Editor** main window reappears.

6. Choose **Exit** to apply the change. An alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes, appears.

7. Choose **Apply Changes**. The main menu of the Data Dictionary reappears.
Changing field order

To view the current order of a database’s fields, run either a Detailed Table Report or a Quick Field Report.

To change field order:


2. Select the table for which you want to reorder the fields. A dialog box similar to the following appears:

```
Resequence Order of Fields in 'Customer'
Sort by Field Name or Order: Order (F/D)
Start Numbering From: 10
Increment Each By: 10
Table contains 18 fields.
<OK> <Cancel>
```

3. Specify whether to order fields either by name or by order number.

4. Specify the starting number and the number by which to increment.

5. Choose OK. The Data Dictionary arranges the fields for the selected table in the specified order.

6. Choose OK to return to the Data Dictionary main window.
Generating a Quick Field Report

A Quick Field Report lists the following:

- Field name
- Order
- Data type
- Flags
- Format

To generate a Quick Field Report:

1. Choose Database → Reports → Quick Field. The Data Dictionary alphabetically lists all the tables defined for your database.

2. Select a table or type ALL. The Data Dictionary generates the report. The Quick Field Report dialog box appears, listing the fields in the table by order number.

3. Choose Switch Tables to generate a quick field report for another table. The Data Dictionary alphabetically lists all the tables for you to choose another table.

4. Choose Change Field Order to reorder the fields alphabetically.

5. Choose Print to print the report or write it to a file. The Print Options dialog box appears.

6. Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.

7. Specify the page length, then choose OK. The Data Dictionary sends the report to the specified output destination and returns you to the Quick Field Report dialog box.

8. Choose OK to return to the Data Dictionary window.
Adjusting the Field Width

The Field Width property allows the ABL client to modify the field width of an SQL client. If you do not adjust the setting, the SQL client uses the default width property.

To adjust the Field Width:

1. Choose Schema → Adjust Field Width. A table name appears in the Table Name field, which is followed by a list of additional table names.

2. Choose the table you want; the name appears in the Table Name field.

3. Press ENTER to open the table. The width of the first field in the field name listing is highlighted.

4. Select the name of the field whose width you want to adjust, select its width setting, and make the adjustment.

5. Choose OK, then press ENTER.

6. Choose Apply Changes.
This chapter describes the following tasks involved in creating and maintaining indexes:

- Creating an index
- Modifying an index
- Deleting an index
- Generating a Quick Index Report
Creating an index

Adding an index modifies the database’s schema.

To create an index:

1. Choose **Schema** → **Index Editor**. The **Index Editor** alphabetically lists all the tables defined for your database, as shown:

```
Database Schema Admin Baser Server Utilities PRO/SQL Tools

Table Name:
Customer

Customer
Invoice
Item
Local-Default
Order
Order-Line
Ref-Call
Salesrep
State

Index Editor | Data Dictionary
Database: Products (PROGRESS) | Table: Customer

Press the [F4] key to end.
```

2. Select a table.

3. Choose **Add**.

4. Enter a name for the index. Specify whether the index is unique or non-unique, active or inactive, and a word index or a standard index.

5. Press **RETURN**.

6. Select an area in which to store the new index. (Scroll down to see available options.)

7. Press **GO**. The **Adding Index** window appears, listing the field names you can add to the new index:

```
Database Schema Admin Baser Server Utilities PRO/SQL Tools

Name: Index  |  Adding index 'Index'  |  UNIQUE  |  ACTIVE
Num Field Name | Type Asc?
-------------------------------
Address2 | character
Balance | decimal
City | character
Comments | character
Contact | character
Country | character
Credit-Limit | decimal
Cust-Hun | integer
Discount | integer
Name | character
Phone | character

Database: Products (PROGRESS) | Table: Customer

Use the cursor-motion keys to navigate, or type field name and highlight will move. As you select fields to be key components, they are moved above the dividing line. Use [RETURN] to select fields. To set asc/desc, use +/- keys.

To unselect fields, move highlight above dividing line and [RETURN]. When done, press [F4].

```
8. Highlight the desired field and press RETURN. The Index Editor prompts you to specify whether you want this index component to be ascending or descending.

9. Specify Yes for ascending or No for descending. The Index Editor moves the selected field above the dividing line of the window. To change the sort order of a field above the dividing bar, highlight the field. Then press the plus sign (+) for ascending or the minus sign (−) for descending.

10. Select the next field. You can specify up to 16 fields to define an index.

11. When you are done choosing fields, press GO. If the index has a character field as the last index component, the Index Editor prompts you to specify whether you want to use the Abbreviate option.

   If you choose to use the Abbreviate option, the Index Editor displays the following Question:

   **Question**
   
   IF OpenEdge finds duplicate values while creating this new unique index, it will UNDO the entire transaction, causing you to lose any schema changes just made. Creating an inactive index and then building it with "proutil -C id=build" will eliminate this problem. Do you want to de-activate this index?

   <Yes> [Y] [N]

12. Specify whether to deactivate the index. The Index Editor main window reappears and adds the new index to the list.

13. Choose Exit. The Data Dictionary displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

Modifying an index

You can use the Data Dictionary to make an index the primary index for the table, rename an index, or deactivate an index.

**Note:** To change the component fields of an index, such as ascending or descending order or uniqueness, you must delete and then recreate the index.

Changing the primary index for a table

To view which index is currently the primary index for a table, run a Detailed Table Report or a Quick Index Report. The primary index is flagged with a "p."

**Note:** A word index cannot be the primary index.

To change the primary index for a table:

1. Choose **Schema**→ **Index Editor**. The **Index Editor** alphabetically lists all the tables defined for your database.
2. Select a table.
3. Select the index you want to make primary.
4. Choose **Change Primary**. The **Index Editor** prompts you to verify that you want to make the selected index the primary index for the table.
5. Choose **Yes** to make the selected index the primary index for the table. The main window of the **Index Editor** reappears.
6. Choose **Exit**. The Data Dictionary displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.
7. Choose **Apply Changes**. The Data Dictionary main window appears.

Renaming an index

Renaming an index modifies the database’s schema.

To rename an index:

1. Choose **Schema**→ **Index Editor**. The **Index Editor** alphabetically lists all the tables defined for your database.
2. Select a table. The **Index Editor** lists the indexes defined for that table.
3. Select the index you want to rename.
4. Choose **Rename**. The **Index Editor** prompts you for the new name.
5. Enter the new index name, then press RETURN. You cannot use the name of an existing index.

6. Choose Exit. The Data Dictionary displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.


Deactivating an index

You can deactivate an active table index.

**Note:** To see which indexes are currently active, run a Detailed Table Report or a Quick Index Report. Inactive indexes are flagged with an “i.”

**To deactivate an index:**

1. Choose Schema → Index Editor. The Index Editor alphabetically lists all the tables defined for your database.

2. Select a table. The Index Editor lists the indexes defined for that table.

3. Select the index you want to deactivate.

4. Choose MakeInactive. The Index Editor prompts you to verify that you want to make the selected index inactive.

5. Choose Yes to deactivate the index.

6. Choose Exit. The Data Dictionary displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

7. Choose Apply Changes. The Data Dictionary main window appears.

**Note:** For more information about indexes, see *OpenEdge Data Management: Database Administration.*
Deleting an index

Deleting an index modifies a database’s schema.

**Note:** You cannot delete the primary index. To delete the index that is currently the primary index, you must first make another index the primary index.

To delete an index:

1. Choose **Schema → Index Editor**. The **Index Editor** alphabetically lists all the tables defined for your database.
2. Select a table. The **Index Editor** lists the indexes defined for that table.
3. Select the index you want to delete. You cannot delete a primary index.
4. Choose **Delete**. The Data Dictionary prompts you to verify that you want to remove this index.
5. Choose **Yes** to delete the index. The **Index Editor** main window reappears.
6. Choose **Exit**. The Data Dictionary displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.
7. Choose **Apply Changes**. The Data Dictionary main window appears.
Generating a Quick Index Report

A Quick Index Report lists the following:

- Flags that indicate if the index is:
  - Primary
  - Unique
  - Word
  - Abbreviated
  - Inactive
  - Ascending
  - Descending
- Index name
- Area number
- Field names

To generate a Quick Index Report:

1. Choose Database→Reports→Quick Index. The Data Dictionary alphabetically lists all the tables defined for your database.
2. Select a table or type ALL, then press RETURN. The Data Dictionary generates the report and displays the Quick Index Report dialog box.
3. Choose Switch Tables to generate a quick index report for another table. The Data Dictionary alphabetically lists all the tables.
4. Choose Print to print the report or write it to a file. The Print Options dialog box appears.
5. Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.
6. Specify the page length, then choose OK. The Data Dictionary sends the report to the specified output destination and returns you to the Quick Index Report dialog box.
7. Choose OK to return to the Data Dictionary main window.
Sequence Tasks

Information related to creating and maintaining sequences is described in the following sections:

- Support for 64-bit sequences
- Creating a sequence
- Modifying a sequence
- Deleting a sequence
- Dumping sequence definitions
- Dumping sequence current values
- Loading sequence current values
- Generating a sequence report

See *OpenEdge Data Management: Database Administration* for more information about dumping and loading sequences.
Support for 64-bit sequences

Large sequences (those with 64-bit values) are enabled by default in newly created databases with OpenEdge Release 10.1C and higher. Once a database is enabled for 64-bit sequence support, you cannot revert that database to 32-bit sequences. If you try to load a sequence value that is too large for an integer value, you are notified of the overflow error and the load process aborts.

When you look at a sequence’s properties, you can now see the current value of a sequence if it has been committed.

If large sequences are not enabled, a Large sequence support not enabled message appears when you look at the Database or Sequence properties in the Data Dictionary.

To enable large sequences on a migrated database, use PROUTIL ENABLESEQ64 (see OpenEdge Data Management: Database Administration for more information).
Creating a sequence modifies a database’s schema.

To create a sequence:

1. Choose Schema → Sequence Editor. The Sequence Editor displays a list of the sequences defined for the working database and the attributes for the sequence that is selected.

2. Choose Add. The following window appears:

3. Define the sequence attributes, then press GO. The Sequence Editor main window appears.

4. Choose Exit. The Data Dictionary displays an alert box asking if you want to apply or undo the changes to the database, or if you want to make other changes.

5. Choose Apply Changes. The Data Dictionary main window appears.
Modifying a sequence

Editing a sequence modifies a database’s schema.

To modify a sequence:

1. Make sure that the database containing the sequences you want to modify is the current working database.

2. Choose Schema → Sequence Editor. The Sequence Editor lists all the sequences defined for the working database and the attributes for the sequence that is selected.

3. Select the sequence you want to modify.

4. Choose Modify.

5. Edit the sequence definition, then press GO.

Deleting a sequence

Deleting a sequence modifies a database’s schema.

To delete a sequence:

1. Make sure that the database containing the sequences you want to delete is the current working database.

2. Choose **Schema** → **Sequence Editor**. The **Sequence Editor** lists all the sequences defined for the working database and the attributes for the sequence that is selected.

3. Select the sequence you want to delete.

4. Choose the **Delete** option from the menu at the bottom of the window. The **Sequence Editor** prompts you to verify that you want to remove the sequence.

5. Choose **Yes** to delete the sequence. The **Sequence Editor** main window appears.

6. Choose **Exit**. The Data Dictionary main window appears.
Dumping sequence definitions

You can dump the sequence definitions for the current working database.

To dump sequence definitions:

1. Choose Admin → Dump Data and Definitions → Sequence Definitions. The Dump Sequence Definitions dialog box appears.
2. Specify the filename or use the default value. The default filename is _seqdefs.df.
3. Specify the code page or use the default value, then choose OK.
   After you dump the sequence values, the Data Dictionary displays a status message and prompts you to continue.
4. Choose OK to return to the Data Dictionary main window.
Dumping sequence current values

You can dump the sequence values for the current working database.

To dump current sequence values:

1. Make sure that the database containing the sequence values you want to dump is the current working database.

2. Choose Admin → Dump Data and Definitions → Sequences Current Values. The Dump Sequence Current Values dialog box appears.

3. Specify the filename or use the default value. The default filename is _seqvals.d.

4. Specify the code page or use the default value, then choose OK.

   After you dump the sequence values, the Data Dictionary displays a status message and prompts you to continue.

5. Choose OK to return to the Data Dictionary main window.
Loading sequence current values

The Data Dictionary uses a contents file that you specify to load sequence values into your database. Usually, this file is called _seqvals.d file.

The loading of sequence current values can accommodate 64-bit sequence values, provided the database has been enabled for 64-bit sequence support.

To load sequence values into a database:

1. Make sure that the working database is the target database where you want to load the sequence values.

2. Choose Admin→ Load Data and Definitions→ Sequences Current Values. The Data Dictionary prompts you for the filename from which you want to write the sequence values.

3. Specify the filename or use the default value, and choose OK. After the sequence values are loaded, the Data Dictionary displays a status message and prompts you to continue.

4. Choose OK to return to the Data Dictionary main window.
Generating a sequence report

A sequence report lists the sequence’s:

- Name
- Initial value
- Incremental value
- Maximum and minimum value

It also indicates whether the sequence should cycle.

To generate a sequence report:

1. Make sure that the working database is the target database for which you want to generate the report.

2. Choose Database → Reports → Sequence. The Sequence Report dialog box appears and displays information about all sequences defined for the working database.

3. Choose Print if you want to print the report. The Print Options dialog box appears.

4. Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.

5. Specify the page length, then choose OK. The Data Dictionary sends the report to the specified output destination and returns you to the Sequence Report dialog box.

6. Choose OK to return to the Data Dictionary main window.
Trigger Tasks

Tasks involved in creating and maintaining schema triggers are described in the following sections:

- Creating a trigger
- Modifying a trigger
- Deleting a trigger
- Generating a trigger report
Creating a trigger

There are two types of triggers: session and schema. You create session triggers as part of an ABL procedure, and you create schema triggers using the Data Dictionary.

There are two types of schema triggers:

- Table trigger
- Field trigger

To create a table trigger:

1. Choose Schema → Add New Table or choose Schema → Modify Table. If you are adding a new table, specify a table name. If you are modifying a table, the Data Dictionary alphabetically lists all the tables defined for your database. Select the table for which you want to create the trigger.

2. Choose Triggers. The Table Triggers dialog box appears.

3. Define the trigger attributes. For more information on the Table Triggers dialog box and the trigger attributes, see the “Table Triggers option” section on page 9–5.

4. Specify whether a session trigger can override this schema trigger. For more information about session and schema triggers, see the chapter on database triggers in OpenEdge Getting Started: ABL Essentials.

5. Specify whether the database engine checks the cyclical redundancy check (CRC) code for the trigger.

6. Press PUT to enter the trigger code. The Data Dictionary displays the Trigger Code dialog box.

7. Enter the trigger code.

8. Choose OK to return to the Table Triggers dialog box.

9. Choose OK to return to the Add Table or Modify Table window.

To create a field trigger from an existing field:

1. Choose Schema → Field Editor. The Field Editor alphabetically lists all the tables defined for your database.

2. Select the table that contains the field for which you want to create the trigger. The Field Editor window appears.

3. Choose Triggers.
4. Select the field for which you want to define the trigger. The **Field Triggers** dialog box appears:

| Trigger Programs | Override- | Check
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>able?</td>
<td>CRC?</td>
</tr>
<tr>
<td>For ASSIGN: (removing the trigger file name deletes the trigger)</td>
<td>no, yes</td>
<td></td>
</tr>
<tr>
<td>(press F61 to edit trigger program)</td>
<td>(OK) (Cancel)</td>
<td></td>
</tr>
</tbody>
</table>

5. Specify the name of the trigger procedure, then press **PUT** to enter the trigger code. The **Trigger Editor** appears.

6. Specify whether a session trigger can override the schema trigger. For more information about session and schema triggers, see the chapter on database triggers in *OpenEdge Getting Started: ABL Essentials*.

7. Specify whether the database engine verifies the cyclical redundancy check (CRC) code.

8. Choose **OK** to create the trigger and return to the **Field Editor**.

**Note:** To generate r-code for schema trigger definitions, you must define the trigger procedure in the Data Dictionary, then compile and save the r-code with the Application Compiler.
Modifying a trigger

Editing either a table trigger or a field trigger modifies the database’s schema.

To modify a table trigger:

1. Choose Schema → Modify Table. The Data Dictionary alphabetically lists all the tables defined for your database.

2. Select the table for which you want to modify the trigger. The Modify Table window appears.

3. Choose Triggers. The Table Triggers dialog box appears.

4. Select the trigger you want to modify.

5. Press PUT to enter the trigger code. The Trigger Code dialog box appears.

6. Edit the trigger code.

7. Choose OK to return to the Table Triggers dialog box.

8. Choose OK to return to the Modify Table window.

To modify a field trigger:

1. Choose Schema → Field Editor. The Field Editor alphabetically lists all the tables defined for your database.

2. Select the table for which you want to modify the trigger. The Field Editor appears.

3. Choose Triggers.

4. Select the field for which you want to modify the trigger. The Field Triggers dialog box appears.

5. Specify the name of the trigger procedure, then press PUT to enter the trigger code. The Trigger Code window appears.

6. Edit the trigger code.

7. Choose OK to return to the Field Triggers dialog box.

8. Choose OK to return to the Field Editor.
Deleting a trigger

You can delete both table triggers or field triggers.

**Note:** When you delete a trigger from the Data Dictionary, you do not delete the trigger procedure itself.

**To delete a table trigger:**

1. Choose Schema → Modify Table. The Data Dictionary alphabetically lists all the tables defined for your database.
2. Select the table for which you want to delete the trigger. The Modify Table dialog box appears.
3. Choose Triggers. The Table Triggers dialog box appears.
4. Delete the trigger filename for the trigger you want to remove.
5. Choose OK to return to the Modify Table dialog box.

**To delete a field trigger:**

1. Choose Schema → Field Editor. The Field Editor alphabetically lists all the tables defined for your database.
2. Select the table for which you want to delete the trigger. The Field Editor appears.
3. Choose Triggers.
4. Choose the field for which you want to delete the trigger. The Field Triggers dialog box appears.
5. Delete the trigger filename for the trigger you want to remove.
6. Choose OK in the Field Triggers dialog box to return to the Field Editor.
Generating a trigger report

A trigger report lists:

- The table/field name
- The event
- The procedure filename

The report also indicates if the CRC should be checked and whether the trigger is overridable or has mismatched CRC, or if there is no r-code found.

To generate a trigger report:

2. Choose the Print button if you want to print the report. The Print Options dialog box appears.
3. Specify the report destination. If you want to save the report to a file, specify the filename and whether to append the report to an existing file.
4. Specify the page length, then choose OK. The Data Dictionary sends the report to the specified output destination and returns you to the Trigger Report dialog box.
5. Choose OK to return to the Data Dictionary main window.
The Data Dictionary lets you create and modify database schema, including tables, fields, sequences, indexes, and triggers. You can also generate schema and auditing reports and perform database administration tasks.

This chapter describes the following options:

- Database menu
- Select Working Database option
- Create option
- Connect option
- Disconnect option
- Reports option
- Exit option
Database menu

From the Database menu you can:

- Select a working database
- Switch to a different working database
- Create new databases
- Connect databases
- Disconnect databases
- Generate schema or auditing reports
- Exit from the Data Dictionary

Table 8–1 describes the options that appear when you open the Database menu. The sections following the table contain detailed descriptions of these options.

Table 8–1: Database menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Working Database</td>
<td>Specifies the working database</td>
</tr>
<tr>
<td>Create</td>
<td>Creates a new database</td>
</tr>
<tr>
<td>Connect</td>
<td>Connects to an existing database</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects a database</td>
</tr>
<tr>
<td>Reports</td>
<td>Generates reports about schema, and provides access to reports on</td>
</tr>
<tr>
<td></td>
<td>the Alternate Buffer Pool, auditing, and encryption policies</td>
</tr>
<tr>
<td>Exit</td>
<td>Exits the Data Dictionary</td>
</tr>
</tbody>
</table>
Select Working Database option

The *Database → Select Working Database* menu option allows you to specify the working database if you have more than one database connected. When you choose this option, the *Select Working Database* dialog box appears, as shown in Figure 8–1.

![Select Working Database dialog box](image)

### Figure 8–1: Select Working Database dialog box

The Data Dictionary lists the available databases and basic database information. If you name a database with the PRO startup command, that database is connected and is the working database.
Create option

Use the Database→Create menu option to create an OpenEdge database. When you choose this option, the Create Database dialog box appears, as shown in Figure 8–2.

![Create Database dialog box](image)

**Figure 8–2: Create Database dialog box**

The Create Database dialog box contains the following fields:

- **New Physical Database Name** — Specifies the actual name for the database on a disk. This name can contain an explicit path or it can be relative to your propath.
- **Files** — Displays a list of available options for its corresponding field. For example, when you choose Files next to the New Physical Database Name field, the Find Database File dialog box appears.
- **Start with** — Specifies the type of database you want to create. You can create:
  - An EMPTY Database
  - A Copy of the SPORTS Database
  - A Copy of the Sports2000 Database
  - A Copy of Some Other Database
- **Replace If Exists** — Indicates that if you specify the name of an existing database, you want to replace the existing database with the one you are creating.
- **New Instance** — Indicates that the Data Dictionary will create a unique DB identifier for the created database, if it is a copy of another database. If you are starting with a copy of a database and do not select New Instance, the Data Dictionary assigns the source database’s DB Identifier to the copy.

Database copies created for backup purposes typically do not need to be created as new instances. If the copy is of an audit-enabled database and will be used for storing new audit data, select New Instance, as audit-enabled databases should have unique DB identifiers. For more information about auditing, see *OpenEdge Getting Started: Core Business Services*. 

---

8–4
Connect option

Use the Database → Connect menu option to connect to an existing database. After you choose the Connect option, the Connect Database dialog box, as shown in Figure 8–3, appears. Note that the Connect Database dialog box also appears after you create a database using the Database → Create menu option.

![Connect Database dialog box](image)

**Figure 8–3: Connect Database dialog box**

The Connect Database dialog box contains the following fields:

- **Physical Name** — Specifies the actual name of the database on a disk. This name can contain an explicit path or it can be relative to your propath.

- **Logical Name** — Specifies the database name that references a connected physical database. For example, if the physical database name is mydb1.db, the default logical database name is mydb1.

- **Database Type** — Specifies the database type. By default, the database type is PROGRESS.

- **Multiple Users** — Specifies if multiple users can access this database at the same time. The default is single-user. An x indicates multiple users can access the database. Before you can connect to a multi-user database, a server must be started for it using the PROSERVE command.

  **Note:** You must select this option if you are running a multi-user server and you want to connect to a self-service client.

- **Passphrase** — For databases enabled for transparent data encryption, specifies that a user wants to be prompted for a key store passphrase.

---

8–5
• **User ID** — Identifies your user ID. Whether you enter a user ID and password depends on how your security is configured. If you created a new database from an empty database or the SPORTS database, you do not have to enter any values for user ID or password because, by default, the security for these databases is not enabled. However, if you created a database from a copy of another database that has a list of user IDs and passwords in the user list, you must specify a valid user ID and password.

The Data Dictionary searches for the entered identification and passwords and if a match is found in the user list, it assigns the user ID to the new session. Otherwise, you will not be allowed you to connect to the database. For more information about user IDs and passwords, see *OpenEdge Data Management: Database Administration*.

• **Password** — Identifies your password.

• **Trigger Location** — Specifies the directory or r-code library containing the trigger code for the database.

• **Parameter File** — Specifies the parameter filename that contains the startup parameters for the database. The parameter file can include any number of startup parameters. For more information about parameter files, see *OpenEdge Data Management: Database Administration* and *OpenEdge Deployment: Managing ABL Applications*.

• **Other CONNECT Statement Parameters** — Specifies any other startup parameters for the database that are not included in the parameter file. For example, enter any parameters you want to use that are not defined by this dialog box (or by the parameter file, if you included one in the **Parameter File** area).

Internally, the Data Dictionary constructs and executes a CONNECT statement using the information you supply. Therefore, any rules that apply to the CONNECT statement also apply to database connections using the Data Dictionary. See *OpenEdge Getting Started: ABL Essentials* for more information about the CONNECT statement. For more information about connection parameters, see *OpenEdge Deployment: Startup Command and Parameter Reference*.
Disconnect option

Choose this option to disconnect a database. When you have more than one database connected, the Data Dictionary displays the **Select Database to Disconnect** dialog box. After you select the database you want to disconnect, the Data Dictionary prompts you for verification. If you have two databases connected and you disconnect the working database, the Data Dictionary automatically switches to the other database. If you have more than two databases connected and you disconnect the working database, the Data Dictionary displays the **Select Working Database** dialog box and prompts you to select the new working database.
The **Database→Reports** menu option allows you to generate reports about the schema in the database. Table 8–2 describes the submenu that appears when you choose this option.

### Table 8–2: Reports menu

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detailed Table</td>
<td>Displays details about one or all tables in this database, including field,</td>
</tr>
<tr>
<td></td>
<td>sequence, and index information</td>
</tr>
<tr>
<td>Quick Table</td>
<td>Displays a summary of all the tables in this database</td>
</tr>
<tr>
<td>Quick Field</td>
<td>Displays field information for one or all tables defined in the working</td>
</tr>
<tr>
<td></td>
<td>database</td>
</tr>
<tr>
<td>Quick Index</td>
<td>Displays index information for one or all tables defined in the working</td>
</tr>
<tr>
<td></td>
<td>database</td>
</tr>
<tr>
<td>PRO/SQL View</td>
<td>Displays PRO/SQL view information for all the views defined in the working</td>
</tr>
<tr>
<td></td>
<td>database</td>
</tr>
<tr>
<td>Sequence</td>
<td>Displays sequence information for all the sequences defined in the working</td>
</tr>
<tr>
<td></td>
<td>database</td>
</tr>
<tr>
<td>Trigger</td>
<td>Displays trigger information for all the triggers defined in the working</td>
</tr>
<tr>
<td></td>
<td>database</td>
</tr>
<tr>
<td>User</td>
<td>Displays the current database users, user IDs, and whether each user has a</td>
</tr>
<tr>
<td></td>
<td>password</td>
</tr>
<tr>
<td>Table Relations</td>
<td>Displays the relations for one or all tables defined in the working database</td>
</tr>
<tr>
<td>Storage Areas</td>
<td>Displays the storage area information for all storage areas defined in the</td>
</tr>
<tr>
<td></td>
<td>working database</td>
</tr>
<tr>
<td>Verify Data Width</td>
<td>Displays information about data that exceeds either the Data Dictionary</td>
</tr>
<tr>
<td></td>
<td>format or width field</td>
</tr>
<tr>
<td>Alternate Buffer Pool</td>
<td>Displays information about ABL objects assigned to the Alternate Buffer pool</td>
</tr>
<tr>
<td>Auditing Reports</td>
<td>A submenu of auditing reports; see Table 8–3 for a list of auditing reports</td>
</tr>
<tr>
<td>Encryption Policy Reports</td>
<td>A submenu of encryption policy reports; see Table 8–4 for a list of encryption policy reports</td>
</tr>
</tbody>
</table>
Detailed Table Report option

Use the Reports → Detailed Table option to generate information about the structure of the selected table or of all the tables in the working database. This option creates a report that displays all information about the tables, including information about fields and indexes defined for the tables.

When you choose this option and you select a table, the Data Dictionary displays the Report Options dialog box, as shown in Figure 8–4.

![Report Options dialog box](image)

**Figure 8–4: Report Options dialog box**

The Report Options dialog box contains the following fields:

- **Send Output to** — Allows you to send output to the screen, the default printer for your computer system, or a file.

  If you choose to send output to a file, specify the filename and whether to append the report information to an existing file. Choose Files to display a list of existing files.

- **Page Length** — Specifies the number of lines per page. The default for the printer is 60 lines. For the terminal or a file, the default is 0 (continuous).

- **Order Fields** — Specifies whether to order the fields for the report by order number or alphabetically.

  Figure 8–5 shows a portion of a Detailed Table Report for all the tables in the Sports database.

![Detailed Table Report dialog box](image)

**Figure 8–5: Detailed Table Report dialog box**
Quick Table Report option

Choose the Reports→Quick Table option to display summary information about all the tables in the working database. For each table the Quick Table Report lists the table name, storage area number, dump name, table flags, field count, index count, and table label. Figure 8–6 shows an example of a Quick Table Report for the Sports database.

![Quick Table Report](image)

Figure 8–6: Quick Table Report dialog box

Quick Field Report option

Choose Reports→Quick Field to display information about the structure of fields in one or all tables in the working database. For each field, the report lists the order number, name, data type, flags, and format.

The Data Type column lists the first four characters of the field’s data type. If the type is decimal, the report includes the number of stored decimals digits (for example, deci–2). If the field is an array, the report includes the extent (for example, char[10]). The Flags column indicates whether the field is case sensitive, mandatory, an index component, or a view component.

Figure 8–7 shows an example of a Quick Field Report for the customer table in the Sports database. The report includes the data type, flag, and format of each field.

![Quick Field Report](image)

Figure 8–7: Quick Field Report dialog box
The **Quick Field Report** dialog box contains the following fields:

- **Switch Tables** — Returns you to the list of tables for the working database
- **Change Field Order** — Changes the sort order of the fields to either alphabetical by name or numeric by order
- **Print** — Enables you to print the report or write it to a file

### Quick Index Report

Choose **Reports**→**Quick Index** to display information about the indexes defined in one or all tables in the working database. For each table, the report lists the index flags, name of the index, storage area number, and fields that define the index. The **Flags** column indicates whether the indexes are primary, unique, word, abbreviated, or inactive. A plus sign (+) or minus sign (-) next to the field name indicates if the index is ascending or descending. **Figure 8–8** shows an example of a Quick Index Report for the customer table of the Sports database.

![Quick Index Report](image)

**Figure 8–8: Quick Index Report dialog box**

The **Quick Index Report** dialog box contains the following fields:

- **Switch Tables** — Returns you to the list of tables for the working database
- **Print** — Enables you to print the report or write it to a file
Sequence Report option

Choose Reports → Sequence to display information about all sequences defined in the current database. For each sequence, it shows the name of the sequence, its initial value, increment value, minimum or maximum value, and whether the sequence cycles or stops at the maximum value. Figure 8–9 shows an example of a Sequence Report for the Sports database.

<table>
<thead>
<tr>
<th>Sequence Name</th>
<th>Initial Value</th>
<th>Increment</th>
<th>Max/Min Value</th>
<th>Cycle?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next-Cust-Num</td>
<td>1000</td>
<td>5</td>
<td>?</td>
<td>no</td>
</tr>
<tr>
<td>Next-Inv-Num</td>
<td>1000</td>
<td>1</td>
<td>?</td>
<td>no</td>
</tr>
<tr>
<td>Next-Item-Num</td>
<td>1000</td>
<td>10</td>
<td>?</td>
<td>no</td>
</tr>
<tr>
<td>Next-Ord-Num</td>
<td>1000</td>
<td>5</td>
<td>?</td>
<td>no</td>
</tr>
<tr>
<td>Next-Ref-Num</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>no</td>
</tr>
</tbody>
</table>

Figure 8–9: Sequence Report dialog box

Choose Print to print the report or to write it to a file.

Trigger Report option

Choose Reports → Trigger to list in the report all tables and fields that have triggers defined for them. The report lists the table or field name for which the trigger is defined, the type of event that causes the trigger to fire; whether the Dictionary checks the cyclical redundancy check (CRC) code for the trigger procedure; the trigger flags; and the name of the procedure. The Flags column indicates if the trigger is overridable, has a mismatched CRC code, or has no r-code available. Figure 8–10 shows an example of a Trigger Report for the Sports database.

<table>
<thead>
<tr>
<th>Table/Field Name</th>
<th>Event</th>
<th>Check</th>
<th>CRC</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>CREATE no</td>
<td>sports/crcust.p</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DELETE no</td>
<td>sports/delcust.p</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WRITE no</td>
<td>sports/wrcust.p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>CREATE no</td>
<td>sports/crinv.p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Invoice</td>
<td>DELETE no</td>
<td>sports/delinv.p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>CREATE no</td>
<td>sports/critem.p</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flags: * = overridable, ’m’ = mismatched crc, ’nn’ = no r-code found

Figure 8–10: Trigger Report dialog box

Choose Print to print the report or to write it to a file.

User Report option

Choose Reports → User to display the list of users who have permission to use the working database. This information is stored in the user list in the _User file. Figure 8–11 shows an example of a User Report window.
Reports option

Figure 8–11: User Report dialog box

Choose Print to print the report or to write it to a file.

Table Relations Report option

Choose Reports→Table Relations to list the implicit join relations for one or all tables in the working database. Figure 8–12 shows an example of a Table Relations Report for the Customer table of the Sports database.

Figure 8–12: Table Relations Report dialog box

The Table Relations Report dialog box contains the following fields:

- **Switch Tables** — Returns you to the list of tables for the working database
- **Print** — Enables you to print the report or write it to a file
Storage Areas Report option

Choose Reports → Storage Areas to list the storage areas in the working database. The Storage Areas report lists storage area number, type, block size, total extents, and area name. Figure 8–13 shows an example of a Quick Storage Area Report for all tables of the Sports database.

![Quick Storage Area Report](image)

Figure 8–13: Quick Storage Area Report dialog box

Choose Print to print a copy of the report.

Verify Data Width Report option

Choose Reports → Verify Data Width to compare the actual size of the data in a file to the definition of the corresponding field. The report shows the number of records that exceed the criteria selected, specifies the data type, and identifies the length of the longest element to exceed the length criteria. Figure 8–14 shows an example of a Data Width Report for all tables of the Sports database.

![Data Width Report](image)

Figure 8–14: Data Width Report dialog box

The Print option allows you to either print the report or write it to a file.
Alternate Buffer Pool Report option

Choose Reports → Alternate Buffer Pool to view the ABL objects assigned to the Alternate Buffer Pool. The report lists all the database objects that are assignable to the Alternate Buffer pool at the object-level and are currently assigned to the Alternate Buffer Pool.

![Quick Alternate Buffer Pool Report](image)

**Figure 8–15: Quick Alternate Buffer Pool Report**

The Print option allows you to either print the report or write it to a file.

Objects are assigned to the Alternate Buffer Pool as follows:

- For ABL, objects use Admin → Alternate Buffer Pool → Alternate Buffer Pool Maintenance.

- For SQL, objects are assigned with SQL syntax. For information on the SQL syntax, see *OpenEdge Data Management: SQL Reference*.

- For Type I and Type II areas, areas are assigned with the PROUTIL ENABLEB2 utility. For information on PROUTIL ENABLEB2, see *OpenEdge Data Management: Database Administration*.

For general information on the Alternate Buffer Pool, see *OpenEdge Data Management: Database Administration*. 
Auditing Reports option

If you are connected to an audit-enabled database, you can access the Auditing Reports submenu by choosing Reports→ Auditing Reports. For a description of each auditing report, see Table 8–3. For a complete description of OpenEdge auditing events and event IDs, see OpenEdge Getting Started: Core Business Services.

Table 8–3: Auditing reports

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track Audit Policy Changes</td>
<td>Details audit policy maintenance, commit, and dump/load events—events with the following IDs are tracked by this report: 300, 301, 10010, 10303, and 10304</td>
</tr>
<tr>
<td>Track Database Schema Changes</td>
<td>Details schema modification actions such as creating tables and modifying sequence definitions—events with IDs between 5000 and 5041 are tracked by this report</td>
</tr>
<tr>
<td>Track Audit Data Administration (Dump/Load)</td>
<td>Details the dump, load, and archival of audit data. Events with the following IDs are tracked by this report: 10300, 10301, and 10302</td>
</tr>
<tr>
<td>Track Application Data Administration (Dump/Load)</td>
<td>Details the dump and load of application data—events with the following IDs are tracked by this report: 10213 and 10214</td>
</tr>
<tr>
<td>Track User Account Changes</td>
<td>Details create, update, and delete events, as well as the user who made the changes—events with the following IDs are tracked by this report: 100, 101, and 102</td>
</tr>
<tr>
<td>Track Security Permissions Changes</td>
<td>Details role maintenance and permissions maintenance auditing events—events with IDs between 510 and 517 are tracked by this report</td>
</tr>
<tr>
<td>Track SQL Permissions Changes</td>
<td>Details create, update, and delete events, as well as the user who made the change—events with IDs of 210 through 212 and 400 through 422 are tracked by this report</td>
</tr>
<tr>
<td>Track Authentication System Changes</td>
<td>Details authentication system maintenance events and authentication system domain maintenance events—events with IDs between 500 and 507 are tracked by this report</td>
</tr>
<tr>
<td>Client Session Authentication Report</td>
<td>Details application access through authentication domains</td>
</tr>
<tr>
<td>Database Administration Report (Utilities)</td>
<td>Details activities performed by command-line database administration utilities—events with IDs between 10100 and 10212 are tracked by this report</td>
</tr>
<tr>
<td>Database Access Report (Login/Logout/etc)</td>
<td>Details login and logout events performed on an OpenEdge database—events with IDs between 10500 and 10611 are tracked by this report</td>
</tr>
</tbody>
</table>
The Track Audit Policy Changes report details audit policy maintenance, commit, and dump/load events.

To generate a Track Audit Policy Changes report:

1. Choose Reports → Auditing Reports → Track Audit Policy Changes. The Date Range dialog box appears.

2. To specify a range of dates on which to report, select Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, select Select All.

3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.

The Track Database Schema Changes report details schema modification actions such as creating tables and modifying sequence definitions.

To generate a Track Database Schema Changes report:

1. Choose Reports → Auditing Reports → Track Database Schema Changes. The Date Range dialog box appears.

2. To specify a range of dates on which to report, choose Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, choose Select All.

3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.

Table 8–3: Auditing reports

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track Encryption Policy Changes</td>
<td>Reports any events related to encryption policy maintenance—events with IDs 11400-11402, 11500-11502, and 11600-11602 are tracked by this report</td>
</tr>
<tr>
<td>Track Key-store Changes</td>
<td>Reports any events related to the key store and autostart—events with IDs 11100-11114 and 11200-11207 are tracked by this report</td>
</tr>
<tr>
<td>Database Encryption Administration (Utilities)</td>
<td>Reports events such as enabling and disabling encryption for your database, and scan and update utilities—events with IDs 11000, 11001, 11300, and 11301 are tracked by this report</td>
</tr>
<tr>
<td>Custom Audit Data Filter Report</td>
<td>Allows you to limit the amount of custom audit data on which the Data Dictionary reports</td>
</tr>
</tbody>
</table>
Track Audit Data Administration (Dump/Load) report

The Track Audit Data Administration (Dump/Load) report details the dump, load, and archival of audit data.

To generate a Track Audit Data Administration (Dump/Load) report:

1. Choose the Reports → Auditing Reports → Track Audit Data Administration (Dump/Load). The Date Range dialog box appears.
2. To specify a range of dates on which to report, choose Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, choose Select All.
3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.

Track Application Data Administration (Dump/Load) report

The Track Application Data Administration (Dump/Load) report details the dump and load of application data.

To generate a Track Application Data Administration (Dump/Load) report:

1. Choose Reports → Auditing Reports → Track Application Data Administration (Dump/Load). The Date Range dialog box appears.
2. To specify a range of dates on which to report, choose Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, select Select All.
3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.

Track User Account Changes report

The Track User Account Changes report details changes to the _User table and lists the user ID of the user who made the changes.

To generate a Track User Account Changes report:

1. Choose Reports → Auditing Reports → Track User Account Changes. The Date Range dialog box appears.
2. To specify a range of dates on which to report, select Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, select Select All.
3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.
Track Security Permissions Changes report

The Track Security Permissions Changes report details role maintenance and permissions maintenance events.

To generate a Track Security Permissions Changes report:

1. Choose Reports→ Auditing Reports→ Track Security Permissions Changes. The Date Range dialog box appears.
2. To specify a range of dates on which to report, select Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, select Select All.
3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.

Track SQL Permissions Changes report

The Track SQL Permissions Changes report details create, update, and delete events, as well as the user ID of the user who made the change.

To generate a Track SQL Permissions Changes report:

1. Choose Reports→ Auditing Reports→ Track SQL Permissions Changes. The Date Range dialog box appears.
2. To specify a range of dates on which to report, choose Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, select Select All.
3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.

Track Authentication System Changes report

The Track Authentication System Changes report details authentication system maintenance events and authentication system domain maintenance events.

To generate a Track Authentication System Changes report:

1. Choose Reports→ Auditing Reports→ Track Authentication System Changes. The Date Range dialog box appears.
2. To specify a range of dates on which to report, choose Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, choose Select All.
3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.
Client Session Authentication report

The Client Session Authentication report details application access through authentication domains.

To generate a Client Session Authentication report:

1. Choose Reports → Auditing Reports → Client Session Authentication. The Date Range dialog box appears.
2. To specify a range of dates on which to report, select Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, select Select All.
3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.

Database Administration Utilities report

The Database Administration Utilities report details activities performed by command-line database administration utilities.

To generate a Database Administration Utilities report:

1. Choose Reports → Auditing Reports → Database Administration Report (Utilities). The Date Range dialog box appears.
2. To specify a range of dates on which to report, select Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, select Select All.
3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.

Database Access report (Login/Logout/etc)

The Database Access report details login and logout events performed on an OpenEdge database.

To generate a Database Access report (Login/Logout/etc):

1. Choose Reports → Auditing Reports → Database Access Report (Login/Logout/etc). The Date Range dialog box appears.
2. To specify a range of dates on which to report, choose Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, choose Select All.
3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information.
Track Encryption Policy Changes report

The Track Encryption Policy Changes report details encryption policy maintenance changes.

To generate a Track Encryption Policy Changes report:

1. Choose Reports → Auditing Reports → Track Encryption Policy Changes. The Date Range dialog box appears.

2. To specify a range of dates on which to report, select Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, select Select All.

3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.

Track Key-store Changes report

The Track Key-store Changes report details any changes to the key store and the autostart configuration.

To generate a Track Key-store Changes report:

1. Choose Reports → Auditing Reports → Track Key-store Changes. The Date Range dialog box appears.

2. To specify a range of dates on which to report, select Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, select Select All.

3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.

Database Encryption Administration (Utilities) report

The Database Encryption Administration (Utilities) report details events such as enabling and disabling encryption, and running the encryption policy scan and update utilities.

To generate a Database Encryption Administration (Utilities) report:

1. Choose Reports → Auditing Reports → Database Encryption Administration (Utilities). The Date Range dialog box appears.

2. To specify a range of dates on which to report, select Select Some and enter the dates in the Start Range and End Range fields. To report on all audit policy records, select Select All.

3. Choose OK. The OpenEdge Auditing Report Options dialog box appears. See the “OpenEdge Auditing Report Options dialog box” section on page 8–23 for more information about the OpenEdge Auditing Report Options dialog box.
Custom Audit Data Filter report

The Custom Audit Data Filter report allows you to limit the amount of audit data reported on by the Data Dictionary.

To generate an audit report using the Custom Audit Data Filter:


2. To filter based on an audit date range, complete the following fields:
   - From Date — The date of the earliest audit transaction that you want reported
   - To Date — The date of the last audit transaction that you want reported

3. To filter based on audit event ID range, complete the following fields:
   - From Id — The first ID that you want reported
   - To Id — The last ID that you want reported

4. To report on audit events associated with a user ID, enter the ID into the Miscellaneous User Id field.

5. To report on audit events associated with a transaction ID, enter the ID into the Miscellaneous Transaction Id field.

6. To filter on event context information, complete the desired context field:
   - System Context — The system context that you want reported. The system context is typically the name of a table. Select SEP to insert a separator character. The separator character indicates a delimited string.
   - Policy Context — The policy context that you want reported. Select SEP to insert a separator character. The separator character indicates a delimited string.

Use the BEGINS, MATCHES, and EQUALS options to filter the Event Context information.

Note: If you filter on Event Context information, the report may take a long time to generate.
OpenEdge Auditing Report Options dialog box

The OpenEdge Auditing Report Options dialog box allows you to set display characteristics for your auditing report. The OpenEdge Auditing Report Options dialog box appears, and contains the following fields:

- **Send Output to** — Choose the output destination from the radio set: Terminal, File (XML), or File (Text).
- **File Name** — If File (XML) or File (Text) is the selected output destination, enter a file name for the report or browse for a file using Files.
- **Append to Existing File** — Select to append an existing file. If you do not select the append option, an existing file may be overwritten.
- **Page Length** — Enter a number. Zero (0) creates continuous output with no line breaks. The default is 60 lines per printed page. The default for a terminal is the number of lines displayed on the screen.
- **Orientation** — Select the report’s orientation: Detail or Summary. Select Detail if you want details for each of the report’s entries. Select Summary if you only want to print summary information.

Encryption Policy Reports option

If you are connected to an encryption-enabled database, you can access the Encryption Policy Reports submenu by choosing Reports → Encryption Policy Reports. You cannot access encryption policy reports for a remote database; you must have a local connection. For a description of the encryption policy reports see Table 8–4. For more information on encryption policies, see OpenEdge Getting Started: Core Business Services.

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quick Encryption Policies</strong></td>
<td>For ABL objects encrypted at the object level in a Type II area, this report details the cipher name and policy version of object’s current encryption policy.</td>
</tr>
<tr>
<td><strong>Detailed Encryption Policies</strong></td>
<td>For all encryptable ABL objects at the object level in a Type II area:  &lt;ul&gt;&lt;li&gt;If the object is encrypted, the policy version, cipher name, and policy state is reported.&lt;/li&gt;&lt;li&gt;If the object is not encrypted, “No policy information available for object” is reported.&lt;/li&gt;&lt;/ul&gt; <strong>Note:</strong> If ALL is selected, only encrypted objects are reported.</td>
</tr>
</tbody>
</table>
Quick Encryption Policies report

Choose Reports→ Encryption Policy Reports→ Quick Encryption Policies to generate a Quick Encryption Policy report. The report details the cipher name and policy version of an ABL object’s current encryption policy. Figure 8–16 shows an example of a Quick Encryption Policy report.

![Quick Encryption Policy Report dialog box](image)

The Print option allows you to either print the report or write it to a file.

Detailed Encryption Policies report

Choose Reports→ Encryption Policy Reports→ Detailed Encryption Policies to generate a Detailed Encryption Policy report. For all encryptable ABL objects at the object level, the detailed report contains the following information:

- If the object is encrypted, the policy version, cipher name, and policy state is reported
- If the object is not encrypted, “No policy information available for object” is reported

To generate a Detailed Encryption Policy report:

2. Select either a table name or ALL.
3. Select the report destination: terminal, printer, or file. Click OK.
4. The report is generated. If you chose terminal as the destination, the Print option allows you to either print the report or write it to a file.
Figure 8–17 shows an example of a Detailed Encryption Policy Report.

Figure 8–17: Detailed Encryption Policy Report dialog box
Exit option

Choose this option to close the Data Dictionary and you return to where you started the Dictionary. For example, if you started the Dictionary from the Procedure Editor, you return there.
Schema menu commands are used to modify the structure of a database. This chapter describes the following information related to the schema:

- Schema menu
- Modify Table option
- Add New Table option
- Delete Table(s) option
- Field Editor option
- Reorder Fields option
- Global Field Name Change option
- Index Editor option
- Sequence Editor option
- Adjust Field Width option
Schema menu

The schema of the database is the definition of its structure: the tables it contains, the fields within the tables, sequences, etc. In addition to database structure, schemas contain validation expressions and validation messages.

Use the Schema options to add, update, or delete table, field, index, or sequence definitions. Table 9–1 lists the Schema menu options and gives a brief description of each. The sections following the table contain detailed descriptions of these schemas.

Table 9–1: Schema menu

<table>
<thead>
<tr>
<th>Schema</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify Table...</td>
<td>Changes information defined for an existing table</td>
</tr>
<tr>
<td>Add New Table...</td>
<td>Creates a new table definition</td>
</tr>
<tr>
<td>Delete Table(s)...</td>
<td>Removes one or more table definitions</td>
</tr>
<tr>
<td>Field Editor...</td>
<td>Adds, alters, or removes fields from a table definition</td>
</tr>
<tr>
<td>Reorder Fields...</td>
<td>Resequences the fields into alphabetical order, or resets the order numbers of the fields</td>
</tr>
<tr>
<td>Global Field Name Change...</td>
<td>Changes a field name throughout the database</td>
</tr>
<tr>
<td>Index Editor...</td>
<td>Adds, alters, or removes indexes in a table</td>
</tr>
<tr>
<td>Sequence Editor...</td>
<td>Adds, alters, or removes sequences in a table</td>
</tr>
<tr>
<td>Adjust Field Width...</td>
<td>Alters field width value in a table</td>
</tr>
</tbody>
</table>
Modify Table option

The Schema → Modify Table option allows you to change the structure of a table.

When you choose this option, the Data Dictionary alphabetically lists all the tables defined for your database. After you select a table, a window similar to the one shown in Figure 9–1 appears.

Figure 9–1: Modify Table window

The Modify Table window contains the following fields:

- **Table Name** — Specifies the name of the table. Table names can be up to 32 characters long and can consist of alphabetic characters, digits, and the characters $, &, %, -, and _. In addition, table names must begin with a letter (A–Z or a–z). You cannot use reserved ABL keywords as table names. Table names are not case sensitive.

- **Area** — Displays the name of the storage area. The Area Name is assigned to a table when it is created. Because the Table Name cannot be updated, this field is disabled.

- **Dump File** — Specifies the file to which you want to dump the table contents. The name must be unique among the other tables in the database and can be up to 32 characters long. The default filename is the table name.

- **Table Type** — Displays the table type.

- **Hidden** — Specifies if the table is a hidden table. The default value is no.

- **Label** — Specifies the label used in error messages.

- **Frozen** — Specifies if the table is frozen. You can freeze or unfreeze a table using the Freeze/Unfreeze option from the Utilities menu. When a table is frozen, you can only view its properties. The actual record data is not frozen, only the data definitions.

- **Owner** — Displays the user ID of the table owner. This information is necessary for manipulating certain DataServers.

- **Record Size** — Specifies the size of the record used for DataServers with fixed-length records.

- **Replication** — Specifies the name of the procedure that performs the replication for this table. For more information, see the chapter on data replication in OpenEdge Data Management: Database Administration.
**DataServer Name** — Specifies the table name in the appropriate DataServer database if the table is not an OpenEdge table.

**DB Link** — Specifies the link name for the distributed ORACLE database.

**Description** — Describes the table contents.

The following sections describe the options that appear at the bottom of the window.

**Validation option**

The **Validation** option in the **Modify Table** dialog box allows you to specify validation criteria and a message for record deletions.

**Note:** You should use table validation for backward compatibility for procedure-driven applications. For event-driven programs, use table delete triggers instead of table validation.

If you choose **Validation**, the **Table Validation** dialog box shown in **Figure 9–2** appears.

![Table Validation dialog box](image)

**Figure 9–2:** Table Validation dialog box

The **Table Validation** dialog box contains the following fields:

- **Valmsg** — Specifies the message that appears when a user tries an invalid deletion. There is no limit to the number of characters you can enter, but a message up to 63 characters long can fit onto the two message lines without scrolling.

- **Valexp** — Specifies the criteria for valid deletions. For example, to restrict deleting customers in the Customer table to those customers who have no outstanding orders, enter the following:

  NOT (CAN-FIND(FIRST order OF customer))
Table Triggers option

The **Table Triggers** option in the Modify Table dialog box allows you to define schema triggers for a table. If you choose **Table Triggers**, the **Table Triggers** window shown in Figure 9–3 appears.

![Table Triggers](image)

The **Table Triggers** dialog box contains the following fields:

- **Overrideable** — Specifies whether a session trigger can override the schema trigger. The default value is **no**. See [OpenEdge Getting Started: ABL Essentials](#) for more information about schema and session triggers.

- **Check CRC** — Specifies whether the database engine verifies the cyclical redundancy check (CRC) code. The default value is **yes**. If this option is on, then the AVM stores in the metaschema, along with the trigger procedure name, a unique Cyclic Redundancy Check (CRC) identifier for the compiled version of the trigger procedure. The AVM raises an error if the r-code file it encounters at run time does not match or if there is no compiled version of the procedure.

- **Trigger Programs** — Choose the database event for the trigger you want to create. Table 9–2 lists the available database events.

![Table 9–2: Table Trigger events](image)

**Table 9–2: Table Trigger events**

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>Specifies that the trigger executes every time a CREATE or INSERT statement executes for the table you are modifying.</td>
</tr>
<tr>
<td>DELETE</td>
<td>Specifies that the trigger executes every time a DELETE statement executes for the table you are modifying.</td>
</tr>
<tr>
<td>FIND</td>
<td>Specifies that the trigger executes every time a record in the table you are modifying is read using a FIND statement or a FOR EACH loop. Only records that satisfy the full search conditions (such as a WHERE clause) cause a FIND trigger to execute. The CAN-FIND statement does not cause a FIND trigger to execute.</td>
</tr>
</tbody>
</table>
Press F6 to display the trigger code dialog box. Figure 9–4 shows an example of a delete trigger procedure.

**Table 9–2: Table Trigger events**

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRITE</td>
<td>Specifies that the trigger executes every time OpenEdge RDBMS validates a record. OpenEdge RDBMS automatically validates a record when it releases it. <strong>Note:</strong> OpenEdge RDBMS does not compare the new record with the old record when determining if it should fire the WRITE trigger. Even temporary changes to a table (as when a field is changed, then changed back to its original value) cause WRITE to fire. If it is necessary to handle this situation, use the NEW BUFFER and OLD BUFFER options in your trigger procedure to compare the values and detect a “dummy” change case.</td>
</tr>
<tr>
<td>REPLICATION-CREATE</td>
<td>Specifies the trigger that executes immediately following the execution of the CREATE trigger (if you specify a CREATE trigger). The purpose of the trigger is to replicate the creation of records to another database.</td>
</tr>
<tr>
<td>REPLICATION-DELETE</td>
<td>Specifies the trigger that executes immediately following the execution of the DELETE trigger (if you specify a DELETE trigger). The purpose of the trigger is to replicate the deletion of records to another database.</td>
</tr>
<tr>
<td>REPLICATION-WRITE</td>
<td>Specifies the trigger that executes immediately following the execution of the WRITE trigger (if you specify a WRITE trigger). The purpose of the trigger is to replicate updated records to another database.</td>
</tr>
</tbody>
</table>

![Trigger Code]

**Figure 9–4: Example of a delete trigger procedure**
String Attributes option

The String Attributes option allows you to control how much space is allocated within an r-code text segment for a string. It is important when you translate a string that you specify enough space for the translated string. String attributes can also specify the text justification within the r-code and whether or not the Translation Manager should translate the string. When you choose String Attributes, the Table String Attributes dialog box shown in Figure 9–5 appears.

![Table String Attributes dialog box](image)

The Table String Attributes dialog box contains the following fields:

- **Label** — Specifies the string attribute characters for the table label
- **Validation Message** — Specifies the string attribute characters for the table validation message

Use the following character and number combinations to enter the string attribute label:

```
[ R | L | C | T ] [ U ] [ max-length ]
```

See the “Specifying string attributes” section on page 3–3 for details about the control characters you can enter in the label field.
Table 9–3 lists several examples of string attributes.

<table>
<thead>
<tr>
<th>Example</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>L15</td>
<td>Specifies a width of 15 characters within the r-code text segment and left-justifies the string within the allocated space</td>
</tr>
<tr>
<td>15R</td>
<td>Specifies a width of 15 characters within the r-code text segment and right-justifies the string within the allocated space</td>
</tr>
<tr>
<td>T20</td>
<td>Specifies a width of 20 characters and trims the leading and trailing spaces within the allocated space</td>
</tr>
<tr>
<td>UC20</td>
<td>Specifies a width of 20 characters and centers the string within the allocated space, and also indicates that the Translation Manager should not translate the string</td>
</tr>
<tr>
<td>20RU</td>
<td>Specifies a width of 20 characters and right-justifies the string within the allocated space, and also indicates that the Translation Manager should not translate the string</td>
</tr>
<tr>
<td>30</td>
<td>Specifies a width of 30 characters</td>
</tr>
</tbody>
</table>

See *OpenEdge Development: ABL Reference* for more information about string attributes and string translation.
Add New Table option

Use the Schema → Add New Table menu option to create a new table definition. When you choose this option, the window shown in Figure 9–6 appears.

![Add New Table window](image)

**Figure 9–6: Add New Table window**

See the “Modify Table option” section on page 9–3 for a description of this window’s fields.
Delete Table(s) option

Use the **Schema → Delete Table(s)** menu option to delete one or more tables from the working database. The Data Dictionary prompts you for verification before it deletes the tables you select.
Field Editor option

Use the Schema → Field Editor menu option to modify schema definitions for a field. When you choose this option, the Field Editor alphabetically lists all the tables defined for your database. When you select the table you want to edit, the Field Editor lists all the fields defined for the selected table. Figure 9–7 shows the fields defined for the customer table.

Figure 9–7: Field Editor window

You can add, edit, delete, or view field attributes defined for the current table from the Field Editor. Table 9–4 describes the options for the Field Editor.

Table 9–4: Field Editor options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NextPage</td>
<td>Displays the next page of fields.</td>
</tr>
<tr>
<td>PrevPage</td>
<td>Displays the previous page of fields.</td>
</tr>
<tr>
<td>Add</td>
<td>Adds a new field.</td>
</tr>
<tr>
<td>Modify</td>
<td>Updates field attributes. Note: You can modify the Data Type field only if the field is an integer and you want to change it to INT64. For all other data types, the field is read-only. For more information, see the “Changing a field data type from integer to INT64” section on page 4–3.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes a field.</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies fields from table definitions.</td>
</tr>
<tr>
<td>Triggers</td>
<td>Defines or modifies triggers.</td>
</tr>
<tr>
<td>View-As</td>
<td>Defines the View-As phrase for field widget type.</td>
</tr>
<tr>
<td>StringAttrs</td>
<td>Defines or modifies string attributes.</td>
</tr>
<tr>
<td>GoIndex</td>
<td>Opens the Index Editor.</td>
</tr>
<tr>
<td>SwitchTable</td>
<td>Switches to a different table.</td>
</tr>
<tr>
<td>Browse</td>
<td>Displays the field attributes.</td>
</tr>
</tbody>
</table>
To use the Field Editor, choose an option, then select the field you want to edit. The following sections describe the Add, Triggers, View-As, and GoIndex options in more detail.

### Add

Use the Add option from the Field Editor dialog box to create a new field in the selected table. When you choose this option, a window similar to the one shown in **Figure 9–8** appears.

![Field Editor Add option window](image)

**Figure 9–8:** Field Editor Add option window

The Field Editor Add option window contains the following fields:

- **Field-Name** — Specifies the field name. Names can be up to 32 characters long and can consist of alphabetic characters, digits, and the characters $, &amp;, #, %, -, and _. Field names must begin with A–Z or a–z. You cannot use ABL keywords as field names. Field names are not case sensitive.

- **Format** — Specifies the length of a field and the way the data is shown on the screen and in printed reports. The Field Add Option window automatically supplies a default format for each data type, but you can change it. You can also override formats you define in the Data Dictionary by using the FORMAT option with the Format phrase. For more information about formatting, see *OpenEdge Getting Started: ABL Essentials*.
A description of the default format for each of the data types follows:

- **Character** — The default format for a character field is \( x(8) \). The \( x \) represents an alphanumeric character position and the 8 indicates the number of characters to display. Table 9–5 describes the different symbols you can use to define a character format.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x )</td>
<td>Represents any character.</td>
</tr>
<tr>
<td>( n )</td>
<td>Represents a digit or a letter. A space is not allowed.</td>
</tr>
<tr>
<td>( a )</td>
<td>Represents a letter. A space is not allowed.</td>
</tr>
<tr>
<td>!</td>
<td>Represents a letter that is converted to uppercase during input. A space is not allowed.</td>
</tr>
<tr>
<td>9</td>
<td>Represents a digit. A space is not allowed.</td>
</tr>
<tr>
<td>( (n) )</td>
<td>A number that indicates how many times to repeat the previous format character. For example, ( !(5) ) is the same as ( !!!!! ) and represents 5 characters that are to be converted to uppercase when entered.</td>
</tr>
<tr>
<td>fillchar</td>
<td>Represents any character or characters you want to fill a display. For example, if you display the value abc with a format of ( x(3)*** ), the displayed value is abc***. To use X, N, A, !, and 9 as fill characters, you must precede the character with a tilde (<del>). To use a left parenthesis ( ( ) as a fill character after a non-fill character, you must precede it with a tilde (</del>). See OpenEdge Getting Started: ABL Essentials for more information about fill characters.</td>
</tr>
</tbody>
</table>

Table 9–6 shows examples of the different formats of a character field.

<table>
<thead>
<tr>
<th>Format</th>
<th>Value in field</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>x(9)</td>
<td>These are characters</td>
<td>These are</td>
</tr>
<tr>
<td>x(20)</td>
<td>These are characters</td>
<td>These are characters</td>
</tr>
<tr>
<td>xxx</td>
<td>These are characters</td>
<td>The</td>
</tr>
<tr>
<td>AAA–9999</td>
<td>abc1234</td>
<td>abc–1234</td>
</tr>
<tr>
<td>(999) 999–9999</td>
<td>6172754500</td>
<td>(617) 275-4500</td>
</tr>
<tr>
<td>!!</td>
<td>ma</td>
<td>MA</td>
</tr>
</tbody>
</table>
Trailing spaces are truncated in character fields. If a character field contains only one space, it is truncated to a null value. You can use the TRIM function to truncate leading and trailing spaces.

- **Integer, INT64, and decimal** — The default display format for an integer or an INT64 field is –»,>>>>,>>9. The default display format for a decimal field is –>>,>>9.99. Table 9–7 describes the format characters for numeric display formats.

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( )</td>
<td>The Data Dictionary displays parentheses if the number is negative. If you use one parenthesis (left or right), you must use the other.</td>
</tr>
<tr>
<td>string1</td>
<td>A string made up of any characters except plus (+), minus (−), greater than (&gt;) or less than (&lt;), comma (,), digits (0–9), letter z (z or Z), asterisk (*), or period (.).</td>
</tr>
<tr>
<td>+</td>
<td>The Data Dictionary replaces this character with a plus sign if the number is positive, and a negative sign if the number is negative. You can also use only one plus or minus sign or CR, DR, or DB, or one set of parentheses in a numeric data format. The position of the plus sign in the number format can also hold a digit if the number is positive.</td>
</tr>
<tr>
<td>–</td>
<td>When you use this character to the left of the decimal point, the Data Dictionary replaces this character with a minus sign if the number is negative and a space or null if the number is positive. When you use this character to the right of the decimal point, the Data Dictionary replaces this character with a minus sign if the number is negative and a space if the number is positive.</td>
</tr>
<tr>
<td>&gt;</td>
<td>The Data Dictionary replaces this character with a digit if that digit is not a leading zero. If the digit is a leading zero, the Data Dictionary replaces this character with a null and moves characters to the left one space to the right, if you are using top labels. If you are using side labels, the Data Dictionary left-justifies the characters.</td>
</tr>
<tr>
<td>,</td>
<td>The Data Dictionary displays this character as a comma unless it is preceded by a &gt; or Z. If the comma is preceded by a &gt; and the &gt; is replaced by a leading zero, the Data Dictionary replaces the comma with a null. If the comma is preceded by a Z and the Z is replaced by a space, the Data Dictionary replaces the comma with a space.</td>
</tr>
<tr>
<td>9</td>
<td>The Data Dictionary replaces this character with a digit, including cases where the digit is a leading zero.</td>
</tr>
<tr>
<td>Z</td>
<td>The Data Dictionary replaces this character with a digit. If the digit is a leading zero, Z suppresses that digit and puts a space in its place.</td>
</tr>
<tr>
<td>*</td>
<td>The Data Dictionary replaces this character with a digit. If the digit is a leading zero, the Data Dictionary replaces that zero with an asterisk (*).</td>
</tr>
</tbody>
</table>
When specifying a numeric data format, you must use at least one of the following characters: 9, Z, *, or >.

Table 9–8 shows examples of the different formats for a numeric field.

<table>
<thead>
<tr>
<th>Format</th>
<th>Value in field</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>9999</td>
<td>123</td>
<td>0123</td>
</tr>
<tr>
<td>9,999</td>
<td>1234</td>
<td>1,234</td>
</tr>
<tr>
<td>$zzz9</td>
<td>123</td>
<td>$123</td>
</tr>
<tr>
<td>$&gt;&gt;&gt;9</td>
<td>123</td>
<td>$123</td>
</tr>
</tbody>
</table>

**Note:** This display value is right-justified if it has a column label, and left-justified if it has a side label.

<table>
<thead>
<tr>
<th>Format</th>
<th>Value in field</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-&gt;,&gt;&gt;&gt;9.99</td>
<td>1234</td>
<td>$1,234.00</td>
</tr>
<tr>
<td>$&gt;,&gt;&gt;&gt;9.99</td>
<td>1234</td>
<td>$1,234.00</td>
</tr>
<tr>
<td>#–zzz9.999</td>
<td>–12.34</td>
<td>#–12.340</td>
</tr>
<tr>
<td>Tot=&gt;&gt;&gt;9Units</td>
<td>12</td>
<td>Tot=12Units</td>
</tr>
</tbody>
</table>
Note: If you use the European Numeric Format (-E) startup parameter, the Data Dictionary interprets commas as decimal points and decimal points as commas when displaying or prompting for numeric values. However, always enter formats in the Data Dictionary as described in Table 9–8. For more information about formatting in procedures, see OpenEdge Getting Started: ABL Essentials.

**Date** — The default date format is mm/dd/yy. Date formats specify a two-digit month and a two-digit day. You can use a slash (/) or a hyphen (-) as a separator. You can specify a year with two or four digits.

When -yy is set at 1950, the default, the Data Dictionary determines if the two-digit value in the date is greater or less than 50. If the date value is greater than 50, the Data Dictionary writes the date for the twentieth century. If the date value is less than 50, the Data Dictionary writes the date for the twenty-first century. For example, if you start the Data Dictionary with -yy 1950, years 50-99 are treated as 1950-1999, and years 00-49 are treated as 2000-2049.

### Table 9–8: Numeric display format examples (2 of 2)

<table>
<thead>
<tr>
<th>Format</th>
<th>Value in field</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&gt;,&gt;&gt;9.99</td>
<td>-12.34</td>
<td>??? ?????????</td>
</tr>
<tr>
<td>Note: There is a negative sign in the value -12.34, but the display format of $&gt;,&gt;&gt;9.99 does not accommodate that sign.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$&gt;,&gt;&gt;9.99</td>
<td>1234567</td>
<td>??? ?????????</td>
</tr>
<tr>
<td>Note: The value 1234567 is too large to fit in the display format of $&gt;,&gt;&gt;9.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt;&gt;,&gt;99.99&lt;&lt;&lt;</td>
<td>12.345.6789</td>
<td>12.345.68</td>
</tr>
<tr>
<td>&gt;&gt;,&gt;99.99&lt;&lt;&lt;</td>
<td>1.234.5678</td>
<td>1.234.568</td>
</tr>
<tr>
<td>&gt;&gt;,&gt;99.99&lt;&lt;&lt;</td>
<td>123.45</td>
<td>123.45</td>
</tr>
<tr>
<td>&gt;&gt;,&gt;99.99&lt;&lt;&lt;</td>
<td>12.45678</td>
<td>12.45678</td>
</tr>
</tbody>
</table>

1. Floating-decimal display format. The < character must follow the decimal point and be balanced by an equal or greater number of > characters. The full value is displayed if it can fit in the display format. If the value is too large to fit in the display format, the decimal portion of the value will be rounded so that the value can fit.
Table 9–9 shows some date display format examples.

Table 9–9: Date display format examples

<table>
<thead>
<tr>
<th>Format</th>
<th>Value in field</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>99/99/99</td>
<td>8/10/2001</td>
<td>08/10/01</td>
</tr>
<tr>
<td>99/99/9999</td>
<td>08/10/2090</td>
<td>08/10/2090</td>
</tr>
<tr>
<td>99-99-99</td>
<td>08/10/1993</td>
<td>08-10-93</td>
</tr>
<tr>
<td>99-99-99</td>
<td>08/10/2090</td>
<td>???????</td>
</tr>
</tbody>
</table>

Note: The value 8/10/2090 is too large to fit into the display format. If a field’s value does not fit in the display format, then the Data Dictionary displays each integer as a ?.

<table>
<thead>
<tr>
<th>Format</th>
<th>Value in field</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>999999</td>
<td>08/10/1997</td>
<td>081097</td>
</tr>
<tr>
<td>999999</td>
<td>08/10/01</td>
<td>081001</td>
</tr>
<tr>
<td>99999999</td>
<td>08/10/2001</td>
<td>08102001</td>
</tr>
</tbody>
</table>

- **DATETIME** — Stores date values as year, month, and day. Stores time values as hours, minutes, seconds and milliseconds. Table 9–10 shows a DATETIME display format example.

Table 9–10: DATETIME display format examples

<table>
<thead>
<tr>
<th>Format</th>
<th>Value in field</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>99/99/99 HH:MM:SS.SSS</td>
<td>8/10/01/11:09.32.213</td>
<td>08/10/01/11:09.32.213</td>
</tr>
</tbody>
</table>

- **DATETIME-TZ** — Stores date and time values as offset from the Coordinated Universal Time. Table 9–11 shows a DATETIME-TZ display format example.

Table 9–11: DATETIME-TZ display format example

<table>
<thead>
<tr>
<th>Format</th>
<th>Value in field</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>99/99/9999HH:MM:SS.SSS+HH:MM</td>
<td>8/10/2003/11:09.32.213+02:00</td>
<td>Same as value in field</td>
</tr>
</tbody>
</table>
Logical — The default logical format is YES/NO. You can define any strings to represent those TRUE/FALSE values. If you define your own logical values, the false value cannot begin with “y” or “t”, and a true value cannot begin with “n” or “f”. If input is coming from a file and you have defined a format for a logical field that is something other than TRUE/FALSE or YES/NO, you can still use TRUE/FALSE or YES/NO as input to that logical field. Table 9–12 shows logical display format examples.

Table 9–12: Logical display format examples

<table>
<thead>
<tr>
<th>Format</th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes/no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Yes/No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>true/false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>shipped/waiting</td>
<td>shipped</td>
<td>waiting</td>
</tr>
</tbody>
</table>

Label — Specifies the label that represents the field in windows or in printed reports. The default value is a question mark (?). If you leave the question mark and do not supply a label, the Data Dictionary uses the field names as the label. If you replace the question mark with a space, the Data Dictionary uses no label for the field. Labels can be up to 30 characters long. Table 9–13 shows field label examples.

Table 9–13: Field label examples

<table>
<thead>
<tr>
<th>Format</th>
<th>Label you define</th>
<th>Label displayed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>?</td>
<td>Name</td>
</tr>
<tr>
<td>Curr-bal</td>
<td>Unpaid Bal</td>
<td>Unpaid Bal</td>
</tr>
<tr>
<td>Sales-rep</td>
<td>space</td>
<td>–</td>
</tr>
</tbody>
</table>

You can override the labels you define in the Data Dictionary by using LABEL, NO-LABEL, or COLUMN-LABEL options in a Format phrase, or NO-LABEL in a Frame phrase. The COLUMN-LABEL is used only for fields that do not have SIDE-LABEL. See OpenEdge Getting Started: ABL Essentials for more information about these phrases.
• **Column-label** — Specifies that you want to use a different label when the data are listed in columns. If you do not specify a column label, the Data Dictionary uses the label specified in the Label field. If you do not enter a label, the Data Dictionary uses the field name. You can override these labels by using NO-LABEL or COLUMN-LABEL options in a Format phrase, or NO-LABEL in a Frame phrase.

If you want the column label to have more than one line (stacked), separate each line of the label with an exclamation point (!). For example, if you want the label for the **Curr-bal** field to be Unpaid Balance, with the word “Unpaid” displayed above the word “Balance,” enter **Unpaid!Balance** for the column label. If you want to use the exclamation point as one of the characters in a column label, you must use two exclamation points (!!). Any spaces to the right or left of the exclamation point become part of the label.

• **Initial** — Defines the initial value for the field. Each data type has an initial value. When you create a new record for a table, each field contains this initial value. You can change a field’s default initial value when you define the field. **Table 9–14** lists the default initial values for each data type.

**Table 9–14: Field initial values**

<table>
<thead>
<tr>
<th>Data type</th>
<th>Initial value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOB</td>
<td>Unknown value (?)</td>
</tr>
<tr>
<td>Character</td>
<td>Null string (displays spaces)</td>
</tr>
<tr>
<td>CLOB</td>
<td>Unknown value (?)</td>
</tr>
<tr>
<td>DATETIME</td>
<td>?</td>
</tr>
<tr>
<td>DATETIME-TZ</td>
<td>?</td>
</tr>
<tr>
<td>Integer</td>
<td>0</td>
</tr>
<tr>
<td>INT64</td>
<td>0</td>
</tr>
<tr>
<td>Decimal</td>
<td>0</td>
</tr>
<tr>
<td>Logical</td>
<td>no (false)</td>
</tr>
<tr>
<td>Date</td>
<td>Unknown value (?) (displays spaces)</td>
</tr>
<tr>
<td>Raw</td>
<td>Null string (displays spaces)</td>
</tr>
<tr>
<td>RECID</td>
<td>Unknown value (?) (displays spaces)</td>
</tr>
</tbody>
</table>

You can use a question mark (?) as a special character to represent an Unknown value. This lets you handle data even when some critical item of information is not yet known. If you put a single question mark in any field, the Data Dictionary treats the item of data as an Unknown value (?).

You can also use TODAY as the initial value for a date field. When you create a new record, the Data Dictionary fills in the current date as the initial value for the field.
- **Component of View** — Displays information only. It tells you whether the field is used in an SQL view. When you use a field as a component of an SQL view, you cannot delete it.

- **Component of Index** — Displays information only. It tells you whether the field is a component of an index.

- **Position** — Displays the order number.

- **Data-Type** — Determines the kind of data values the field can store. Table 9–15 describes the available data types.

### Table 9–15: Data types

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOB ¹</td>
<td>Binary Large Object. A collection of binary data stored as a single unit in a database management system. BLOBs are mainly used to store large objects such as image, sound or video files. BLOBs are a column value in a record. The BLOB column is occupied by a locator that points to a separate assigned area in the database where the BLOB resides.</td>
</tr>
<tr>
<td>Character</td>
<td>Contains data of any kind. Character data can include uppercase and lowercase characters and can consist of alphabetic characters, digits, and the characters $, &amp;, %, –, and _. Although the Data Dictionary allows character field display formats of up to 255 characters, restrict the format length of a character field to the input/output line width of your terminal (typically 80 characters) by specifying the appropriate length in the <strong>Format</strong> field.</td>
</tr>
<tr>
<td>CLOB²</td>
<td>Character Large Object. Similar to a BLOB, the CLOB differs in that it consists solely of character data. There are two types of CLOBs. A <strong>DBCODE PAGE CLOB</strong> has all of its records in the dcodepage and uses the <strong>dbcollation</strong> for comparisons.</td>
</tr>
<tr>
<td>Date</td>
<td>Contains dates from 1/1/32768 B.C. to 12/31/32767 A.D., inclusive. You can specify dates in this century with a two-digit year, such as 8/9/03, or a four-digit year, 8/9/2003. Dates in other centuries require a four-digit year. If you enter dates without the year, the Data Dictionary assumes the current year. If you supply spaces as the value of a date field, the Data Dictionary stores that field as an Unknown value (?) and displays it as spaces.</td>
</tr>
<tr>
<td>DATETIME ¹</td>
<td>Maps to SQL “timestamp” data type. Stores date values as year, month, and day. Stores time values as hours, minutes, seconds and milliseconds. DATETIME value limits for dates are 1/1/32768 B.C. to 12/31/32767 A.D. DATETIME value limits for time are 00:00:00.000 to 23:59:59.999.</td>
</tr>
<tr>
<td>DATETIME-TZ¹</td>
<td>Maps to SQL “timestamp with time zone” data type. Stores date and time values as offset from the Coordinated Universal Time.</td>
</tr>
<tr>
<td>Integer</td>
<td>Contains only whole numbers. They can be positive or negative, from –2,147,483,648 through 2,147,483,647, inclusive. If you enter spaces as the value of an integer field, the Data Dictionary stores the value of that field as 0. The default initial value for the field is 0.</td>
</tr>
</tbody>
</table>
### Table 9–15: Data types

<table>
<thead>
<tr>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT64</td>
<td>Contains only whole numbers. They can be positive or negative, from -9,223,372,036,854,775,808 through 9,223,372,036,854,775,807, inclusive. If you enter spaces as the value of an INT64 field, the OpenEdge RDBMS stores the value of that field as 0. The default initial value for the field is 0.</td>
</tr>
<tr>
<td>Decimal</td>
<td>Contains decimal numbers up to 50 digits. You can use up to 10 digits to the right of the decimal point. If you enter spaces as the value of a Decimal field, the Data Dictionary stores the value of that field as 0. When you choose decimal as the data type, the Data Dictionary enables the Decimal field. You specify the number of decimal places in the Decimal field.</td>
</tr>
<tr>
<td>Logical</td>
<td>Contains logical values such as yes/no or true/false.</td>
</tr>
<tr>
<td>Raw</td>
<td>Contains the value of a field from a non-OpenEdge database that has a data type with no equivalent in the OpenEdge development environment. When you want to work with Raw data, you generally work with the data at the byte level. A Raw value cannot be displayed.</td>
</tr>
<tr>
<td>Recid</td>
<td>Contains the record ID.</td>
</tr>
</tbody>
</table>

1. This data type is currently not available on supported on the ODBC DataServer.
2. This data type is currently not available on supported the MS SQL Server and ODBC DataServers.
- **BLOB** — If you specify a data type of BLOB or CLOB, the Data Dictionary prompts you for more information.

When you assign the BLOB data type to the field you are adding to a table, the **Blob Field Attributes** dialog box appears, as shown in Figure 9–9.

![Blob Field Attributes dialog box](image)

**Figure 9–9:** Blob Field Attributes dialog box

The **Blob Field Attributes** dialog box contains the following fields:

- **Field Name** — Enter the name of the field.

- **Area** — Scroll through the list of available areas of the database to which you can assign the BLOB.

- **Max Size** — Enter the maximum size of the BLOB. Specify any number between 1 and 1073741823 bytes.

**Note:** You can enter the size of the BLOB in bytes, kilobytes, megabytes, or gigabytes. However, if you enter only numbers, the Data Dictionary assumes you’ve entered the BLOB’s size in bytes.

- **Order** — Define the BLOB’s order or accept the default value.

- **Desc** — Provide a description of the field.
• **CLOB**—When you assign the CLOB data type to the field you are adding to a table, the Clob Field Attributes dialog box appears, as shown in Figure 9–10.

![Clob Field Attributes dialog box](image)

**Figure 9–10: Clob Field Attributes dialog box**

The Clob Field Attributes dialog box contains the following fields:

- **Field Name** — Enter the name of the field.
- **Area** — Scroll through the list of available areas of the database to which you can assign the CLOB.
- **Max Size** — Enter the maximum size of the CLOB. Specify any number between 1 and 1073741823 bytes.

**Note:** You can enter the size of the CLOB in bytes, kilobytes, megabytes, or gigabytes. However, if you enter only numbers, the Data Dictionary assumes you have entered the CLOB’s size in bytes.

- **Order** — Define the CLOB’s order or accept the default value.
- **Case Sensitive** — Specify either yes or no.
- **Code Page** — Specify a code page for the CLOB. Make a selection by choosing the default database code page or scroll through the list of available code pages to make another selection.
- **Collation** — Specify database collation.
- **Desc** — Provide a description of the field.

- **Extent** — Defines the extent of an array field. Most fields represent a single value. However, array fields contain multiple elements. For example, the Mnth-shp field of the time table is an array field. It contains 12 elements, one for every month of the year. The extent is the number of elements contained in an array. If you define a field with an extent greater than 0, that field is an array field.
• **Decimals** — Defines the decimal places for a field. When you define a decimal field, you must define the number of digits to the right of the decimal point. For example, **Max-credit** has been defined for two digits to the right of the decimal point to accommodate dollars and cents. If the field is not a decimal, you cannot enter a value.

• **Order** — Specifies the default display order of a field. By default, the Data Dictionary numbers fields in the order you enter them, by increments of 10. This defines where the Data Dictionary lists the field in the field selection list in this table. You can set the order for this field in the table. This display order is not related to the order in which the data are stored in your database. In addition, you can override the Data Dictionary display order of fields in your procedures by naming the fields in the order you want to display them.

The default order numbering is in increments of 10 to let you insert fields in between. The numbers do not have to be contiguous, nor do they have to follow in even-numbered increments. For example, if you decide to add a field called **Category** to the customer table, and you want it to appear by default on your window between **Cust-num** and **Name**, you might assign 15 as its order value. If you want to change the order number increments to 20, you can do so by choosing the **Reorder Fields** option of the **Schema** menu. From the same menu option, you can also change the order values to reflect the alphabetical order of the table names.

• **Mandatory** — Specifies if the field is mandatory. If you define a field as mandatory, it cannot contain the Unknown value (?). However, it can have a space as its value. The default value for mandatory is no. If you accept no for this field, you indicate that the Unknown value (?) is an allowed value for the field.

• **Case-sensitive** — Specifies if the field is case sensitive. The default value is no. Because case-sensitive fields depart from the OpenEdge standard, they are not recommended. However, if you require strict adherence to the ANSI SQL standard or if you are using a DataServer that supports case sensitive fields, you might have to define all character fields as case sensitive. Once you define a field as case sensitive, you can redefine it, unless it is a component of an index. If a field is a component of an index, you cannot change its case sensitivity unless the index is undone.

You can index case-sensitive fields and group them with case-insensitive field components in an index. With case-sensitive indexes, JOHN, John, and john are all unique values. However, they do not sort next to each other. All uppercase letters sort ahead of all lowercase letters. Define case-sensitive variables to hold values moving to and from case-sensitive fields.

• **Valexp** — Specifies the validation criteria for the field. You can enter up to 63 characters on each of the four lines. For messages that exceed 252 characters, you can specify an include file enclosed in brackets (use the syntax: `filename.i`). There is no default value.

**Note:** You should use field validation for backward compatibility for procedure-driven applications. For event-driven programs, use field ASSIGN triggers instead of field validation.

When a user enters a value for a field, you might want to test it to make sure it is a valid entry for the field. The **Valexp** option lets you define a test or validation expression. The expression must be logical; that is, it must be a valid ABL expression that produces a true or false result.
For example, the validation expression for the Cust-num field in the customer table is cust-num > 0. When the user enters a customer number, it is validated against valexp. If the number is greater than 0, valexp is true, and the validation succeeds. If the number is less than 0, the validation fails. The text is then displayed in the Valmsg option.

When you write procedures, you can override any validation expressions you define in the Data Dictionary. If you define validation for an array field in the Data Dictionary, the validation only works if you update the entire array. The validation does not work if you update the array elements individually.

- **Valmsg** — Defines the validation message for a field. You can enter up to 63 characters on each line. You cannot define (and do not need) a validation message unless you specify a validation criteria for a field. If the result of Valexp is false (the validation fails), OpenEdge displays the text specified in Valmsg. For example, the validation expression for the Cust-num field in the customer table is cust-num > 0. When the user enters a 0 for the customer number, OpenEdge displays this message:

```
Customer number must be greater than zero.
```

Because ABL treats the message you define as constant (literal) text, it cannot contain references to the number of the fields or variables. If you want to use fields, variables, or expressions in validation messages, use this VALIDATE option in a Frame phrase.

- **Help** — Defines help information for the field. For certain fields, users might be unsure of the kind of data they need to enter. Therefore, you can specify a help message to indicate what information to enter. OpenEdge displays this message whenever users are prompted for input to the field. For example, the State field in the customer table displays this help message:

```
Enter standard state abbreviation.
```

- **Desc** — Describes the field. You might want to document the purpose of a field just as you might supply a description for a table. OpenEdge does not use this option when running procedures; it is strictly to help you document your application.

### Field Triggers

Choose **Triggers** from the Field Editor to define ASSIGN trigger programs for the field. ASSIGN triggers execute when you assign a value to a database field. When you choose Triggers, a **Field Triggers** dialog box similar to the one shown in Figure 9–11 appears.

![Field Triggers dialog box](image)

**Figure 9–11**: Field Triggers dialog box
The **Field Triggers** dialog box contains the following fields:

- **For ASSIGN** — Specifies the name of the trigger procedure. If you want to create or edit the procedure, press **PUT** and a trigger editor appears.

- **Overrideable** — Specifies whether a session trigger can override the schema trigger. The default value is **no**. For more information about session and schema triggers, see the chapter on database triggers in *OpenEdge Getting Started: ABL Essentials*.

- **Check CRC** — Specifies whether the database engine verifies the cyclical redundancy check (CRC) code. The default value is **yes**. If this option is on, then the AVM stores in the metaschema, along with the trigger procedure name, a unique Cyclic Redundancy Check (CRC) identifier for the compiled version of the trigger procedure. The AVM raises an error if the r-code file it encounters at run time doesn’t match or if there is no compiled version of the procedure.

### View As

Choose **View As** from the **Field Editor** to define the View-As phrase for the type of field widget. When you choose this option, the **View-As Phrase** dialog box appears, as shown in **Figure 9–12**.

![View-As Phrase dialog box](image)

**Figure 9–12:** View-As Phrase dialog box
The list of available widgets depends on the data type of the field. Table 9–16 lists which widgets you can use for each data type. The numeric data type refers to the decimal, integer, INT64, and date data types.

Table 9–16: View As widgets

<table>
<thead>
<tr>
<th>Widget type</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Character</td>
</tr>
<tr>
<td>Combo-box</td>
<td>✓</td>
</tr>
<tr>
<td>Editor</td>
<td>✓</td>
</tr>
<tr>
<td>Fill-in</td>
<td>✓</td>
</tr>
<tr>
<td>Radio-set</td>
<td>✓</td>
</tr>
<tr>
<td>Selection-list</td>
<td>✓</td>
</tr>
<tr>
<td>Text</td>
<td>✓</td>
</tr>
</tbody>
</table>

When you select a widget type, the appropriate View-As phrase syntax for the field appears. Choose the **Copy Syntax** button to copy the syntax to and overwrite any existing text in the editor area of the dialog box. After you edit the View-As phrase, press **GO** to compile the code. If there are errors in your code, the Data Dictionary displays the code and lists the errors. If the code is syntactically correct, the Data Dictionary stores the value and returns you to the **Field Editor**. When you use this field in an application, it appears as the specified widget type.

### GoIndex option

Use the **GoIndex** option to access the **Index Editor**. When you choose this option, a window similar to the one shown in Figure 9–13 appears.

![Figure 9–13: GoIndex option window](image)

For more information about the Index Editor, see the “Index Editor option” section on page 9–30.
Reorder Fields option

Use the **Reorder Fields** menu option to resequence your database’s fields alphabetically by name or numerically by order number. When you choose this option, the Data Dictionary alphabetically lists all the tables defined for your database. Select the table whose fields you want to reorder. The Data Dictionary displays a **Resequence Order of Fields** dialog box similar to the one shown in Figure 9–14.

![Resequence Order of Fields in "Customer"]

Sort by Field Name or Order: Order: [F/A]
Start Numbering From: 10
Increment Each By: 10

Table contains 16 fields.

<OK> <Cancel>

**Figure 9–14:** Resequence Order of Fields dialog box
Global Field Name Change option

Use the **Global Field Name Change** option to change the name of a field in all occurrences in the working database. This option is available only when there is a working database. When you choose this option, the Data Dictionary displays the **Global Field Name Change** dialog box shown in Figure 9–15.

![Global Field Name Change Dialog Box](image)

**Figure 9–15: Global Field Name Change dialog box**

Specify the current field name and the new name. When you specify the old field name, the Data Dictionary checks for a field with the old field name in the working database. If none exists, a message tells you so. If you specify an existing field name, you can enter the new field name.

If you try to change the name of a field to a name that is already in use somewhere in the database, the Data Dictionary displays an error message and does not accept the new field name.

To view a list of all the fields in the database, press F5. You can pick a field from the list that appears. To remove the list without picking a field, press F4.

Once you specify acceptable field names in both fields and choose **OK**, the Data Dictionary prompts you to verify the change.
Index Editor option

Use the Index Editor option to define, rename, or delete indexes for the selected table. An index is a field or combination of fields used to rapidly retrieve a particular record in a table. A single index can be made up of multiple fields. For example, if you know you are always going to access the order-line table by using a combination of the Order-num and Line-num fields, you might want to create an index with those two fields as its components.

**Note:** The current table must contain fields before you can define an index.

When you choose this option, the Data Dictionary alphabetically lists all the tables defined for your database. When you select a table, a window similar to the one shown in Figure 9–16 appears.

![Figure 9–16: Index Editor window](image)

Use the Index Editor to display the existing indexes and their information for the selected table.

The Index Editor window contains the following fields:

- **Name** — Specifies the index name. It does not have to be the same name as the field name. Index names can be up to 32 character long and can consist of alphabetic characters, digits, and the characters $, &, #, %, –, and _. In addition, index names must begin with a letter (A–Z or a–z). You cannot use reserved ABL keywords as index names. Index names are not case sensitive.

- **Unique/Non-Unique** — Defines whether every entry in the index must be different. Enter yes if only one record in a table has a particular index value. For example, the customer table cannot contain two records with the same customer number, so customer number can be a unique index. Conversely, since many customers might be in the same sales region, sales-region should not be a unique index.
- **ACTIVE** — Defines whether the index is initially active, that is, whether it updates itself every time a new record is created, deleted, or modified. If you are defining an index in a very large table, it is more efficient to define the index as initially inactive. (When an index is inactive, you cannot use it to retrieve or order records.) You can subsequently activate the index by running the PROUTIL utility with the IDXBUILD qualifier. This is much faster than building the index from within the Data Dictionary when the database contains a large amount of data. For more information about PROUTIL, see *OpenEdge Data Management: Database Administration*.

- **ROWID** — For non-OpenEdge databases only. See the applicable OpenEdge DataServer guide for more information on the ROWID property.

- **Word** — Defines whether you can search on any word within a field. A word index contains all the words from a text field or array of text fields; thus, you can search for records that contain specific words or phrases. All words in the field are index entries. For more information about word indexes, see the chapter on database access in *OpenEdge Getting Started: ABL Essentials*.

- **Seq** — Defines the order in which fields form the index. For example, cust-order is a compound index, formed by the Cust-num and the Order-num fields. The Cust-num field has a sequence number of 1, and the Order-num field has a sequence number of 2.

- **Field Name** — Displays the field name.

- **Type** — Specifies the field data type. See the “Field Editor option” section on page 9–11 in this chapter for more information about data types.

**Note:** BLOB and CLOB data types cannot participate in indexes.

- **Asc/Dsc** — Specifies how the index sorts records. Value is either **asc** for ascending or **dsc** for descending.

- **Abbr** — Specifies that you can search an index using the first few characters of a field without using the BEGINS phrase, if the field is a character data field. Indexes not comprised of character data require an exact match. If the index has more than one field, this setting applies only to the last field.
Besides the fields previously described, the Index Editor also lists options at the bottom of its display. Table 9–17 describes these options.

Table 9–17: Index Editor options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next</td>
<td>Displays the next index in the table.</td>
</tr>
<tr>
<td>Prev</td>
<td>Displays the previous index in the table.</td>
</tr>
<tr>
<td>First</td>
<td>Displays the first index in the table.</td>
</tr>
<tr>
<td>Last</td>
<td>Displays the last index in the table.</td>
</tr>
<tr>
<td>Rename</td>
<td>Renames the selected index. The name you enter cannot already be used to name an index in the current database.</td>
</tr>
<tr>
<td>Add</td>
<td>Adds a new index to the table.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected index. You cannot delete a primary index without making another index the primary index.</td>
</tr>
<tr>
<td>ChangePrimary</td>
<td>Changes the primary index for the table.</td>
</tr>
<tr>
<td>Uniqueness</td>
<td>(Not supported for OpenEdge databases.)</td>
</tr>
<tr>
<td>MakeInactive</td>
<td>Makes an active index inactive. (When an index is inactive, you cannot use it to retrieve or order records.) To reactivate an index, you must run PROUTIL with the IDXBUILD qualifier.</td>
</tr>
<tr>
<td>Browse</td>
<td>Browses through the entire index component list. This is especially helpful when the index is made up of multiple fields.</td>
</tr>
<tr>
<td>SwitchTable</td>
<td>Shows the list of tables in the database, then switches to another table to edit its indexes.</td>
</tr>
<tr>
<td>GoField</td>
<td>Accesses the Field Editor.</td>
</tr>
<tr>
<td>Undo</td>
<td>Undoes changes you made to the index attributes during this session.</td>
</tr>
<tr>
<td>Exit</td>
<td>Saves your changes, exits the Index Editor, and returns to the Data Dictionary main menu.</td>
</tr>
</tbody>
</table>

Choose the field you want to edit, then choose a menu option. When you choose Add from the Index Editor, the Index Editor prompts you for the name of the new index.
When you enter data in the **Name**, **Unique**, **Active**, and **Word** fields, the **Adding Index** dialog box appears, as shown in Figure 9–17.

The **Index Editor** lists the available fields and their data types. When you choose a field, the **Index Editor** prompts you to specify whether you want the index component to be ascending or descending. After you specify the sort order, the **Index Editor** lets you choose more fields. You can specify up to 16 fields to define an index.

The **Index Editor** lists the selected fields above the divider line on the window. Select all fields, then press **GO**. The **Index Editor** prompts you to specify whether you want to use the **Abbreviate** option for character fields then returns you to the main **Index Editor** window and adds the new index to the list.
Sequence Editor option

Use the Sequence Editor option to view, define, modify, and delete sequences. Sequences are database objects that provide sequential values within any integer range and with your choice of increment.

When you choose this option, a window similar to the one shown in Figure 9–18 appears.

![Sample Sequence Editor window](image)

Figure 9–18: Sample Sequence Editor window

The Sequence Editor displays the existing sequences. When you define or modify a sequence, you provide information in the following attribute fields:

- **Name** — Enter a valid identifier as a name for your sequence. The name can be up to 32 characters long.

- **Initial Value** — Enter an initial integer value for the sequence, or accept the default of 0. The initial value becomes the lower limit if the Increment By value is positive and becomes the upper limit if the Increment By value is negative. You cannot specify an initial value that is greater than your upper limit or less than your lower limit, if defined.

- **Increment By** — Enter a positive or negative integer value by which you want to change the sequence value at each increment, or accept the default of 1. If you enter a positive value, then each increment increases the sequence toward a defined upper limit. If you enter a negative value, then each increment decreases the sequence toward a defined lower limit, and the Upper Limit label changes to Lower Limit.

- **Upper Limit or Lower Limit** — Enter an INT64 (if 64-bit sequences are enabled) or integer value to define the upper or lower limit of the sequence, or accept the default value of unknown (?). If you specify a positive value for the Increment By field, you can only define an upper limit. If you specify a negative value, you can only specify a lower limit. You cannot specify an upper limit that is less than the initial value. Likewise, you cannot specify a lower limit that is greater than the initial value. If you accept the default, the Sequence Editor uses the maximum integer value on your system as the defined upper limit and the minimum integer value as the defined lower limit.
- **Cycle at limit** — Indicates whether the sequence becomes a cycling or terminating sequence. A cycling sequence restarts at the initial value when it reaches the upper limit or decrements beyond its lower limit, providing non-unique values for the life of the sequence. A terminating sequence stops incrementing when it reaches the upper limit or stops decrementing when it reaches its lower limit, providing unique values as long as you do not explicitly reset it. The default is **no**, indicating a terminating sequence.

- **DataServer Name** — For non-OpenEdge databases only. Displays the name of the corresponding non-OpenEdge sequence, if there is one.

- **Owner** — For non-OpenEdge databases only. Displays the name of the user who has permission to access the sequence.

Besides the attribute fields previously listed, the **Sequence Editor** lists options along the bottom of its display. Table 9–18 describes these options.

**Table 9–18: Sequence Editor options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next</td>
<td>Displays the next sequence</td>
</tr>
<tr>
<td>Prev</td>
<td>Displays the previous sequence</td>
</tr>
<tr>
<td>&gt;NextPage</td>
<td>Displays the next set of sequences</td>
</tr>
<tr>
<td>&lt;PrevPage</td>
<td>Displays the previous set of sequences</td>
</tr>
<tr>
<td>First</td>
<td>Displays the first sequence</td>
</tr>
<tr>
<td>Last</td>
<td>Displays the last sequence</td>
</tr>
<tr>
<td>Add</td>
<td>Adds a new sequence</td>
</tr>
<tr>
<td>Modify</td>
<td>Updates sequence attributes</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected sequence</td>
</tr>
<tr>
<td>Undo</td>
<td>Undoes the changes you made to the sequence structures during this session</td>
</tr>
<tr>
<td>Exit</td>
<td>Saves your changes, exits the Sequence Editor, and returns to the Data Dictionary main window</td>
</tr>
</tbody>
</table>
Adjust Field Width option

The Field Width property allows the ABL client to modify the field width of an SQL client. If you do not adjust the setting, the SQL client uses the default width property.

See the “Adjusting the Field Width” section on page 4–10 for more information about this menu option.
This chapter describes the Admin menu and dialog boxes. From the Admin menu you can dump and load database definitions and files, implement security, create bulk loader description files, and enable support for large key entries, as described in the following sections:

- Admin menu
- Dump Data and Definitions option
- Load Data and Definitions option
- Database Identification option
- Security option
- Export Data option
- Import Data option
- Create Bulk Loader Description File option
- Database options
- Enabling large key entries
- Alternate Buffer Pool Maintenance
Admin menu

Table 10–1 describes the Admin menu’s options. The sections following the table contain detailed descriptions of these options.

Table 10–1: Admin menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Task performed . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dump Data and Definitions</td>
<td>Dumps data definitions, table contents, SQL views, user table contents, collation/conversion tables, auto-connect records, sequence definition, and sequence current values</td>
</tr>
<tr>
<td>Load Data and Definitions</td>
<td>Loads data definitions, table contents, SQL views, auto-connect records, user table contents, and sequence values; also reconstructs bad load records</td>
</tr>
<tr>
<td>Database Identification</td>
<td>Maintains and generate DB Identifiers and DB Passkeys for databases</td>
</tr>
<tr>
<td>Security</td>
<td>Specifies database security options</td>
</tr>
<tr>
<td>Export Data</td>
<td>Exports data files in a specific format to the specified directory</td>
</tr>
<tr>
<td>Import Data</td>
<td>Imports data in a specific format to the working database</td>
</tr>
<tr>
<td>Create Bulk Loader Description File...</td>
<td>Creates a bulk loader description file</td>
</tr>
<tr>
<td>Database Options</td>
<td>Maintains auditing and security options</td>
</tr>
<tr>
<td>Enable Large Key Entries</td>
<td>Increases the index key width limit substantially on the database</td>
</tr>
<tr>
<td>Alternate Buffer Pool</td>
<td>Performs Alternate Buffer Pool maintenance, such as specifying ABL objects for the Alternate Buffer Pool, or removing objects from the Alternate Buffer Pool</td>
</tr>
</tbody>
</table>
Dump Data and Definitions option

Table 10–2 describes the menu that appears when you choose **Dump Data and Definitions**. More detailed descriptions of the options follow the table.

**Note:** The Data Dictionary allows you to dump data definition files (.df) greater than 2GB on platforms where ABL supports large files.

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Task performed . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Definitions (.df file)...</strong></td>
<td>Dumps the table definitions</td>
</tr>
<tr>
<td><strong>Table Contents (.d file)...</strong></td>
<td>Dumps table contents</td>
</tr>
<tr>
<td><strong>SQL Views...</strong></td>
<td>Dumps view contents</td>
</tr>
<tr>
<td><strong>User Table Contents...</strong></td>
<td>Dumps user information</td>
</tr>
<tr>
<td><strong>Security Authentication Records</strong></td>
<td>Dumps contents of the security authentication system and domain tables</td>
</tr>
<tr>
<td><strong>Security Permissions</strong></td>
<td>Dumps roles and granted permissions records</td>
</tr>
<tr>
<td><strong>Auto-Connect Records Only...</strong></td>
<td>Dumps auto-connect records</td>
</tr>
<tr>
<td><strong>Collation Tables...</strong></td>
<td>Dumps collation or translation tables</td>
</tr>
<tr>
<td><strong>Sequences Definitions...</strong></td>
<td>Dumps sequence definitions</td>
</tr>
<tr>
<td><strong>Sequences Current Values...</strong></td>
<td>Dumps sequence values</td>
</tr>
<tr>
<td><strong>Database Identification Properties</strong></td>
<td>Dumps database identification information</td>
</tr>
<tr>
<td><strong>Database Options</strong></td>
<td>Dumps database security and auditing settings</td>
</tr>
<tr>
<td><strong>Audit Policies</strong></td>
<td>Dumps audit data as XML or text or to dump application audit events</td>
</tr>
<tr>
<td><strong>Audit Data</strong></td>
<td>Dumps recorded audit data</td>
</tr>
<tr>
<td><strong>Create incremental .df File...</strong></td>
<td>Compares the schema of two OpenEdge databases and creates a .df file that contains any differences</td>
</tr>
</tbody>
</table>
**Data Definitions (.df file) option**

Choose this option to dump table definitions for an individual table or for all the tables in the database. The definition file contains the definitions of tables, which include fields, indexes, and all their characteristics. The Data Dictionary alphabetically lists all the tables defined for your database. If you choose **ALL**, the definition file also contains the sequence and auto-connect record definitions.

**Note:** The .df file can exceed 2GB on platforms where the ABL client supports large files.

After you select the tables you want to dump, the Data Dictionary displays a default name for the file into which it dumps your data definitions (hidden tables are not dumped). This default file is the dump file name you specified when you created the table, or the database name with a .df extension. Dump file names can be up to 32 characters in length.

Specify the filename or use the default value. The Data Dictionary displays each object name as it writes its definition to the definition file. When it has dumped the data definitions, the Data Dictionary displays a status message and prompts you to continue.

**Note:** When a menu option prompts you for a filename only, you can specify a complete pathname.

**Table Contents (.d file) option**

Choose this option to dump table contents. The Data Dictionary alphabetically lists all the tables defined for your database.

**Note:** The .d file can exceed 2GB on platforms where the ABL client supports large files.

When you select only one table, the Data Dictionary displays the **Dump Data Contents** dialog box for that table. If you select multiple tables, the Data Dictionary displays the **Dump Data Contents for Some Tables** dialog box, as shown in Figure 10–1.

![Figure 10–1: Dump Data Contents for Some Tables dialog box](image)
The Dump Data Contents for Some Tables dialog box contains the following fields:

- **Output Directory or Output File** — Specifies the output directory if you choose to dump multiple tables. Specifies the output file if you choose to dump only one table. The default filename for each table is the dump filename you specified when you created the table, or `table-name.d`.

- **Include LOB** — The Data Dictionary also provides the option of dumping LOBs.

- **LOB Directory** — If you select yes in the Include LOB field, you must specify the name of the LOB directory.

- **Character Mapping** — Specifies the type of character mapping:
  - **Default Map** — Specifies the default mapping.
  - **Map** — Specifies the character mapping. Enter the type of mapping in the fill-in field. For example, `MAP laserwriter/french` or `MAP hp2/spanish/italian`.
  - **NO MAP** — Specifies not to perform any character mapping.

- **Code Page** — Specifies the default code page (character set). A code page maps characters to numbers (code-points) and defines the character’s visual representation (glyph). Code pages usually represent a particular language’s alphabet, such as the ibm861 code page (Icelandic).

The Data Dictionary displays each table name as it writes the table contents to the file’s contents file. The Data Dictionary displays a status message and prompts you to continue.

**SQL Views option**

Choose this option to dump SQL view contents. The Data Dictionary prompts you for the filename to which it writes the view. The default filename is `_view.d`. Specify the filename. The Data Dictionary displays a status message and prompts you to continue.

**User Table Contents option**

Choose this option to dump user information. The Data Dictionary prompts you for the filename to which it writes the user table contents. The default filename is `_user.d`. Specify the filename. The Data Dictionary displays a status message and prompts you to continue.

**Security Authentication Records option**

Choose this option to dump the contents of the `_sec-authentication-system` and the `_sec-authentication-domain` tables. These tables contain details about third party authentication systems and domains used to authenticate users who connect to your database through their applications.
When you choose this option, the **Dump Security Authentication Records** dialog box appears, and contains the following fields:

- **Output Directory** — Specifies the directory into which the Data Dictionary dumps `_sec-authentication-system.d` and `_sec-authentication-domain.d`.

- **Character Mapping** — Specifies the type of character mapping:
  - **Default Map** — Leave the **Character Mapping** field blank to use the default mapping.
  - **MAP** — Enter the type of character mapping (for example, MAP `laserwriter/french` or MAP `hp2/spanish/italian`).
  - **NO MAP** — Specifies not to perform any character mapping.

- **Code Page** — Specifies the default code page (character set).

**Note:** The connected user must be a DBA to access the **Dump Security Authentication Records** option.

---

**Security Permissions option**

Choose this option to dump the contents of the `_sec-role`, `_sec-granted-role`, and `_sec-granted-role-condition` tables. These tables store information about user- and OpenEdge-defined security roles including auditing role definitions, audit-privilege to user account mappings, and whether users have permission to grant privileges.

When you choose this option, the **Dump Security Permissions** dialog box appears, and contains the following fields:

- **Output Directory** — Specifies the directory into which the Data Dictionary dumps `_sec-role.d`, `_sec-granted-role.d`, and `_sec-granted-role-condition.d`.

- **Character Mapping** — Specifies the type of character mapping:
  - **Default Map** — Leave the **Character Mapping** field blank to use the default mapping.
  - **MAP** — Enter the type of character mapping (for example, MAP `laserwriter/french` or MAP `hp2/spanish/italian`).
  - **NO MAP** — Specifies not to perform any character mapping.

- **Code Page** — Specifies the default code page (character set).

**Note:** The connected user must be a DBA to access the **Dump Security Permissions** option.
Auto-Connect Records Only option

Choose this option to dump auto-connect records. The Data Dictionary prompts you for the filename to which it writes the auto-connect records. The default filename is _auto.df. Specify the filename and the code page. The Data Dictionary displays a status message and prompts you to continue.

Note: The Data Dictionary does not have a separate option for loading auto-connect records. When you choose Load→Data Definitions from the menu bar, the Data Dictionary automatically loads all the table definitions, regardless of the table type.

Collation Tables option

Choose this option to dump collation tables, case conversion tables, and translation tables. The Data Dictionary prompts you for the filename to which it writes the tables. The default filename is _tran.df. Specify the filename and the code page. The Data Dictionary displays a status message and prompts you to continue.

Note: The Data Dictionary does not have a separate option for loading the collation, case conversion, or translation tables. When you choose Load→Data Definitions from the menu bar, the Data Dictionary automatically loads all the table definitions, regardless of the table type.

Sequences Definitions option

Choose this option to dump sequence definitions. The Data Dictionary prompts you for the name of the file to which it writes the sequence definitions. The default filename is _seqdefs.df. Specify the filename and the code page. The Data Dictionary displays a status message and prompts you to continue.

Note: The Data Dictionary does not have a separate option for loading sequence definitions. When you choose Load→Data Definitions from the menu bar, the Data Dictionary automatically loads all the table definitions, regardless of the table type.

Sequences Current Values option

Choose this option to dump sequence values. The Data Dictionary prompts you for the filename to which it writes the sequence values. The default filename is _seqvals.d. Specify the filename and the code page. The Data Dictionary displays a status message and prompts you to continue.

Note: The Table Contents (.d file) option dumps only the table contents. You must dump the sequence values separately.
Database Identification Properties option

Choose this option to dump the contents of the _db-detail table. The _db-detail table contains information about audit-enabled databases, including the globally unique identifier (GUID), the database description, and the database’s Passkey (if available).

When you choose this option, the Dump Database Identification Properties dialog box appears, and contains the following fields:

- **Output File** — Specifies the name of the dump file. The default is _db-detail.d
- **Character Mapping** — Specifies the type of character mapping:
  - Default Map — Leave the Character Mapping field blank to use the default mapping
  - MAP — Enter the type of character mapping (for example, MAP laserwriter/french or MAP hp2/spanish/italian)
  - NO MAP — Specifies not to perform any character mapping
- **Code Page** — Specifies the default code page (character set)

**Note:** The connected user must be a DBA to access the Dump Database Identification Properties option.

Database Options

Choose this option to dump the contents of the _db-option table. The _db-option table contains user-defined information about the database.

When you choose this option, the Dump Database Options dialog box appears, prompting you to specify the Output File. The default file is _db-option.d.

**Note:** The connected user must be a DBA to access the Dump Database Options option.
Audit Policies option

The Audit Policies submenu allows you to dump audit data as XML or text or to dump application audit events.

Dump as XML

When you choose the Dump as XML (.xml file) option from the Audit Policies submenu, the Select Audit Policies for Dump dialog box appears, containing the names and descriptions of all audit policies.

After you select the policies you want to dump and click OK, the Dump Audit Policies as XML dialog box appears, and contains the following fields:

- **Output File** — The name of the output file. The default is _audit-policies.xml.
- **Code Page** — Indicates that the file will be dumped in the UTF-8 format. You cannot change the code page when dumping audit policies.

If the Data Dictionary encounters an error, it places an error file (_dmpsec.e) in the intended dump directory. If the dump directory does not exist, the Data Dictionary places _dmpsec.e in the session’s current working directory.

**Note:** The connected user must have Audit Administrator permissions in order to access the Dump as XML option. For information about granting Audit Administrator permissions, see the “Edit Audit Permissions option” section on page 10–28.

Dump as Text

When you choose the Dump as Text (.ad file) option from the Audit Policies submenu, the Select Audit Policies for Dump dialog box appears, containing the names and descriptions of all audit policies.

After you select the policies you want to dump and click OK, the Dump Audit Policies as Text dialog box appears, and contains the following fields:

- **Output Directory** — The name of the directory into which the Data Dictionary places the dumped files. Leave this field blank to use the current directory. The dumped files are named for the tables from which they are dumped with a .ad extension.
- **Code Page** — Indicates that the file will be dumped in the UTF-8 format. You cannot change the code page when dumping audit policies.

If the Data Dictionary encounters an error, it places an error file (_dmpsec.e) in the intended dump directory. If the dump directory does not exist, the Data Dictionary places _dmpsec.e in the session’s current working directory.

**Note:** The connected user must have Audit Administrator permissions in order to access the Dump as Text option. For information about granting Audit Administrator permissions, see the “Edit Audit Permissions option” section on page 10–28.
Application Audit Events

Choose the **Application Audit Events** option from the **Audit Policies** submenu to dump the contents of the `_aud-event` table. The `_aud-event` table defines supported audit events for the database.

The **Dump Application Audit Events** dialog box appears and contains the following fields:

- **Output File** — Specifies the name of the output file, with a `.ad` extension. The default is `_aud-event.ad`.

- **Code Page** — Specifies the default code page (character set).

**Note:** The user must have Audit Archiver privileges to access the **Dump Application Audit Events** option. For information about granting Audit Archive permissions, see the “Edit Audit Permissions option” section on page 10–28.

Audit Data option

Choose this option to dump audit data. In an audit-enabled database, audit data is stored in multiple tables. When dumping audit data, the Data Dictionary dumps each table into a separate file named with the table name and a `.ad` extension. To ensure the integrity of the audit data, the Data Dictionary creates a data seal for each dumped file. Each file is part of a set of files that can only be loaded together.

When you select the **Audit Data** option from the **Dump Data and Definitions** menu, the **Dump Audit Data Filter** dialog box appears, and contains the following fields:

- **Select All** — Indicates that all audit data tables will be dumped.

- **Select Some** — Allows you to specify a date range. When you choose **Select Some**, the Data Dictionary activates the **Start Date** and **End Date** fields:
  - **Start Date** — Enter the date of the first audit transaction you want to dump. The default entry is the earliest audit transaction stored in the database.
  - **End Date** — Enter the date of the last audit transaction you want to dump. The default entry is the most recent audit transaction stored in the database.
After you click OK in the Dump Audit Data Filter dialog box, the Dump Audit Data dialog box appears, and contains the following fields:

- **Output Directory** — The name of the directory into which the Data Dictionary places the dumped files. Leave this field blank to use the current directory. The dumped files are named for the tables from which they are dumped with a .ad extension.

- **Character Mapping** — Specifies the type of character mapping:
  - **Default Map** — Leave the Character Mapping field blank to use the default mapping
  - **MAP** — Enter the type of character mapping (for example, MAP laserwriter/french or MAP hp2/spanish/italian)
  - **NO MAP** — Specifies not to perform any character mapping

- **Code Page** — Indicates that the file will be dumped in the UTF-8 format. You cannot change the code page when dumping audit data.

**Note:** The user must have Audit Archiver permissions in order to access the Audit Data option. If a user without Audit Archiver permissions selects the Audit Data option, the Data Dictionary creates an error file named _aud-audit-data.e in the dump directory.

If the Data Dictionary encounters an error while validating the file or directory name or when verifying that the database is connected, it places a file named _dmpsec.e in the intended dump directory. If the intended dump directory does not exist, the Data Dictionary places _dmpsec.e in the session’s current working directory.
Create Incremental .df File option

Use this option to compare two databases’ schemas and create a .df file that contains any differences. You can then use the new .df file to upgrade a database to the current schema.

**Note:** You need at least two databases connected to create an incremental .df file.

When you choose this option, the Create Incremental .df File window appears, as shown in Figure 10–2.

![Create Incremental .df File window](image)

**Figure 10–2: Create Incremental .df File window**

The Data Dictionary lists all connected databases except the working database (which should be the database that has the newest version of the database schema). If you have more than two other databases connected, select the database that has the older version of the database schema. The Data Dictionary then prompts you for the filename to which you want to write the differences. The default filename is delta.df. The Data Dictionary displays the file, field, and index names as it compares the databases.

You can then use this file to apply schema changes to an existing database.

**Note:** If you use this option to create a .df file in conjunction with an object file to update schema changes, you must load the .df file and recompile before you can run the new r-code. You must recompile because the Data Dictionary reorders the indexes during the dump and load procedure.

Batch Incremental utility

This command line utility creates a .df file from comparing two OpenEdge databases. A new file called prodict/dump_inc.p is parallel to prodict/dump_df.p and prodict/load_df.p. Procedure dump_inc.p will query the following environmental variables:

- **DUMP_INC_DFFILE** — Name of file to dump to
- **DUMP_INC_CODEPAGE** — Output code page
- **DUMP_INC_INDEXMODE** — Index-mode for newly created indexes; allowed values are **active** or **inactive**
• **DUMP_INC_RENAMEFILE** — Name of the file that contains rename information. The format of this file is:
  
  - T, <old-table-name>, <new-table-name>
  - F, <table-name>, <old-field-name>, <new-field-name>
  - S, <old-sequence-name>, <new-sequence-name>

  **Note:** There is no need to rename indexes because the code compares index elements and changes them automatically.

• **DUMP_INC_DEBUG**
  
  - 0 = debug off, only errors and important warnings
  - 1 = all the above plus warnings
  - 2 = all the above plus configuration information

The first connected database is the source database and automatically receives the alias DICDB.

Use the following code to call the “dump_inc.p” procedure:

```
OpenEdge-install-dir/bin/_progres -b -db source-db -db target-db -p producit/dump_inc.p
```

The resulting delta.df file can then be applied to the target database, giving it the same definitions as the source database.
Load Data and Definitions option

Choose this option to load table definitions for specified tables or for all the tables in the database.

**Notes:** The Data Dictionary allows you to load data definition files (.df) greater than 2GB on platforms where ABL supports large files.

When a menu option prompts you for a filename only, you can specify a complete pathname.

Table 10–3 describes the menu that appears when you choose this option. More detailed descriptions of the options follow the table.

**Table 10–3: Load Data and Definitions menu**

<table>
<thead>
<tr>
<th>Menu options</th>
<th>Task performed . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Definitions (.df file)...</td>
<td>Loads table, auto-connect record, sequence, and collation table definitions</td>
</tr>
<tr>
<td>Table Contents (.d file)...</td>
<td>Loads table contents</td>
</tr>
<tr>
<td>SQL Views...</td>
<td>Loads SQL view contents</td>
</tr>
<tr>
<td>User Table Contents...</td>
<td>Loads user table contents</td>
</tr>
<tr>
<td>Security Authentication Records</td>
<td>Loads data into the security authentication system and domain tables</td>
</tr>
<tr>
<td>Security Permissions</td>
<td>Loads roles and granted permissions records</td>
</tr>
<tr>
<td>Sequences Current Values...</td>
<td>Loads sequence values</td>
</tr>
<tr>
<td>Reconstruct Bad Load Records...</td>
<td>Uses the error file and the original data file to build a new data file for bad records</td>
</tr>
<tr>
<td>Database Identification Properties</td>
<td>Loads database identification information</td>
</tr>
<tr>
<td>Database Options</td>
<td>Loads database security and auditing settings</td>
</tr>
<tr>
<td>Audit Policies</td>
<td>Loads audit policies as XML, text, or load application audit event data</td>
</tr>
<tr>
<td>Audit Data</td>
<td>Loads recorded audit data</td>
</tr>
</tbody>
</table>
Load Data and Definitions option

Data Definitions (.df file) option

Choose this option to load table and sequence definitions. The .df file contains the definitions of tables, which can include fields and indexes, sequences, auto-connect records, collate/translate tables, and all their characteristics.

Note: The .df file may exceed 2GB on platforms where the ABL client supports large files.

When you choose this option, the Load Data Definitions dialog box shown in Figure 10–3 appears.

![Load Data Definitions dialog box](image)

The Load Data Definitions dialog box contains the following fields:

- **Input File** — The Data Dictionary prompts you for the filename that contains the data definitions you want to load into your current working database. The default file is the working physical database name with a .df extension.

- **Stop If Errors Found** — Specifies if you want to stop the load in the event a bad definition statement is encountered.

- **Output Errors to File** — Specifies if you prefer error messages to be output as files.

- **Add New objects on-line** — Specifies if you want to add database objects while online.

- **Commit Even with Errors** — Specifies whether or not to commit data definitions, even if doing so will create a error message.

- **Output Errors to Screen** — Specifies if you prefer error message outputs to your screen instead of a file.

When you are through selecting your options, choose OK. The Data Dictionary displays each item as it loads the definitions for that object. The Data Dictionary displays a status message and prompts you to continue.

Note: When you choose Load Data Definitions → Data Definitions from the menu bar, the Data Dictionary automatically loads all the table definitions, including any collation or translation tables. If you change the collate/translate tables, you must rebuild your indexes.
Table Contents (.d file) option

Choose this option to load table contents. The Data Dictionary alphabetically lists all the tables defined for your database.

**Note:** The .d file may exceed 2GB on platforms where the ABL client supports large files.

After you select the table or tables, the Data Dictionary opens the **Load Data Contents for Some Tables** dialog box, as shown in Figure 10–4.

![Figure 10–4: Load Data Contents for Some Tables dialog box](image)

The **Load Data Contents for Some Tables** dialog box contains the following fields:

- **Input Directory** — If you load table contents for a single table, the Data Dictionary displays a default name for the file from which you can load your table contents. This default file is always the name of the table dump file, usually the table name with a .d extension.

  When you load the contents of more than one table, the Data Dictionary prompts you for a directory from which it loads the contents files. If you do not specify a directory, the Data Dictionary loads the files from the current directory. The Data Dictionary loads each table from the corresponding `table-name.d` file.

- **Include LOB** — The Data Dictionary provides the option of specifying an LOB directory.

- **LOB Directory** — If you select yes in the **Include LOB** field, you must specify the name of the LOB directory.
• **Acceptable Error Percentage** — The Data Dictionary prompts you for an acceptable error rate. As the Data Dictionary loads records from any files you designate, it might encounter data that cannot be loaded. For example, a record might have data that would cause a duplicate value to be stored for a unique key field. The Data Dictionary does not load the record. If you specify an error rate of 10 percent, the Data Dictionary must successfully load 90 records from every set of 100 records loaded. If the error rate exceeds the specified rate, the Data Dictionary terminates the load. The Data Dictionary places bad records in the `table-name.e` file.

• **Output Errors to Screen** — The Data Dictionary provides you with the option to display error messages to your screen.

---

**Note:** You can move and reorganize data by performing a binary dump and load. The binary method improves the dump and load performance significantly because the data remains in binary format during the process. For more information about performing a binary dump and load, see Chapter 2, “Database Tasks.”

---

**SQL Views option**

Choose this option to load SQL view contents. The Data Dictionary prompts you for the filename from which it reads the view contents. The default filename is `_view.d`. Specify the filename. The Data Dictionary displays a status message and prompts you to continue.

**User Table Contents option**

Choose this option to load user information. The Data Dictionary prompts you for the filename from which it reads the user table contents. The default filename is `_user.d`. Specify the filename. The Data Dictionary displays a status message and prompts you to continue.

**Security Authentication Records option**

Choose this option to load data into the `_sec-authentication-system` and the `_sec-authentication-domain` tables. The `_sec-authentication-system` table defines supported and trusted authentication systems for the database. The `_sec-authentication-domain` table contains valid authentication system domains used to authenticate and validate the authentication of a user account.

When you choose this option, the **Load Security Authentication Records** dialog box appears and prompts you to enter an **Input Directory**.

---

**Note:** The connected user must be a DBA to access the **Load Security Authentication Records** option.
Security Permissions option

Choose this option to load data into the _sec-role, _sec-granted-role, and _sec-granted-role-condition tables. These tables store information about user- and OpenEdge-defined security roles including auditing role definitions, audit-privilege to user account mappings, and whether users have permission to grant privileges.

When you choose this option, the **Load Security Permissions** dialog box appears and prompts you to enter an **Input Directory**.

**Note:** The connected user must be a DBA to access the **Load Security Authentication Records** option.

Sequences Current Values option

Choose this option to load sequence values. The Data Dictionary prompts you for the filename from which it reads the sequence values. The default filename is _seqvals.d. Specify the filename. The Data Dictionary displays a status message and prompts you to continue.

Reconstruct Bad Load Records option

Choose this option to build a new data file for bad records from the error file and original data file. If the Data Dictionary encounters errors while loading a data file, it creates the error file. When you choose this option, the **Reconstruct Bad Load Records** dialog box shown in Figure 10–5 appears.

![Reconstruct Bad Load Records](image)

**Figure 10–5: Reconstruct Bad Load Records dialog box**

The **Reconstruct Bad Load Records** dialog box contains the following fields:

- **Original Data File** — Specifies the name of the file that contains the original data
- **Error File** — Specifies the name of the error file
- **Output Data File** — Specifies the name of the new data file—the default filename is error.d

After you complete the **Reconstruct Bad Load Records** dialog box’s fields and select **OK**, the Data Dictionary builds the new file. You can use a text editor to edit the new data file and fix the bad records after the Data Dictionary builds the new file. Once you have fixed the records, you can reload the file.
Database Identification Properties option

Choose this option to load data into the _db-detail table. The _db-detail table contains information about audit-enabled databases, including the globally unique identifier (GUID), the database description, and the database's Passkey (if available).

When you choose this option, the **Load Database Identification Properties** dialog box appears, and contains the following fields:

- **Input File** — The filename that contains the data you want to load into your current working database. The default file is _db-detail.d.

- **Acceptable Error Percentage** — The Data Dictionary prompts you for an acceptable error rate. As the Data Dictionary loads records from any files you designate, it might encounter data that cannot be loaded. For example, a record might have data that would cause a duplicate value to be stored for a unique key field. The Data Dictionary does not load the record. If you specify an error rate of 10 percent, the Data Dictionary must successfully load 90 records from every set of 100 records loaded. If the error rate exceeds the specified rate, the Data Dictionary terminates the load. The Data Dictionary places bad records in the *table-name*.e file.

- **Display Errors to Screen** — Select this option to display error messages on your screen.

  **Note:** The connected user must be a DBA to access the **Load Database Identification Properties** option.

Database Options option

Choose this option to load data into the _db-option table. The _db-option table contains user-defined information about the database.

When you choose this option, the **Load Database Options** dialog box appears, and contains the following fields:

- **Input File** — The filename that contains the data you want to load into your current working database. The default file is _db-option.d.

- **Acceptable Error Percentage** — The Data Dictionary prompts you for an acceptable error rate. As the Data Dictionary loads records from any files you designate, it might encounter data that cannot be loaded (for example, a duplicate value for a unique key field). The Data Dictionary does not load the record. If you specify an error rate of 10 percent, the Data Dictionary must successfully load 90 records from every set of 100 records loaded. If the error rate exceeds the specified rate, the Data Dictionary terminates the load. The Data Dictionary places bad records in the *table-name*.e file.

- **Display Errors to Screen** — Select this option to display error messages on your screen.

  **Note:** The connected user must be a DBA to access **Load Database Options**.
Audit Policies option

The Audit Policies submenu allows you to load audit data as XML or text or load application audit event data.

Load as XML

When you choose the Load as XML (.xml file) option from the Load Audit Policies submenu, the Load Audit Policies XML dialog box appears, and contains the following fields:

- **Input File** — The name of the file that contains the data you want to load into your current working database. The default file is _audit-policies.xml_.

- **Overwrite Duplicate Policies** — Select if you want the Data Dictionary to overwrite existing policies with the loaded data. If you do not select this option and the Data Dictionary detects duplicate policies, it will display a list of the duplicates and ask you if you want to overwrite them. If the load is running in Batch Mode, the Data Dictionary creates an error file that lists the duplicate policies.

**Note:** The connected user must have Audit Archiver permissions in order to access the Load Audit Policies XML option.

If the Data Dictionary encounters errors when validating the file or directory name or when verifying that the database is connected, it creates an error file called _lodsec.e_ and places it in the intended load directory. If the load directory doesn’t exist, the Data Dictionary places _lodsec.e_ in the session’s current working directory.

If the Data Dictionary encounters errors during the load process, it creates an error file named _audit-policies-xml.e_.

Load as Text option

When you choose the Load as Text (.ad file) option from the Load Audit Policies submenu, the Load Audit Policies Text dialog box appears, prompting you to enter the name of an Input Directory. The Input Directory is the directory where the .ad files are stored.

**Note:** The connected user must have Audit Archiver permissions in order to access the Load Audit Policies Text option. If a user without Audit Archiver permissions selects this option, the Data Dictionary creates the _aud-audit-policy.e_ file in the intended load directory.

If the Data Dictionary encounters errors when validating the file or directory name or when verifying that the database is connected, it creates an error file called _lodsec.e_ and places it in the intended load directory. If the load directory does not exist, the Data Dictionary places _lodsec.e_ in the session’s current working directory.

If the Data Dictionary encounters errors during the load process, it creates an error file in the intended load directory. The Data Directory names the error file for the table it encountered the error on and gives it a .e extension.
Application Audit Events option

Choose the Application Audit Events option from the Load Audit Policies submenu to load data into the _aud-event table. The _aud-event table defines supported audit events for the database.

The Load Application Audit Events dialog box prompts you for an Input File. The default is _aud-event.ad.

**Note:** The user must have Audit Archiver privileges to access the Load Application Audit Events option. For information about granting Audit permissions, see the “Edit Audit Permissions option” section on page 10–28.

If the Data Dictionary encounters errors during the load process, it creates an error file named for the table with a .e extension.

Audit Data option

Choose this option to load audit data. In an audit-enabled database, audit data is stored in multiple tables. When dumping audit data, the Data Dictionary dumps each table into a separate file named with the table name and a .ad extension. To ensure the integrity of the audit data, the Data Dictionary creates a data seal for each dumped file. Each file is part of a set of files that can only be loaded together.

When you select the Audit Data option from the Load Data and Definitions menu, the Load Audit Data dialog box appears, prompting you to enter an Input Directory. During the load, the Data Dictionary searches the specified directory for appropriate .ad files.

**Note:** In order to access the Load Audit Data option, the user must have Audit Archiver permissions.

If the Data Dictionary encounters errors during the load process, it creates an error file named for the table with a .e extension.

If the Data Dictionary encounters errors when validating the file or directory name or when verifying that the database is connected, it creates an error file called _lodsec.e and places it in the intended load directory. If the load directory doesn’t exist, the Data Dictionary places _lodsec.e in the session’s current working directory.
Database Identification option

Choose the **Database Identification** menu option to maintain and generate identification information for audit-enabled databases.

*Table 10–4 describes the options that appear when you choose this menu. More detailed descriptions of the options follow the table. For more information about enabling auditing for databases, see *OpenEdge Data Management: Database Administration*. For more information about auditing, see *OpenEdge Getting Started: Core Business Services*. *

### Table 10–4: Database Identification menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Task performed . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Identification Maintenance</td>
<td>Maintains and updates the DB Identifier, the DB Passkey, the description, and the details for an audit-enabled database</td>
</tr>
<tr>
<td>Database Identification History</td>
<td>Manages historical information about audit archive databases</td>
</tr>
</tbody>
</table>

Database Identification Maintenance option

Choose the **Database Identification Maintenance** option to maintain fields in the `_db-detail` table. The `_db-detail` table contains information about audit-enabled databases, including the globally unique identifier (GUID), the database description, and the Passkey (if available).

**Note:** Only DBAs may access the **Database Identification Maintenance** dialog box.

The **Database Identification Maintenance** dialog box contains the following fields:

- **DB Identifier** — Displays the DB Identifier.
- **DB Passkey** — A password field.
- **DB Description** — A description of the database.
- **Additional Details** — A field for entering additional information about the database.
- **New DB Passkey/Identifier** — Select this option to generate a new DB Identifier or to change the DB Passkey. When you select this option, the Data Dictionary displays the **New Database Passkey/Identifier** dialog box. For information about the **New Database Passkey/Identifier** dialog box, see the “**New Database Passkey/Identifier dialog box**” section on page 10–23.

**Note:** Every time the **New Database Passkey/Identifier** dialog box launches, the Data Dictionary generates a new database identifier.
New Database Passkey/Identifier dialog box

The **New Database Passkey/Identifier** dialog box displays a newly generated DB Identifier and allows you to modify the existing DB Passkey. Access this dialog box from the Data Dictionary’s **Admin** menu (Admin → Database Identification → Database Identification Maintenance). From the Database Identification Maintenance dialog box, select **New DB Passkey/Identifier**. The New DB Passkey/Identifier dialog box appears, and contains the following fields:

- **DB Identifier** — A read-only field displaying the DB Identifier.

- **DB Passkey** — The new DB Passkey. If you want to update the DB Passkey, you must first select **Change Passkey**.

| Note: When you select Change Passkey, the Data Dictionary enables the Blank toggle, which hides the existing value in the DB Passkey field. |

- **Verify Passkey** — Verify the new DB Passkey by re-entering it. Note that the value is hidden.

When you select **OK**, the Data Dictionary dismisses this dialog box and passes the new values back to the Database Identification Maintenance dialog box.

Database Identification History option

Choose the **Database Identification History** option to manage history information about audit-enabled databases by modifying descriptions of audit archive databases (by indicating why a new DB Identifier or DB Passkey was generated, for example). The Database Identification History dialog box appears, and contains the following fields:

- **DB Identifier** — Displays the DB Identifier for the selected database

- **DB Description** — A description of the selected database

- **Additional Details** — A field for entering additional information about the audit archive database (for example, to annotate cycles in the life of the database)

- **Has Passkey** — Indicates if the selected database has a passkey assigned to it

When the Data Dictionary opens this dialog box, it lists the DB Identifier of the current _db-detail record first.

| Note: Only DBAs may view or edit the information in this dialog box. |
Security option

Choose the Security menu option to perform the following tasks:

- Define access permissions for the tables and fields in your database
- Set up a security administrator
- Set up Audit Administrators, Audit Archivers, and Audit Application Event Inserters
- Define and maintain user IDs and passwords
- Perform Authentication system maintenance
- Create and maintain encryption policies

Table 10–5 describes the options that appear when you choose this menu. See OpenEdge Data Management: Database Administration for more information on security administration.

**Table 10–5: Security menu**

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Task performed . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit User List...</td>
<td>Modifies the user list</td>
</tr>
<tr>
<td>Change Your Password...</td>
<td>Changes your password</td>
</tr>
<tr>
<td>Edit Data Security...</td>
<td>Defines access to security tables and fields</td>
</tr>
<tr>
<td>Edit Audit Permissions</td>
<td>Grants, updates, and revokes audit-related permissions</td>
</tr>
<tr>
<td>Security Administrators...</td>
<td>Designates security administrators</td>
</tr>
<tr>
<td>Disallow Blank Userid Access...</td>
<td>Denies blank user IDs access to the working database</td>
</tr>
<tr>
<td>User Report...</td>
<td>Displays or prints user information</td>
</tr>
<tr>
<td>Authentication System Maintenance</td>
<td>Creates and maintains authentication systems and authentication system domains</td>
</tr>
<tr>
<td>Encryption Policies</td>
<td>Creates and maintains encryption policies for an encryption-enabled database</td>
</tr>
</tbody>
</table>
Edit User List option

Choose Security → Edit User List to add or delete users from the user list. You can also change the user name. When you choose this option, the Edit User List dialog box shown in Figure 10–6 appears.

![Edit User List dialog box](image)

**Figure 10–6:  Edit User List dialog box**

The Edit User List dialog box contains the following fields:

- **User ID** — Lists the users defined for the working database
- **User Name** — Displays the user name of the selected user
- **Password** — Indicates whether the selected user has been assigned a password

Table 10–6 describes the options at the bottom of the Edit User List dialog box.

**Table 10–6:  Edit User List dialog box options** *(1 of 2)*

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next</td>
<td>Highlights the next user in the user list</td>
</tr>
<tr>
<td>Prev</td>
<td>Highlights the previous user in the user list</td>
</tr>
<tr>
<td>First</td>
<td>Highlights the first user in the user list</td>
</tr>
<tr>
<td>Last</td>
<td>Highlights the last user in the user list</td>
</tr>
<tr>
<td>Add</td>
<td>Adds a new user</td>
</tr>
<tr>
<td>Modify</td>
<td>Modifies the selected user’s name</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected user ID</td>
</tr>
<tr>
<td>CallAdmin</td>
<td>Accesses the Security Administrators dialog box</td>
</tr>
</tbody>
</table>
Once you designate security administrators, only they can edit the user list.

### Change your password

Choose this option to change your password. The Data Dictionary prompts you for your new password. Enter your new password, then choose **OK**. Remember that passwords are case sensitive. The Data Dictionary prompts you to verify the new password.

The OpenEdge RDBMS allows all users to change their own passwords.

### Edit Data Security

Choose **Security → Edit Data Security** to define compile-time security for an application database. The Data Dictionary alphabetically lists the names of the tables defined for the working database. When you select a table, the Data Dictionary displays the current permissions lists for the table. **Figure 10–7** shows the default permissions for the customer table.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Accesses the <strong>Data Security</strong> option dialog box</td>
</tr>
<tr>
<td>Report</td>
<td>Generates a user report for the working database</td>
</tr>
<tr>
<td>Undo</td>
<td>Discards any changes you made and exits to the Data Dictionary window</td>
</tr>
<tr>
<td>Exit</td>
<td>Exits, saves changes, and returns to the Data Dictionary window</td>
</tr>
</tbody>
</table>

### Table 10–6: Edit User List dialog box options

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Once you designate security administrators, only they can edit the user list.
The upper section of the **Edit Data Security** dialog box displays the six levels of security permissions for the table. By default, the Data Dictionary initializes all levels with an asterisk (*), which means that all users are allowed to modify table permissions. The lower section describes the fields for the permissions lists.

The six levels of security permissions for tables are:

- **Can-Read** — Specifies the users who have permission to read a table. The default is an asterisk (all users).

- **Can-Write** — Specifies the users who can write to a table or update records. The default is an asterisk (all users).

- **Can-Create** — Specifies the users who can create new records. In order to create a new record, a user must have Can-Write privileges. The default is an asterisk (all users).

- **Can-Delete** — Specifies the users who can delete records from a table. The default is an asterisk (all users).

- **Can-Dump** — Specifies if you can dump database or table definitions and data with triggers disabled. The default is an asterisk (all users).

- **Can-Load** — Specifies if you can load database or table definitions and data with triggers disabled. The default is an asterisk (all users).

Table 10–7 describes the values you use to define the permissions for a table.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Indicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>All users are allowed access.</td>
</tr>
<tr>
<td>user</td>
<td>Only this user has access.</td>
</tr>
</tbody>
</table>
| ! user, *  | All users have access except this user.  
**Note:** Do not place the asterisk (*) before !user. If you do, all users are allowed access. |
| acct*      | User IDs that begin with acct have access. |

Table 10–8 describes the options at the bottom of the **Edit Data Security** dialog box.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>NextField</td>
<td>Displays the next field defined for the specified table</td>
</tr>
<tr>
<td>PrevField</td>
<td>Displays the previous field defined for the specified table</td>
</tr>
<tr>
<td>ForwardTable</td>
<td>Displays the permissions for the next table defined for the working database</td>
</tr>
<tr>
<td>BackwardTable</td>
<td>Displays the permissions for the previous table defined for the working database</td>
</tr>
</tbody>
</table>
Once you designate security administrators, only they can use this option.

**Edit Audit Permissions option**

Choose this option to control which users can access audit data.

Until a user has been granted the Audit Administrator permission with grant rights, only the DBA may grant audit-related permissions. Once an Audit Administrator is established, he grants specific rights to other users and determines whether to give these users grant rights. The DBA maintains the right to revoke and/or delete the Audit Administrator’s permissions until the Audit Administrator has granted Audit Administrator permissions to at least one other user.

**Note:** For a complete explanation of audit security, see *OpenEdge Getting Started: Core Business Services*.

To edit audit permissions for a user, from the **Edit Audit Permissions** dialog box select a user from the browser and edit the fields described below:

- **UserId** — The ID of the user to whom you grant permissions.
- **Permission** — Description of the permission being granted. This menu is populated with the permissions you are permitted to set. If you are an Audit Administrator with grant rights for Audit Administrator, all available audit-related permissions appear in this drop-down menu. There are four possible audit security permissions:
  - **Audit Administrator** — An authenticated user who has privileges to create, update, and delete audit policies and manage audit data
  - **Application Audit Event Inserter** — An authenticated user who has privileges to generate application audit events
  - **Audit Data Archiver** — An authenticated user who has privileges to maintain (truncate, dump, or load) audit data
  - **Audit Data Reporter** — An authenticated user who has privileges to read audit data

---

**Table 10–8: ** Edit Data Security dialog box options

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify</td>
<td>Modifies the permissions for a table or field</td>
</tr>
<tr>
<td>SwitchTable</td>
<td>Displays the list of tables again so you can select a new table</td>
</tr>
<tr>
<td>JumpField</td>
<td>Displays a list of all the fields defined for the current table</td>
</tr>
<tr>
<td>CallAdmin</td>
<td>Accesses the <strong>Security Administrators</strong> dialog box</td>
</tr>
<tr>
<td>UserEditor</td>
<td>Accesses the <strong>Edit User List</strong> dialog box</td>
</tr>
<tr>
<td>Report</td>
<td>Generates a user report for the working database</td>
</tr>
<tr>
<td>Exit</td>
<td>Exits, saves changes, and returns to the Data Dictionary window</td>
</tr>
</tbody>
</table>

Once you designate security administrators, only they can use this option.
• **Grantor** — The current user’s userid. The Data Dictionary automatically populates this field.

• **Comments** — Additional information about the user or granted permission.

• **Can Grant Permissions for** — If selected, indicates that the user can grant permissions to other users for the specified permission.

The following control options appear on the **Edit Audit Permissions** dialog box:

• **Done** — Closes the dialog box. This option is disabled until you select **Save** or **Cancel**.

• **Grant** — Grants the permission displaying the **Permission** field to the selected userid.

• **Save** — Saves changes to the current record and resets the fields for input.

• **Cancel** — Cancels current changes to the record. Note that this option is disabled until you edit one of the fields for an existing record. If you click **Cancel** while creating a new record, the Data Dictionary displays the previously selected record.

• **Revoke** — Revokes the permission displayed in the **Permissions** field from the selected userid, after prompting for verification. If you revoke a permission from a userid, the next userid in the browser appears. If you revoke all permissions from all existing userids, the Data Dictionary clears the **Edit Audit Permission** dialog box’s fields.

### Security Administrators option

Choose **Security** → **Security Administrators** to designate a security administrator. The Security Administrators dialog box shown in Figure 10–8 appears.

#### Security Administrators

Enter login ids to control who can change security.

Examples:
- `*` — All users (login ids) are allowed access.
- `<user>,<user>,...` — Only these users have access.
- `<user>,<user>,<user>,<user>,<user>` — All except these users have access.
- `<user>` — Only users that begin with ‘`<user>`’ allowed.

Do not use spaces in the string (they will be taken literally).

<OK> <Cancel>

**Figure 10–8:** Security Administrators dialog box

You can enter many user IDs here, but you must include your own user ID. Separate the user IDs with commas, but no spaces. When you are done entering the user IDs, choose **OK**. The Data Dictionary prompts you to verify the entries you made.

Once you designate security administrators, only they can use this option.

### Disallow Blank User Access option

Choose this option to prevent users who are not listed in the user list from accessing the working database. When you choose this option, the Data Dictionary displays an alert box prompting you to verify that you want to prevent blank user ID access to the database.
User Report option

Choose **Security → User Report** to display or print a list of user IDs and user names, as well as to determine which user IDs are protected by passwords. When you choose this option, the Data Dictionary displays the user ID list in the **User Report** dialog box, as shown in **Figure 10–9**.

<table>
<thead>
<tr>
<th>User ID</th>
<th>User Name</th>
<th>Has Password?</th>
</tr>
</thead>
<tbody>
<tr>
<td>aspauldi</td>
<td>Annabelle Spauldi</td>
<td>no</td>
</tr>
<tr>
<td>Ernest</td>
<td>Ernest P</td>
<td>yes</td>
</tr>
<tr>
<td>Ian</td>
<td>Ian Logan</td>
<td>no</td>
</tr>
<tr>
<td>Mary</td>
<td>Mary Persons</td>
<td>no</td>
</tr>
<tr>
<td>Math</td>
<td>Matt Harvey</td>
<td>no</td>
</tr>
</tbody>
</table>

**Figure 10–9: User Report dialog box**

To send the report to a file or printer, choose **Print**. The Data Dictionary prompts you for the printer or filename, whether you want to append the report to an existing file, and the page length.

Authentication System Maintenance option

The options of the **Authentication System Maintenance** submenu allow you to specify details about third party authentication systems and modules used to authenticate users who connect to your database through their applications. For more information about authentication and application security, see *OpenEdge Getting Started: Installation and Configuration*.

The **Authentication System Maintenance** submenu provides two options, as described in the following sections:

- **Security Authentication Systems**
- **Authentication System Domains**

**Security Authentication Systems**

Use the **Authentication Systems** dialog box to define authentication systems for use with your ABL applications.

*Note:* The connected user must be a DBA to access this dialog box.

When you select the **Security Authentication Systems** option from the **Authentication System Maintenance** submenu, the **Authentication Systems** dialog box appears. The browser at the top of the dialog box contains the following fields:

- **Domain Type** — Displays the type of authentication system (for example, LDAP, KERBOS, RSA, KEON, Internal)
- **Description** — Description of the authentication system
To create a new authentication system record, select **Create** and enter information into the following fields:

- **Domain Type** — The type of authentication system (LDAP, KERBOS, RSA, KEON, Internal, for example)

  **Note:** This field is enabled only when you are creating a new system. You cannot edit an existing domain type. To change the domain type of an existing system, delete the current system record and add a new one.

- **Description** — Description of the authentication system

- **Comments** — Specific authentication system details

To update an authentication system record, select it in the browser and edit the **Description** and **Comment** fields.

The following control options appear on the **Authentication Systems** dialog box:

- **Done** — Closes the dialog. This option is disabled until you select **Save** or **Cancel**.

- **Create** — Creates a new authentication system record and resets the fields for input.

- **Save** — Saves changes to the current authentication system record and resets the fields for input.

- **Cancel** — Cancels current changes to the authentication system record. Note that this option is disabled until you select **Create** or edit one of the fields for an existing record. If you select **Cancel** while creating a new record, the previously selected record displays.

- **Delete** — Deletes the current administration system record after prompting for verification. If you delete a record, the next record in the browser appears. If you delete the last record, the Data Dictionary clears all of the fields in the **Authentication Systems** dialog box.

Table 10–9 describes the hot keys used with the Authentication Systems dialog box.

**Table 10–9:** Authentication Systems dialog box hot keys

<table>
<thead>
<tr>
<th>Hot key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTRL+A</td>
<td>Selects <strong>Create</strong>, if the option is enabled</td>
</tr>
<tr>
<td>CTRL+N</td>
<td>Selects <strong>Cancel</strong>, if the option is enabled</td>
</tr>
<tr>
<td>DEL</td>
<td>Selects <strong>Delete</strong>, if the option is enabled</td>
</tr>
<tr>
<td>F1/GO</td>
<td>Selects <strong>Save</strong>, if the option is enabled; otherwise, selects <strong>Done</strong></td>
</tr>
</tbody>
</table>
**Authentication System Domains**

Use the **Authentication System Domains** dialog box to define authentication system domains for use with your ABL applications.

**Note:** The connected user must be a DBA to access this dialog box.

When you select the **Authentication System Domains** option from the **Authentication System Maintenance** submenu, the **Authentication System Domains** dialog box appears. The browser at the top of the dialog box contains the following fields:

- **Name** — Displays the logical name of a single authentication system that can be used to authenticate or validate the authentication of a user account
- **Type** — Displays the type of authentication system (LDAP, KERBOS, RSA, KEON, Internal, for example)
- **Description** — Description of the authentication domain

To update an authentication system domain, select it in the browser and edit the fields described below. To create a new authentication system domain, select **Create** and enter information into the following fields:

- **Type** — The type of authentication system (for example: LDAP, KERBOS, RSA, KEON, or Internal)

**Note:** This field is enabled only when you are creating a new domain. You cannot edit an existing domain type. To change the existing domain, delete the current domain record and add a new one.

- **Name** — The logical name of a single authentication system that can be used to authenticate or validate the authentication of a user account

**Note:** This field is enabled only when you are creating a new domain. You cannot edit an existing domain name. To change the name of an existing domain, delete the current domain record and add a new one.

- **Access Code** — Encrypted access code phrase used to validate the identity and integrity of any CLIENT-PRINCIPAL issued by the authentication system represented by this domain
- **Audit Context** — The audit record context value when a _client-session record is recorded to the database
- **Runtime Options** — A comma delimited list of runtime options for use with the domain’s ABL Virtual Machine (AVM)
- **Description** — Description of the authentication domain
- **Comments** — Specific authentication domain details
- **Domain Enabled** — Activate to specify the current domain is enabled
The following control options appear on the **Authentication System** dialog box:

- **Done** — Closes the dialog. This option is disabled until you select **Save** or **Cancel**.

- **Create** — Creates a new authentication system domain record and resets the fields for input.

- **Save** — Saves changes to the current authentication system domain record and resets the fields for input.

- **Cancel** — Cancels current changes to the authentication system domain record. Note that this option is disabled until you select **Create** or edit one of the fields for an existing record. If you select **Cancel** while creating a new record, the previously selected record displays.

- **Delete** — Deletes the current administration system domain record after prompting for verification. If you delete a record, the next record in the browser appears. If you delete the last record, the Data Dictionary clears all of the fields in the **Authentication System Domains** dialog box.

Table 10–10 describes the hot keys used with the Authentication Systems Domains dialog box.

<table>
<thead>
<tr>
<th>Hot key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTRL+A</td>
<td>Selects <strong>Create</strong>, if the option is enabled</td>
</tr>
<tr>
<td>CTRL+N</td>
<td>Selects <strong>Cancel</strong>, if the option is enabled</td>
</tr>
<tr>
<td>DEL</td>
<td>Selects <strong>Delete</strong>, if the option is enabled</td>
</tr>
<tr>
<td>F1/GO</td>
<td>Selects <strong>Save</strong>, if the option is enabled. Otherwise, selects <strong>Done</strong></td>
</tr>
</tbody>
</table>

**Note:** This hot key applies only if an authentication system record is selected in the browser.
Encryption Policies

The options of the Encryption Policies submenu allow you to create, update, and view encryption policies for ABL objects in Type II areas.

**Note:** Encryption policy maintenance for Type I areas must be performed with PROUTIL EPOLICY. Encryption policy maintenance for SQL objects must be performed with OpenEdge SQL syntax, such as CREATE TABLE and ALTER TABLE. For more information on PROUTIL EPOLICY, see OpenEdge Data Management: Database Administration. For more information on OpenEdge SQL syntax, see OpenEdge Data Management: SQL Reference.

The Encryption Policies submenu provides three options, as described in the following sections:

- Edit Encryption Policy
- Generate Encryption Keys
- Encryption Policy History

For details on encryption and encryption policies, see OpenEdge Getting Started: Core Business Services.

**Edit Encryption Policy**

Create and update encryption policies with the Edit Encryption Policy dialog box.

The Edit Encryption Policy dialog box allows you to:

- Enable or disable encryption for an object
- Generate a new encryption key for an encrypted object
- Copy encryption policies to multiple objects
To create an encryption policy for a database object:

1. From the Data Dictionary, choose Admin→Security→Encryption Policies→Edit Encryption Policy. The Object Selector dialog box appears:

The initial view displays the tables and indexes of the connected database with encryption both enabled and disabled. You can change the list to view different sets of objects by checking and unchecking the Show boxes. For example, to also see LOBs, check Show LOBs; to only see objects with an existing encryption policy, for Show Encryption check Enabled, and uncheck Disabled.

2. Scroll through the list, selecting objects, or click Select Some to bring up the Select Objects by Pattern Match dialog box to refine the list. Enter the object name or pattern in the Object Name fill-in and click OK. When you return to the Object Selector dialog box, any objects matching the pattern are added to the selected objects.

3. If you selected too many objects, clicking Deselect Some brings up the Deselect Objects by Pattern Match dialog box, allowing you to enter an object name or pattern to deselect. Click OK to return to the Object Selector dialog box with any objects matching the specified pattern deselected.

4. Once you have selected all the objects, click OK to proceed. The Edit Encryption Policy dialog box appears:

Note: Passphrase and Verify Passphrase are always disabled.
5. Select an object and check **Encryption enabled**. The default cipher, AES_CBC_128, is selected. Change the cipher if desired. Click **Save** to save the change, or **Reset** to undo the change.

**Note:** The change is not committed to the database until you click **Commit**.

6. When you have save a change, the **Copy** button is activated. Click **Copy** to propagate your change to multiple objects at once. Clicking **Copy** brings up the **Copy Encryption Policy Setting To** dialog box, as shown:

You can select a single object or multiple objects. When you have selected all you want to copy the setting to, click **OK** to return to the **Edit Encryption Policy** dialog box.

7. Continue selecting and modifying object settings until you have specified all the objects you want to change. An asterisk (*) in front of the object name indicates you made changes to it. Click **Review** at any time to view the status of your changes.

8. When your changes are complete, click **Commit** to commit the changes to the database, or **Revert**, to cancel all your changes. If you choose to commit your changes, you are asked to confirm; click **Yes** to commit your changes.
Generate Encryption Keys

Generate new encryption keys for existing encryption policies with the Generate Encryption Key dialog box. Periodically generating a new encryption key (rekeying) for an existing policy is part of the recommended encryption policy maintenance best practices.

To generate a new encryption key for an existing encryption policy:


   By default, only tables and indexes with existing encryption policies are shown. You can also show LOBs with encryption policies by checking Show LOBs. You can refine the list by filtering by cipher if you check Filter Cipher and choose a cipher.

2. Select the object or objects you want to generate new encryption keys for and click OK. The Generate Encryption Keys dialog box appears:

   ![Generate Encryption Keys dialog box]

   **Note:** The Save, Copy, and Revert buttons are disabled, as well as the Passphrase and Verify Passphrase fill-ins.

3. Click Commit to generate new keys for all the objects listed in the Generate Encryption Keys dialog box, or click Cancel to not generate any keys.

4. If you click Commit, you are asked to confirm your decision.
Encryption Policy History

You can view the history of encryption policies. Choose Admin→Security→Encryption Policies→Encryption Policy History, and the Encryption Policy History dialog box appears:

For the selected object, the encryption policy history displays the following information for the selected object:

- **Version** — The version number of the encryption policy. Numbering starts at 0 and increases with every policy change.
- **State** — The state of the policy, either Current or Previous.
- **Cipher** — The encryption policy cipher. For objects with encryption disabled, the cipher column lists Disabled.
Export Data option

Choose this option to export data in different formats. Table 10–11 describes the menu that appears when you choose this option.

Table 10–11: Export Data menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIF ...</td>
<td>Exports data records in DIF format</td>
</tr>
<tr>
<td>SYLK ...</td>
<td>Exports data records in SYLK format</td>
</tr>
<tr>
<td>Text ...</td>
<td>Exports data records in plain text format</td>
</tr>
<tr>
<td>Microsoft Word Merge Data...</td>
<td>Exports data records to Microsoft Word in text format</td>
</tr>
</tbody>
</table>

Choose these options to export data in the specified format. When you choose the desired format, the Data Dictionary alphabetically lists all the tables defined for your database. When you select a table, a dialog box similar to the one shown in Figure 10–10 appears.

Figure 10–10: Export Text dialog box
The **Export Text** dialog box contains the following user-interface elements:

- **Output File** — Specifies the name of the destination file. Table 10–12 lists the export format file extensions.

### Table 10–12: Export format file extensions

<table>
<thead>
<tr>
<th>File format</th>
<th>Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIF</td>
<td>.dif</td>
</tr>
<tr>
<td>SYLK</td>
<td>.sl</td>
</tr>
<tr>
<td>Text</td>
<td>.txt</td>
</tr>
<tr>
<td>WordStar</td>
<td>.txt</td>
</tr>
<tr>
<td>Microsoft Word</td>
<td>.doc</td>
</tr>
<tr>
<td>WordPerfect</td>
<td>.txt</td>
</tr>
</tbody>
</table>

If your output is to text format, the **Output Record Formatting** dialog box shown in Figure 10–11 appears.

![Output Record Formatting](image)

**Figure 10–11: Output Record Formatting dialog box**

The **Output Record Formatting** dialog box contains the following user-interface elements:

- **Record Start String** — Specifies the character that indicates the beginning of a record. Enter the appropriate three-digit code.

- **Record End String** — Specifies the character that indicates the end of a record. The default is a carriage return (~012).
– **Field Delimiter** — Specifies the character that surrounds a field. The default is a quotation mark (").

– **Field Separator** — Specifies the character that separates fields. The default is a comma (,).

**Note:** Do not use an escape character, tilde (~), or backslash (\) as a field delimiter or field separator.

For all formats, the Data Dictionary informs you of the number of records it dumped in specified format and prompts you to continue.

- **WHERE-Clause** — Specifies the record selection criteria (for example, \texttt{WHERE cost > 100}). A record becomes a row in a worksheet, and the worksheet starting point is always row 1, column 1. The default is to select all the records.

- **BY-Clause** — Specifies that you want to sort the records using a BY clause (for example, \texttt{BY name}). The default is to use the primary index.

- **Export Fields** — Specifies the fields you want to export. The default value is \texttt{Some}. Each field corresponds to a worksheet column.

  If you specify \texttt{Selected Fields}, the Data Dictionary lists all the fields in the specified file. To select a field, press \texttt{SPACEBAR} to highlight the field, than press \texttt{RETURN}. Select the fields in the order you want to export them into your destination file. To skip a column in the worksheet, do not press \texttt{RETURN} while the field is highlighted. Press \texttt{GO} to export the fields.

- **Disable Triggers During Export** — Specifies whether to disable triggers during the export.
Import Data option

Choose this option to import data with different file formats to the working database. Table 10–13 describes the menu that appears when you choose this option.

Table 10–13: Import Data menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIF ...</td>
<td>Imports files with DIF format</td>
</tr>
<tr>
<td>SYLK ...</td>
<td>Imports files with SYLK format</td>
</tr>
<tr>
<td>Delimited Text ...</td>
<td>Imports files with text format</td>
</tr>
<tr>
<td>Fixed-Length ...</td>
<td>Imports files in fixed-length file format</td>
</tr>
<tr>
<td>dBASE Definitions ...</td>
<td>Imports dBASE II, III, III+, and IV file structures</td>
</tr>
<tr>
<td>dBASE File Contents ...</td>
<td>Imports files with dBASE format</td>
</tr>
</tbody>
</table>

Choose Import Data. Then choose the DIF, SYLK, Delimited Text, or dBASE File Contents option to import data in the specified format. When you choose the desired format, the Data Dictionary alphabetically lists all the tables defined for your database. When you select a table, a dialog box similar to the one shown in Figure 10–12 appears.

Figure 10–12: Import DIF dialog box
The Import DIF dialog box contains the following fields:

- **Import File** — Specifies the load file.

- **Import** — Specifies the fields you want to import. The default value is **Some**. Each field corresponds to a worksheet column. Each row in the worksheet becomes a record in the table. For each row, the Data Dictionary loads worksheet values column by column into the specified record fields.

  If you specify **Selected Fields**, the Data Dictionary lists all the fields in the specified file. To select a field, press **SPACEBAR** to highlight the field, than press **RETURN**. Select the fields in the order you want to import them into your file. To skip a column in the worksheet, do not press **RETURN** while the field is highlighted. Press **GO** to import the fields.

- **Disable Triggers During Import** — Specifies whether to disable triggers during the import.

  **Notes:** To import a file in delimited text format, you must also specify the field separation character.

  To import dBASE file contents, you must also specify if the file originated on an IBM PC or compatible.

### Fixed-Length option

Choose Import Data→**Fixed-Length** to import data in fixed-length format. When you choose this option, the Data Dictionary alphabetically lists all the tables defined for your database. When you select a table, the **Import Fixed-Length** dialog box shown in Figure 10–13 appears.

![Import Fixed-Length dialog box](image)

**Figure 10–13:** Import Fixed-Length dialog box

The Import Fixed-Length dialog box contains the following user-interface elements:

- **Import File** — Specifies the load file

- **Disable Triggers During Import** — Specifies whether to disable triggers during the import
After you choose OK from the Import Fixed-Length dialog box, the Fixed-Length Field Columns dialog box shown in Figure 10–14 appears.

![Fixed-Length Field Columns](image)

**Figure 10–14: Fixed-Length Field Columns dialog box**

The Fixed-Length Field Columns dialog box lists the fields you can import.

### dBASE Definitions option

Choose Import Data → dBASE Definitions to import dBASE II, III+, and IV file structures. When you choose this option, the Import dBASE II, III, III+ and IV Definitions dialog box shown in Figure 10–15 appears.

![Import dBASE II, III, III+ and IV Definitions](image)

**Figure 10–15: Import dBASE II, III, III+ and IV Definitions dialog box**

Specify the DBF filename and up to 10 index (NDX) names. If you want to import more than 10 indexes, you must rerun the option and specify the remaining indexes.
Create Bulk Loader Description File option

You use the bulk loader description file with the Bulk Loader utility to load data files into an OpenEdge database more quickly than you can using the Data Dictionary Load utility. See the “Creating a bulk loader description file” section on page 3–17 for more information about this menu option.
Database options

Choose this option to maintain auditing and security options for your database. When you choose this option, the Database Options dialog box appears, divided into two sections: Auditing Options and Security Options.

The Auditing Options section contains the following fields:

- **Use Application User Id for Auditing** — Overrides the database connection’s user ID for recording audit user ID and uses the application level user ID (if set) for auditing

- **Enforce Audit Insert Privilege** — Requires that users of applications that use Application Auditing, and that set Auditing Context, have the Application Audit Event Inserter privilege granted to them

**Note:** If auditing is not enabled for the database, the Data Dictionary disables the Auditing Option fields.

The Security Options section contains the following fields:

- **Trust Application Domain Registry** — Overrides the database connection’s domain registry settings and uses the application level domain registry settings

- **Disallow Blank UserId** — Blocks database connections with blank user IDs

- **Record Authenticated Client Sessions** — Record user login sessions

- **Use Runtime Permissions Checking** — Applies CAN-WRITE and CAN-READ permissions at runtime, rather than at compile time
Enabling large key entries

For databases of 4K and 8K block size, large key entries are enabled by default on newly created databases, providing a substantial increase in the index key width limits. Large key entries are not available for databases that are read-only, non-OpenEdge, either 1K or 2K in block size, or served by a database server from an earlier OpenEdge release.

You must be a database administrator to enable large key entries on a connected database. Large key entry support cannot be disabled. From the Data Dictionary, choose Admin→Enable Large Key Entries. For OpenEdge Release 10.1C and higher databases, a message appears informing you that large key entries are already enabled. For pre-OpenEdge Release 10.1C databases, a message asking you to confirm that you do want to enable large key entries appears.
Alternate Buffer Pool Maintenance

The Alternate Buffer Pool is a collection of buffers in shared memory that are logically separate from the primary buffer pool. Management of the buffers in the Alternate Buffer Pool is independent of the primary buffer pool. Assigning specific database areas or objects to occupy buffers from the Alternate Buffer Pool might improve your buffer hit rate, thereby reducing the need to read and write buffers to and from disk, and possibly improving performance.

Specifying the best objects for the Alternate Buffer Pool is application-specific. Tables considered “hot” (very active) are good candidates, as are their related indexes. Tables and indexes that are governed by an encryption policy are also considered good candidates because the cost of encrypting and decrypting blocks as they are written and read from disk can be high.

Choose Admin → Alternate Buffer Pool → Alternate Buffer Pool Maintenance to identify objects for the Alternate Buffer Pool. The objects must reside in a Type II area to be individually designated for the Alternate Buffer Pool. Object-level assignments can be made either while the database is online, or while the database is in single-user mode.

**Notes:** Alternate Buffer Pool assignments for SQL objects are made with OpenEdge SQL syntax. For more information, see *OpenEdge Data Management: SQL Reference*. Alternate Buffer pool assignments for Type I and Type II areas are made with PROUTIL ENABLEB2. For more information, see *OpenEdge Data Management: Database Administration*.

To select objects for the Alternate Buffer Pool:

1. Choose Admin → Alternate Buffer Pool → Alternate Buffer Pool Maintenance. The Object Selector dialog box appears, as shown:

   ![Object Selector Dialog Box](image)

   2. Scroll through the list in the Object Selector dialog box, to select objects, or click Select Some to bring up the Select Objects by Pattern Match dialog box to refine the list. The initial view displays the tables and indexes of the connected database that are currently assigned to the Primary buffer pool.

   You can change the list to view objects in the Alternate Buffer Pool by selecting Alternate from the Pool(s) Filter, or you can select both Primary and Alternate. Objects not in the Primary buffer pool have an asterisk before the object type.

   3. If you selected too many objects, clicking Deselect Some brings up the Deselect Objects by Pattern Match dialog box, which allows you to enter an object name or pattern you want to deselect.
4. Once you have selected all the objects, click **OK** to proceed. The *Alternate Buffer Pool Maintenance* dialog box appears, as shown in **Figure 10–16**.

![Figure 10–16: Alternate Buffer Pool Maintenance dialog box](image)

5. Select an object, and change the **Buffer Pool** selection to **Alternate**. Click **Save** to make the change, or click **Reset** to undo the change.

   **Note:** The change is not committed to the database until you click **Commit**.

6. Once you save a change, the **Copy** button is activated. Click **Copy** to propagate your change to multiple objects at once with the *Alternate Buffer Pool Maintenance Copy Current Setting To* dialog box, shown in **Figure 10–17**.

![Figure 10–17: Alternate Buffer Pool Maintenance Copy Current Setting To dialog box](image)

7. Continue selecting and modifying object settings until you have specified all the objects you want to change. An asterisk (*) in front of the object name indicates you made changes to that object. Click **Review** at any time to view the status of your changes.

8. When you complete your changes, click **Commit** to commit the changes to the database, or click **Revert**, to cancel all the changes. If you choose to commit your changes, you are asked to confirm your choice.

To remove objects from the Alternate Buffer Pool, follow the same steps as the procedure to select objects for the Alternate Buffer Pool, except in **Step 5**, specify **Primary** for the **Buffer Pool** assignment.
This chapter provides details about the **Utilities** menu and the **PRO/SQL** menu, and also directs you to where you can find more information about the **DataServer** menu and the **Tools** menu, as described in the following sections:

- DataServer menu
- Utilities menu
- PRO/SQL menu options
- Tools menu
Choose the DataServer menu to manage your DataServers. Table 11–1 describes the menu that appears when you choose this option. For more information on these menu options, see the applicable DataServer documentation:

- For the Dataserver for MS SQL Server, see *OpenEdge Data Management: DataServer for Microsoft SQL Server*
- For the Dataserver for Oracle, see *OpenEdge Data Management: DataServer for Oracle*
- For the Dataserver for ODBC, see *OpenEdge Data Management: DataServer for ODBC*

**Table 11–1: DataServer menu**

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Task performed . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS SQL Server Utilities →</td>
<td>Accesses utilities for managing MS SQL Server DataServers</td>
</tr>
<tr>
<td>ODBC Utilities →</td>
<td>Accesses utilities for managing ODBC DataServers</td>
</tr>
<tr>
<td>ORACLE Utilities →</td>
<td>Accesses utilities for managing ORACLE DataServers</td>
</tr>
</tbody>
</table>
Utilities menu

Options on the Utilities menu allow you to edit parameter files, freeze and unfreeze database tables, access the Quoter utility, generate include files for use with ABL, edit auto-connect lists, deactivate indexes, and view information about the database and the current session parameters.

Table 11–2 describes the menu that appears when you choose this option. The sections that follow provide more detailed descriptions of the options.

Table 11–2: Utilities menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Task performed . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Editor for Parameter Files</strong>...</td>
<td>Edits a parameter file</td>
</tr>
<tr>
<td><strong>Quoter Functions</strong>→</td>
<td>Formats a data file so that OpenEdge can read the data into an OpenEdge database</td>
</tr>
<tr>
<td><strong>Generate Include Files</strong>→</td>
<td>Generates an include file</td>
</tr>
<tr>
<td><strong>Edit OpenEdge Auto-Connect List</strong></td>
<td>Prepares a list of secondary OpenEdge databases to which OpenEdge connects automatically, as required when you run a procedure from a primary database</td>
</tr>
<tr>
<td><strong>Freeze/Unfreeze</strong>...</td>
<td>Freezes or unfreezes a table</td>
</tr>
<tr>
<td><strong>Index Deactivation</strong>...</td>
<td>Deactivates indexes</td>
</tr>
<tr>
<td><strong>Information</strong>...</td>
<td>Displays general information about the database and the OpenEdge product you are using</td>
</tr>
</tbody>
</table>

Editor for Parameter Files option

Choose Utilities→ **Editor for Parameter Files** to view a complete list of startup parameters, fill in the desired values, and write out a .pf file (rather than typing in the parameter file with an ordinary text editor). When you choose this option, the Data Dictionary prompts you for the name of the .pf file to edit or create. After you specify the filename, the Edit Parameter File (param.pf) dialog box shown in Figure 11–1 appears.

Figure 11–1: Edit Parameter File dialog box
Specify the parameters you want, then press **GO**. The Data Dictionary displays a dialog box similar to the one shown in Figure 11–2. Enter comments, then choose **OK**.

![Parameter File Comments dialog box](image)

**Figure 11–2:** Parameter File Comments dialog box

### Quoter Functions option

Choose this option to format a data file so that the Data Dictionary can read the data into an OpenEdge database. For more information about the quoter utility, see *OpenEdge Development: Programming Interfaces*.

Table 11–3 describes the menu that appears when you choose this option. The sections that follow provide more detailed descriptions of the options.

**Table 11–3:** Quoter Functions menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Task performed . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entire Lines...</td>
<td>Copies your input file to a new file formatted with quotes around each line</td>
</tr>
<tr>
<td>By Delimiter...</td>
<td>Copies your input file to a new file formatted with quotes around each delimited field</td>
</tr>
<tr>
<td>By Column Ranges...</td>
<td>Copies your input file to a new file formatted with quotes around each column-defined field</td>
</tr>
<tr>
<td>Quoter Include File...</td>
<td>Creates an include file that contains a quoter command</td>
</tr>
</tbody>
</table>
Entire Lines option

Choose Quoter Functions→Entire Lines to copy your input file to a new file formatted with quotes around each line, ready for input. When you choose this option, the Quote Entire Lines dialog box shown in Figure 11–3 appears.

![Quote Entire Lines dialog box](image)

The Quote Entire Lines dialog box contains the following fields:

- **Input File** — Specifies the filename of the input file to format
- **Formatted File** — Specifies the filename of the new formatted file
- **Files** — Displays a dialog box that lets you select a file from a list of existing files

By Delimiter option

Choose Quoter Functions→By Delimiter to copy your input file to a new file formatted with quotes around each delimited field, ready for input. When you choose this option, the Quote By Delimiter dialog box shown in Figure 11–4 appears.

![Quote By Delimiter dialog box](image)

The Quote By Delimiter dialog box contains the following fields:

- **Input File** — Specifies the filename of the input file to format
- **Formatted File** — Specifies the filename of the new formatted file
- **Delimiting Character** — Specifies the character with which to delimit each field
- **Files** — Displays a dialog box that lets you select a file from a list of existing files
By Column Ranges option

Choose Quoter Functions → By Column Ranges to copy your input file to a new file formatted with quotes around each column-defined field, ready for input. When you choose this option, the Quote By Column Ranges dialog box shown in Figure 11–5 appears.

The Quote By Column Ranges dialog box contains the following fields:

- **Input File** — Specifies the filename of the input file to format
- **Formatted File** — Specifies the filename of the new formatted file
- **Files** — Displays a dialog box that lets you select a file from a list of existing files

Generate Include Files option

Choose this option to quickly prepare an include file to store language statements. Because the ASSIGN, FORM, and DEFINE WORK-TABLE statements can be long and complicated, you might want to store them in an include file. Also, particular instances of these statements are often repeated in different procedures, and using an include file lets you easily insert them into any ABL procedure. See the following reference entries in OpenEdge Development: ABL Reference for more information: Include File, ASSIGN statement, FORM statement, and DEFINE WORK-TABLE statement.

Table 11–4 describes the menu that appears when you choose the Generate Include Files option. The sections that follow provide more detailed descriptions of the options.

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Task performed . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN Statement ...</td>
<td>Generates an ASSIGN statement to copy a table</td>
</tr>
<tr>
<td>FORM Statement ...</td>
<td>Generates a FORM statement for a table</td>
</tr>
<tr>
<td>DEFINE WORK-TABLE Statement ...</td>
<td>Generates a WORK-TABLE statement for a table</td>
</tr>
</tbody>
</table>
ASSIGN Statement option

Use this option to generate an ASSIGN statement to copy a table. The Data Dictionary alphabetically lists all the tables defined for your database. When you choose a table, the Data Dictionary prompts you for the filename of the output file. The default filename is `table-name.i`. The Data Dictionary creates an output file that contains an ASSIGN statement to copy each field of the table from one buffer record to another and prompts you to continue.

Figure 11–6 shows an example of the ASSIGN statement generated for the Customer table in the Sports database.

```plaintext
/* 22/09/06 COPY assignment */

DO:
  ASSIGN
    {1}.Cust-Num  = {2}.Cust-Num
    {1}.Country   = {2}.Country
    {1}.Name      = {2}.Name
    {1}.Address   = {2}.Address
    {1}.Address2  = {2}.Address2
    {1}.City      = {2}.City
    {1}.State     = {2}.State
    {1}.Postal-Code = {2}.Postal-Code
    {1}.Contact   = {2}.Contact
    {1}.Phone     = {2}.Phone
    {1}.Sales-Rep = {2}.Sales-Rep
    {1}.Credit-Limit = {2}.Credit-limit
    {1}.Balance   = {2}.Balance
    {1}.Terms     = {2}.Terms
    {1}.Discount  = {2}.Discount
    {1}.Comments  = {2}.Comments.
END.
```

Figure 11–6: ASSIGN statement example

FORM Statement option

Choose Generate Include Files→FORM Statement to generate a FORM statement for a table. The Data Dictionary alphabetically lists all the tables defined for your database. When you select a table, the Generate FORM for “selected-table-name” dialog box appears, as shown in Figure 11–7 appears.

![Generate FORM for "Customer" dialog box](image)

Figure 11–7: Generate FORM for “Customer” dialog box
The **Generate FORM** dialog box contains the following fields:

- **Output File** — Specifies the filename of the output file. The default filename is the table name with a `.f` extension.

- **Fully Qualify Names** — Specifies whether to qualify field names with table names (for example, `table-name.field-name`). The default value is **No**.

- **Fully Expand Arrays** — Specifies whether to fully expand arrays. For example, `X[1]` is expanded and `X` is not expanded. The default value is **Yes**.

- **Frame Type** — Specifies the type of frame to use. The options are **Normal**, **Dialog**, **Overlay**, or **Top-Only**. The default value is **Normal**.

- **Formatting** — Specifies the formatting to use. The options are **Dictionary** or **Explicit**. **Dictionary** specifies to use the Data Dictionary default formats without listing them in the FORM statement. **Explicit** specifies to explicitly list the format definitions in the FORM statement. The default value is **Explicit**.

- **Validation** — Specifies what type of validation to use. The options are **Dictionary**, **Explicit**, or **None**. The default value is **Dictionary**. **Dictionary** specifies to use the Data Dictionary default validation without listing them in the FORM statement. **Explicit** specifies to explicitly list the validation definitions in the FORM statement. **None** specifies that you do not want to use validation from the Dictionary or from FORM statements.

- **Labeling** — Specifies what type of labeling to use. The options are **Dictionary**, **Explicit**, or **None**. The default value is **Explicit**. **Dictionary** specifies to use the Data Dictionary default labels without listing them in the FORM statement. **Explicit** specifies to explicitly list the label definitions in the FORM statement. **None** specifies that you do not want to use any labels.

- **Labels** — Specifies the label position. The options are **Side**, **Top**, or **None**. The default value is **Side**.
Utilities menu

Figure 11–8 shows an example of the FORM statement generated for the Customer table in the Sports database.

```plaintext
/* 22/09/06 FORM LIKE table Customer */

FORM
  Cust-Num FORMAT ">>>>9"
  Country FORMAT "x(20)"
  Name FORMAT "x(20)"
  Address FORMAT "x(30)"
  Address2 FORMAT "x(5)"
  City FORMAT "x(12)"
  State FORMAT "x(20)"
  Postal-Code FORMAT "x(5)"
  Contact FORMAT "x(20)"
  Phone FORMAT "x(20)"
  Sales-Rep FORMAT "X(4)"
  Credit-Limit FORMAT "->,>>>,.9999"
  Balance FORMAT "->,>>>,.9999"
  Terms FORMAT "x(20)"
  Discount FORMAT ">>%"
  Comments FORMAT "x(60)"
WITH FRAME customer SIDE-LABELS.
```

**Figure 11–8: FORM statement example**

**DEFINE WORK-TABLE Statement option**

Use this option to generate a DEFINE WORK-TABLE statement for a table. The Data Dictionary alphabetically lists all the tables defined for your database. When you choose a table, the Data Dictionary prompts you for the filename of the output file. The default filename is `table-name.i`. The Data Dictionary displays a status message and prompts you to continue.

Figure 11–9 shows an example of the DEFINE WORK-TABLE statement generated for the Customer table in the Sports database.

```plaintext
/* 22/09/06 workfile definition for table Customer */
/* 22 = "", 'NEW' or 'NEW SHARED' */
/* 12 = "" or 'NO-SAME' */

DEFINE (1) WORK-TABLE Customer (2) /* LIKE Customer */
FIELD Address   AS CHARACTER FORMAT "x(30)"
FIELD Address2  AS CHARACTER FORMAT "x(5)"
FIELD Balance   AS DECIMAL DECIMALS 2 FORMAT "->,>>>,.9999"
FIELD City      AS CHARACTER FORMAT "x(12)"
FIELD Comments  AS CHARACTER FORMAT "x(60)"
FIELD Contact   AS CHARACTER FORMAT "x(20)"
FIELD Country   AS CHARACTER FORMAT "x(20)" Initial "USA"
FIELD Credit-Limit AS DECIMAL DECIMALS 2 FORMAT "->,>>>,.9999" Initial 1500
FIELD Cust-Num  AS INTEGER FORMAT ">>>>9"
FIELD Discount  AS INTEGER FORMAT ">>>>9"
FIELD Name     AS CHARACTER FORMAT "x(20)"
FIELD Phone    AS CHARACTER FORMAT "x(20)"
FIELD Postal-Code AS CHARACTER FORMAT "x(5)"
FIELD Sales-Rep AS CHARACTER FORMAT "X(4)"
FIELD State    AS CHARACTER FORMAT "x(20)"
FIELD Terms    AS CHARACTER FORMAT "x(20)" Initial "Net30".
```

**Figure 11–9: DEFINE WORK-TABLE statement example**
Edit OpenEdge Auto-Connect List option

Choose Utilities→Edit Auto-Connect List to create or modify the auto-connect list. The auto-connect list is a list of secondary databases (with connection parameters) that are connected automatically as required during program execution on a primary database. You can define one list for each OpenEdge database.

Note: To use the Auto-Connect feature for non-OpenEdge databases, use the Edit Connection Parameters option under the appropriate DataServer submenu option. See the appropriate DataServer manual for information on auto-connecting to non-OpenEdge databases.

The auto-connect feature uses information stored in a primary application database to connect to a secondary application database. The primary application database contains the database connection information for the secondary database. When data from the second database is referenced in a compiled application at run time, the primary application database must already be connected before the database engine can execute the auto-connect for the secondary application database. The database engine executes an auto-connect immediately prior to running a procedure that accesses a database on the auto-connect list.

If you connect a database with the CONNECT statement, and that database also has an auto-connect entry in an already connected database, the CONNECT statement uses the information from the auto-connect list; however, the explicit connection information in the CONNECT statement takes precedence. For more information about auto-connect, see the chapter on database access in OpenEdge Development: Programming Interfaces.

When you choose this option, the Edit OpenEdge Auto-Connect List dialog box shown in Figure 11–10 appears.

![Edit OpenEdge Auto-Connect List dialog box](image)
The **Edit OpenEdge Auto-Connect List** dialog box contains the following user-interface elements:

- **Database Name** — Displays the list of databases to which the database engine connects automatically.

- **Logical Database Name** — Specifies the logical name of the database highlighted in the Database Name list.

- **Physical Database Name** — Specifies the physical name of the database highlighted in the Database Name list.

- **CONNECT statement parameters for auto-connect** — Specifies any other startup parameters for the database that are not included in the parameter file.

Table 11–5 describes the buttons on the **Edit OpenEdge Auto-Connect List** dialog box.

### Table 11–5: Edit OpenEdge Auto-Connect List options

<table>
<thead>
<tr>
<th>Option</th>
<th>Task performed...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next</td>
<td>Displays the next connect record</td>
</tr>
<tr>
<td>Prev</td>
<td>Displays the previous connect record</td>
</tr>
<tr>
<td>First</td>
<td>Displays the first connect record</td>
</tr>
<tr>
<td>Last</td>
<td>Displays the last connect record</td>
</tr>
<tr>
<td>Add</td>
<td>Adds a new connect record</td>
</tr>
<tr>
<td>Modify</td>
<td>Modifies the selected connect record</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected connect record</td>
</tr>
<tr>
<td>Undo</td>
<td>Exits and undoes all changes you made in this session to the Auto Connect list</td>
</tr>
<tr>
<td>Exit</td>
<td>Exits and saves changes</td>
</tr>
</tbody>
</table>

**Freeze/Unfreeze Table option**

Choose **Utilities** → **Freeze/Unfreeze** to freeze or unfreeze a database table. Freezing a database table prevents users from changing the table, field, or index definitions for the table. The Data Dictionary alphabetically lists all the tables defined for your database. When you select a table, the **Freeze/Unfreeze Table** dialog box shown in **Figure 11–11** appears.
Index Deactivation option

Choose Utilities→Index Deactivation to deactivate indexes. When you choose this option, the Index Deactivation Warning dialog box shown in Figure 11–12 appears.

![Index Deactivation Warning dialog box]

Figure 11–12: Index Deactivation Warning dialog box

After you read the information, choose OK. The Data Dictionary alphabetically lists all the tables defined for the database. Select the table for which you want to deactivate indexes or enter ALL to deactivate all your indexes. If you specified an individual table, the Data Dictionary lists the indexes defined for that table. Select the indexes you want to deactivate, then press GO. The Data Dictionary prompts you to verify your entry.

Information option

Choose Utilities→Information to display general information about the database and the OpenEdge product you are using. Information displayed includes the number of connected databases, physical and logical names of the selected database, whether the selected database is connected, and if 64-bit sequences and large key entries are enabled. To send the report to a file or printer, choose Print. The Data Dictionary prompts you for the printer or filename, whether you want to append the report to an existing file, and the page length.

To exit the Information window, choose OK. The Data Dictionary displays the video attribute settings that determine how the information appears on the monitor.
PRO/SQL menu options

Choose the PRO/SQL menu options to manage views created with Progress/SQL statements.

Table 11–6 describes the menu that appears when you choose this option. The sections that follow provide more detailed descriptions of the options.

Table 11–6: PRO/SQL menu

<table>
<thead>
<tr>
<th>Menu option</th>
<th>Task performed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRO/SQL View Report...</td>
<td>Lists information stored for views</td>
</tr>
<tr>
<td>Dump as CREATE VIEW Statement...</td>
<td>Generates SQL CREATE VIEW statements</td>
</tr>
<tr>
<td>Dump as CREATE TABLE Statement...</td>
<td>Generates SQL CREATE TABLE statements</td>
</tr>
</tbody>
</table>

**PRO/SQL View Report option**

Choose this option to see the information about all SQL views defined for the working database.

**Note:** Views of non-OpenEdge databases are treated as tables. To display information for these views, use the table reports option. See Chapter 8, “Database Menu Reference,” for more information on generating a table report.

When you choose PRO/SQL→PRO/SQL View Report, the Data Dictionary alphabetically lists all the views defined for your database. Select the views for which you want information. Figure 11–13 shows the first portion of a View Report window.

![View Report dialog box](image)

**Figure 11–13: View Report dialog box**
The View Report window contains the following fields:

- **View Name** — The name of the view
- **Updatable** — Whether the view is updatable
- **Group By** — Whether the view is defined with a `GROUP BY` clause
- **Check Option** — Whether the view is defined with a `CHECK OPTION WHERE` clause
- **View Name** — View name of the currently displayed view
- **Base Tables** — Tables from which the view gets its information
- **Where Clause** — The `WHERE` clause
- **View Def** — The SQL statement executed when defining the view
- **Can-Read** — Users who can read the rows of the view
- **Can-Write** — Users who can update records of the view
- **Can-Create** — Users who can create new records
- **Can-Delete** — Users who can delete records from the view
- **Order** — Order of the columns within the view
- **View Column Name** — Names of the columns that comprise the view
- **Base Columns** — Base table field names for each view column

To send the report to a file or printer, choose **Print**. The Data Dictionary prompts you for the printer or filename, whether you want to append the report to an existing file, and the page length.

**Dump as CREATE VIEW Statement option**

Choose PRO/SQL → **Dump as CREATE VIEW Statement** to generate an SQL DDL program that contains `CREATE VIEW` statements equivalent to those originally used to define the selected view.

**Note:** Views of non-OpenEdge databases are treated as tables. To display information for these views, use the table reports option. See the “Reports option” section on page 8–8 for more information on generating a table report.
When you choose this option, the Data Dictionary alphabetically lists all the views defined for your database. Select the views for which you want make CREATE VIEW statements. The Dump CREATE VIEW of All Views dialog box shown in Figure 11–14 appears.

![Dump CREATE VIEW of All Views](image)

This program generates a SQL DDL program containing CREATE VIEW statements equivalent to those originally used to define the view. The generated program makes the assumption that all necessary permissions are available. It does NOT generate any GRANT or REVOKE statements to set permissions on the views.

After generating the file, the Data Dictionary prompts you to continue.

**Dump as CREATE TABLE Statements option**

Choose PRO/SQL → Dump as CREATE TABLE and CREATE INDEX statements to generate an SQL DDL program that contains CREATE TABLE statements equivalent to those originally used to define the selected table.

**Note:** This option also works with non-OpenEdge databases.
When you choose this option, the Data Dictionary alphabetically lists all the tables defined for your database. Select the table for which you want to make CREATE TABLE statements. The **Dump CREATE TABLE** dialog box appears, as shown in Figure 11–15.

![Figure 11–15: Dump CREATE TABLE dialog box](image)

The **Dump CREATE TABLE** dialog box contains the following fields:

- **Output File for CREATE TABLE** — Specifies the filename to which you want to write the CREATE TABLE statements. The default filename for a single table is `table-name.p`. If you specify all tables, the default filename is `database-name.p`.

- **Output File for CREATE INDEX** — Specifies the filename to which you want to write the CREATE INDEX statements. The default filename for a single table is `table-name.i`. If you specify all tables, the default filename is `database-name.i`. If you do not specify a filename, the Data Dictionary writes the CREATE INDEX statements to the CREATE TABLE output file.

- **SQL-Flavor** — Specifies the type of SQL you want to use: OpenEdge, ORACLE, or MS SQL Server. When ORACLE is selected, any field defined as `INT64` is converted to the `NUMBER` data type. When MS SQL Server is selected, any field defined as `INT64` is converted to `BIGINT` data type.

- **Create Recid Field** — Select this option if your database currently contains the `PROGRESS_RECID` field. Selecting this option will maintain the use of `PROGRESS_RECID` in any new tables added by this utility.

- **Create Shadow Columns** — If your DataServer is configured with a case sensitive code page, this box is available for selection. Selecting it provides case insensitivity compatible with the behavior of an OpenEdge database.

- **Include Defaults** — Select if you want the OpenEdge initial value used as the default in the SQL script.

- **Use Unicode Types** — Select if you want Unicode data types preserved.

- **Expand Width (utf-8)** — Select if you want field widths expanded to hold utf-8 encoded characters.

- **Use Revised Sequence Generator** — Select if you want sequences to be revised.

- **Map to MSS ‘Datetime’ type** — Select if you have dates you want converted to the MS SQL Server Datetime data type.
• **For field widths use Width** — Select if your OpenEdge database contains data that is larger than the display format. Change the width to the proper size. The utility will use the Width field instead of the display format.

• **For field widths use ABL Format** — Select if you want the Data Dictionary to use the display format instead of the field’s width.

• **Expand x(8) to 30** — If selected, indicates that fields with FORMAT set to x(8) will be given a default width of 30 characters.

• **For Create RECID use 1.Trigger** — Select if you want a trigger to determine the RECID value.

• **For Create RECID use 2.Computed column** — Select if you want a computed column to determine the RECID value.

After generating the file, the Data Dictionary prompts you to continue.
Tools menu

The Tools menu lets you access other OpenEdge tools. See OpenEdge Development: Basic Development Tools for more information about the menu options.
## Index

**A**

Abbr attribute  
Index Editor 9–31

Actv attribute  
Index Editor 9–31

Add field option  
Field Editor 9–12

Add new table option  
Schema menu 9–9

Administrators  
security 10–29

Alternate Buffer Pool Maintenance Copy  
Current Setting To dialog box 10–49

Alternate Buffer Pool Maintenance dialog box 10–49

Asc attribute  
Index Editor 9–31

ASCII option  
Export menu 10–39, 10–40

ASSIGN statement  
generating include files 11–7

ASSIGN trigger 9–25

Attributes  
abbr 9–31  
actv 9–31  
asc 9–31  
BY-clause 10–41  
case-sensitive 9–24  
check CRC 9–26  
column-label 9–19  
CREATE TABLE statement 11–16  
CREATE VIEW statement 11–15  
cycle at limit 9–35  
databases 11–11  
connecting 8–5  
creating 8–4  
DataServer name 9–4, 9–35  
data-type 9–20  
decimals 9–24  
desc 9–25  
description 9–4  
disable triggers 10–41  
dump file 9–3  
export 10–40  
extent 9–23  
field 9–12 to 9–25  
field name 9–12, 9–31  
FORM statement 11–8  
format 9–12  
frozen 9–3  
help 9–25  
hidden 9–3  
import file 10–43  
increment by 9–34  
index 9–30 to 9–31  
initial 9–19  
initial value 9–34  
label 9–3, 9–7, 9–18  
labeling 11–8  
lower limit 9–34  
mandatory 9–24  
name 9–34  
order 9–24  
output file 11–8, 11–16  
overrideable 9–26  
owner 9–35  
record size 9–3  
seq 9–31
sequences 9–34 to 9–35
String 9–7
table type 9–3
tables 9–3
type 9–31
unique 9–30
upper limit 9–34
valexpr 9–24
valmsg 9–25
WHERE-clause 10–41
word 9–31

Auto-connect lists
auto-connect records only option 10–7
editing 11–10

B

Batch Incremental utility 10–12
Binary dump and load 3–15
Bulk loader description files
creating 3–17
By column ranges option
Quoter Functions menu 11–6
By delimiter option
Quoter Functions menu 11–5
BY-clause
data export attribute 10–41

C

Can-Create
table permission 10–27
Can-Delete
table permission 10–27
Can-Dump
table permission 10–27
Can-Load
table permission 10–27
Can-Read
table permission 10–27
Can-Write
table permission 10–27
Case-sensitive attribute
Field Editor 9–24
Case-sensitivity
names 9–3
Changing passwords
Security menu 10–26

Character format
Field Editor 9–13
Character mapping 3–11, 10–5
Check CRC attribute
triggers 9–26
Collation tables option
dump data and definitions menu 10–7
Column ranges 11–6
Column-Label attribute
Field Editor 9–19
Components
index 9–20
view 9–20
Copy Encryption Policy Setting To dialog
box 10–36
Copying
fields 4–6
CRC code 8–12
CREATE
database events 9–5
CREATE TABLE statement 11–15
attributes 11–16
CREATE VIEW statement 11–14
attributes 11–15
Creating
auto-connect lists 11–10
bulk loader description files 3–17
databases 8–4
fields 4–2
include files 3–18
incremental data definition files 2–8,
10–12
indexes 5–2
parameter files 2–9
reports 3–7
detailed table 3–7
field 4–9
index 5–7
quick table 3–9
sequence 6–9
table relations 3–10
triggers 7–6
schema triggers 7–2
sequences 6–3
tables 3–2, 9–9
Current database
selecting 8–3
Cycle at limit attribute
Sequence Editor 9–35
Cyclic Redundancy Check (CRC) identifier
9–5, 9–26

Cyclical redundancy check. See CRC code

D

Data
dumping 3–11
loading 3–13

Data Administration
loading database contents 3–13

Data definition files
creating an incremental file 2–8
dumping 2–7
loading 2–11

Data Dictionary
character
defining fields 4–2
examining field definitions 4–2

Database menu 8–2
DataServer menu 11–2
dumping
database contents 3–11
main window 1–2
menu bar 1–3
PRO/SQL menu 11–13
starting 1–2
Utilities menu 11–3

Data files
formatting 11–4

Data types 9–13, 9–20
BLOB 9–20
character 9–13, 9–20
CLOB 9–20
date 9–16, 9–20
DATETIME 9–17, 9–20
DATETIME-TZ 9–17, 9–20
decimal 9–14, 9–21
initial values 9–19
INT64 9–21
integer 9–14, 9–20
logical 9–18, 9–21
raw 9–21
recid 9–21

Database contents
dumping 3–11
loading 3–13

Database data
dumping 10–3
exporting 10–39
formats 10–39
ASCII option 10–40
dBASE definitions option 10–44
dBASE file contents option 10–42
delimited ASCII option 10–42
DIF option 10–42
fixed-length option 10–43
SYLK option 10–42
importing 10–42
loading 10–14
security 10–26
table contents (.d file) option 10–4

Database definitions
data definitions (.df file) option 10–4
dumping 10–3
loading 10–14

Database events
CREATE 9–5
DELETE 9–5
FIND 9–5
REPLICATION CREATE 9–6
REPLICATION DELETE 9–6
REPLICATION WRITE 9–6
WRITE 9–6

Database menu
Data Dictionary 8–2

Database types
database attribute 8–5

Databases
attributes 11–11
connecting 8–5
comparing 10–12
copying 8–4
creating 8–4
disconnecting 8–7
dumping
contents 3–11
empty 8–4
generating reports 8–8
information option 11–12
loading contents 3–13
replacing 8–4
selecting 8–3
viewing session information 2–6

DataServer menu
Data Dictionary 11–2

DataServer name
Sequence Editor 9–35

DataServer name attribute 9–4

Data-Type attribute
field editor 9–20

Date format
Field Editor 9–16

dBASE definitions option
Import menu 10–44
Index—4

DIF option
Import menu 10–42

Deactivating indexes 5–5, 11–12

Decimal format
Field Editor 9–14

Decimals attribute
Field Editor 9–24

DEFINE WORKTABLE statement 11–9

DELETE
database events 9–5

Deleting
fields 4–7
indexes 5–6
sequences 6–5
tables 3–5, 9–10
triggers 7–5

Delimited ASCII option
Import menu 10–42

Delimiting
fields 11–5

Desc attribute
Field Editor 9–25

Description attribute 9–4

Detailed Encryption Policy Report 8–25

Detailed table reports 8–9

Dialog boxes
Alternate Buffer Pool Maintenance 10–49
Alternate Buffer Pool Maintenance Copy Current Setting To 10–49
Copy Encryption Policy Setting To 10–36
create database 8–4
database connect 8–5
Detailed Encryption Policy Report 8–25
detailed table report 8–9
dump CREATE TABLE 11–16
dump CREATE VIEW 11–15
dump data contents for some tables 10–4
edit data security 10–26
Edit Encryption Policy 10–35
edit parameter file 11–3
Encryption Policy History 10–38
export DIF 10–39
table triggers 9–25
fixed length field columns 10–44
freeze/unfreeze table 11–11
Generate Encryption Keys 10–37
generate FORM 11–7

global field name change 9–29
import definitions 10–44
import DIF 10–42
import fixed length 10–43
index deactivation 11–12
Object Selector for Alternate Buffer Pool Maintenance 10–48
Object selector for Encryption Policies 10–35
output record formatting 10–40
parameter file comments 11–4
Quick Alternate Buffer Pool Report 8–15
Quick Encryption Policy Report 8–24
quick field report 8–10
quick index report 8–11
quick table report 8–10
quote by column 11–6
quote by delimiter 11–5
quote entire lines 11–5
resequence order of fields 9–28
security administrators 10–29
select working database 8–3
sequence report 8–12
table relations report 8–13, 8–14, 8–24, 8–25
table string attributes 9–7
table triggers 9–5
table validation 9–4
trigger code 9–6
trigger report 8–12
user report 8–13, 10–30
view-as phrase 9–26

DIF option
Export menu 10–39
Import menu 10–42

Disable triggers
data export attribute 10–41

Disconnecting
databases 8–7

Dump as CREATE TABLE statement option
PRO/SQL menu 11–15

Dump file attribute 9–3

Dump views of userid
CREATE VIEW statement attribute 11–15

Dumping
data and definitions 10–3
data definition files 2–7
database contents 3–11
SQL views 11–14
user table contents 10–5, 10–17
E

Edit Encryption Policy dialog box 10–35

Editing
  auto-connect lists 11–10
  data security 10–26
  fields 9–11
  indexes 9–30
  parameter files 2–10
  sequences 9–34
  user lists 10–25

Editor for parameter files option
  Utilities menu 11–3

Encryption Policy History dialog box 10–38

Entire lines option
  Quoter Functions menu 11–5

Error files 10–18

Exit menu
  Data Dictionary 8–26

Export
  attributes 10–40 to 10–41

Export data option
  Admin menu 10–39

Exporting
  fields 10–41

Extent attribute
  Field Editor 9–23

F

Field delimiter
  data export attribute 10–41

Field Editor
  go index option 9–27
  Schema menu 9–11
  triggers option 9–25
  view as option 9–26

Field name attribute
  Field Editor 9–12
  Index Editor 9–31

Field reports 4–9

Field separator 10–41, 10–43

Fields
  adding 9–12
  changing field order 4–8

  changing names globally 9–29
  columns 11–6
  copying 4–6
  creating 4–2
  defining 4–2
  deleting 4–7
  delimiting 11–5
  editing 9–11
  exporting 10–41
  generating reports 8–10
  modifying 4–3
  reordering 9–28
  triggers 9–25

Files
  bulk loader description 3–17
  formatting 11–4
  include file
    generating 3–18
  parameters
    creating 2–9
    editing 2–10

FIND
  database events 9–5

Fixed-length option
  Import menu 10–43

FORM statement
  attributes 11–8
  generating include files 11–7

Format attribute
  Field Editor 9–12

Formats
  character 9–13
  date 9–16
  decimal 9–14
  integer 9–14
  logical 9–18

Formatting
  data files 11–4
  FORM statement attribute 11–8
  text 10–40

Frame type
  FORM statement attribute 11–8

Freezing and unfreezing tables 3–6, 11–11

Frozen attribute 9–3

Fully expand arrays
  FORM statement attribute 11–8

Fully qualify names
  FORM statement attribute 11–8
G

Generate Encryption Keys dialog box 10–37

Generate include files option
Utilities menu 11–6

Generating
include files 3–18, 11–6
reports 3–7
database 8–8
detailed table 3–7, 8–9
field 4–9
index 5–7
quick field 8–10
quick index 8–11
quick table 3–9, 8–10
sequence 6–9, 8–12
table relations 3–10, 8–13
triggers 7–6, 8–12
user 8–12

Global field name change option
Schema menu 9–29

GoIndex option
Field Editor 9–27

H

Help attribute
Field Editor 9–25

Hidden attribute 9–3

I

Import file attribute 10–43
Import menu 10–42 to 10–44
fixed-length option 10–43
Include files
generating 3–18, 11–6 to 11–9

Increment by attribute
Sequence Editor 9–34

Incremental .df file
creating 10–12

Index Editor
accessing 9–27, 9–30

Indexes
attributes 9–30 to 9–31
creating 5–2
deactivating 5–5, 11–12
defining
using multiple files 9–30
deleting 5–6

ingenerating reports 5–7, 8–11
modifying 5–4
renaming 5–4

Information option
Utilities menu 11–12

Initial attribute
Field Editor 9–19

Initial value attribute
Sequence Editor 9–34

INT64 data type 9–21

Integer format
Field Editor 9–14

J

Joins
table reports 8–13

L

Label attribute 9–3
Field Editor 9–18
FORM statement 11–8
table string 9–7

Labeling attribute
FORM statement 11–8

large files 2–7, 2–11

Large key entries, enabling 10–47

Loading
data and definitions 10–14
data definition files 2–11
database contents 3–13
reconstructed bad records 10–18
reconstructing bad records 3–14
sequences current values 10–18

Locking
tables 11–11

Logical format
Field Editor 9–18

Lower limit attribute
Sequence Editor 9–34

M

Mandatory attribute
Field Editor 9–24

MAP. See Character mapping
Menu bar
  Data Dictionary 1–3

Menus
  Data Dictionary
    Database 8–2
    DataServer 11–2
    PRO/SQL 11–13
    Utilities 11–3

Microsoft Word option
  Export menu 10–39

Modifying
  auto-connect lists 11–10
  fields 4–3
  indexes 5–4
    deactivating 5–5
  sequences 6–4
  table schema 9–3
  tables 3–4
    fields 4–3, 4–7, 4–8
    primary index 5–4
    renaming indexes 5–4
  triggers 7–4

Multiple users
  database attribute 8–5

N

Name
  sequence attribute 9–34

Names
  field 9–29

Naming
  indexes 5–4

Non-OpenEdge databases 11–10

O

Object Selector dialog box 10–35

Order attribute
  Field Editor 9–24

Other UNIX-style parameters
  database attribute 8–6, 11–11

Output file attribute 11–15, 11–16

Overrideable attribute
  triggers 9–26

Overriding defaults
  display order 9–24

Owner
  Sequence Editor 9–35

P

Parameter files
  creating 2–9
    database attribute 8–6
  editing 2–10, 11–3

Passwords
  changing 10–26
    database attribute 8–6

Performing
  binary dump 3–15
    binary load 3–16

Permissions
  table 10–27

Physical database names
  attribute 8–4

Primary indexes
  changing 5–4

PRO/SQL menu
  Data Dictionary 11–13

Q

Quick Alternate Buffer Pool Report 8–15

Quick Encryption Policy Report 8–24

Quick field reports 8–10

Quick index reports 8–11

Quick table reports 8–10

R

Reconstructing
  bad load records 3–14, 10–18

Record end string
  data export attribute 10–40

Record size attribute 9–3

Record starter string
  data export attribute 10–40

Records
  reconstructing 10–18

Renaming
  indexes 5–4

Reordering fields 9–28
Replication
CREATE 9–6
DELETE 9–6
WRITE 9–6

Reports
auditing 8–8, 8–16
detailed table 3–7, 8–9
field 4–9
generating 3–7
index 5–7
PRO/SQL view 11–13
quick field 8–10
quick index 8–11
quick table 3–9, 8–10
sequence 6–9, 8–12
table relations 3–10, 8–13
trigger 7–6, 8–12
user 8–12, 10–30

Reports menu 8–8

S

Schema changes
online 2–11

Security
administrators 10–29
changing passwords 10–26
editing data 10–26
editing user lists 10–25
permissions 10–27
user reports 10–30

Security option
Admin menu 10–24

Selecting
working database 8–3

Seq attribute
Index Editor 9–31

Sequence Editor 9–34
attributes 9–34 to 9–35
options 9–35

Sequences
creating 6–3
deleting 6–5
dumping
definitions 10–7
values 10–7
dumping and loading 6–8
generating reports 6–9, 8–12
loading current values 10–18
modifying 6–4
_seqvals.d file 6–8

Session information 11–12

SQL
dumping as CREATE VIEW statements
option 11–14
view report option 11–13

SQL views
dumping 10–5
loading 10–17

Starting
Data Dictionary 1–2

Statement termination character 11–15

Statements
ASSIGN 11–7
CREATE TABLE 11–15
CREATE VIEW 11–14
DEFINE WORK-TABLE 11–9
FORM 11–7, 11–8

String attributes 9–7

SYLK option
Export menu 10–39
Import menu 10–42

T

Table reports
detailed 3–7
quick 3–9
relations 3–10, 8–13

Table type attribute 9–3

Tables
access restrictions 10–27
adding 9–9
attributes 9–3
changing field order 4–8
collation 10–7
converting 10–7
creating 3–2
deleting 3–5, 9–10
dumping as CREATE TABLE statements 11–15
freezing and unfreezing 3–6, 11–11
generating reports 8–9
hidden 9–3
loading
reconstructed bad records 3–14
modifying 3–4, 9–3
adding fields 4–2
copying fields 4–6
deleting fields 4–7
fields 4–3
naming 9–3
owners 9–3
permissions 10–27
string attributes 9–7
translating 10–7
triggers 9–5
validating 3–2, 9–4

Text formats
exporting 10–40

Translation tables 10–7

Trigger locations
database attribute 8–6

Triggers 9–5
attributes 9–26
creating
schema 7–2
database events
CREATE 9–5
DELETE 9–5
FIND 9–5
REPLICATION CREATE 9–6
REPLICATION DELETE 9–6
REPLICATION WRITE 9–6
WRITE 9–6
deleting 7–5
disabling during export 10–41, 10–43
field 9–25
generating reports 7–6, 8–12
modifying 7–4

Type attribute
Index Editor 9–31

U

Unique attribute
Index Editor 9–30

Upper limit attribute
Sequence Editor 9–34

User access 10–29

User IDs
database attribute 8–6

User lists
editing 10–25

User report option
Security menu 10–30

User table contents
dumping 10–5, 10–17

Users
generating reports 8–12

Utilities menu
Data Dictionary 11–3

V

Valexp attribute
Field Editor 9–24

Validation
criteria 3–2, 9–4
FORM statement attribute 11–8
message format, table string attributes 9–7

Valmsg attribute
Field Editor 9–25

View as option
Field Editor 9–26

W

WHERE-clause
data export attribute 10–41

Windows
add new table 9–9
create incremental .df file 10–12edit user lists 10–25
editing auto-connect lists 11–10
field add option 9–12
field editor 9–11
goindex option 9–27
index add option 9–33
Index Editor 9–30
modify table 9–3
Sequence Editor 9–34

Word attribute
Index Editor 9–31

WordPerfect option
Export menu 10–39

WordStar option
Export menu 10–39

Working databases 1–2
selecting 8–3

WRITE
database events 9–6