Copyright Notice

The information in this document is subject to change without notice and does not represent a commitment on the part of Bottomline Technologies (de), Inc. The software described in this document is furnished under a license or non-disclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software onto magnetic tape, disk or any other medium for any purpose other than in accordance with the agreement. It is against the law to copy or distribute this document other than in accordance with the agreement. All rights reserved.

© Copyright 1987-2005 Create!form International, Inc. and Bottomline Technologies (de), Inc. All rights reserved. Create!form, Create!print, Create!flash, Create!stream, Create!fax, Create!email, Create!pdf server and Create!archive are trademarks of Create!form International, Inc. and Bottomline Technologies (de), Inc. which may be registered in certain jurisdictions. PeopleSoft, EnterpriseOne, J.D. Edwards, OneWorld and ERP 8 are trademarks of PeopleSoft, Inc. Adobe, Acrobat, Acrobat Distiller, Acrobat Exchange, Acrobat Reader and PostScript are trademarks of Adobe Systems Incorporated. All other brand/product names are trademarks or registered trademarks of their respective holders.

WARNING. This document is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this document, or any portion of it, may result in civil or criminal liability. Bottomline Technologies (de), Inc. will prosecute and defend its copyrights to the fullest extent of the law.

Order Number:  TR6WN-U11C
# Contents

1 Introducing Create!form Transform .......................... 17
   1.1 Who is this Guide For? ........................................ 18
   1.2 Create!form Transform Features ............................ 18
   1.3 Design Prerequisites ........................................... 18
   1.4 Production Prerequisites ....................................... 19
   1.5 Getting Help .................................................... 19
   1.6 User Guide Conventions ....................................... 19

2 Installing .......................................................... 21
   2.1 Installation Requirements ...................................... 22
   2.2 Installing Create!form Transform ............................ 22
   2.3 Starting Create!form Transform ............................... 22

3 Getting Started .................................................... 23
   3.1 What is a Project? ............................................... 24
   3.2 Project Directories .............................................. 24
      3.2.1 Project Directory ........................................ 24
      3.2.2 Common Project Directory ............................. 25
   3.3 Creating a Project .............................................. 25
      3.3.1 New Project wizard ..................................... 25
      3.3.2 Input file type ........................................... 26
      3.3.3 Using an existing project as a template ................. 26
      3.3.4 DataMap options ......................................... 27
      3.3.5 Project type ............................................. 27
         3.3.5.1 Standard project .................................. 27
3.3.5.2 Simple project ..............................................28
3.3.5.3 Overlay project ............................................28

3.4 Design Flowcharts ..............................................28
3.5 Saving a Project ..................................................32
3.6 Opening a Project .................................................33
3.7 Printing and Previewing Projects.................................33
3.8 Packing Projects ....................................................34
3.9 Transferring Projects to the e-forms Server .....................35
3.10 Managing Project Files ..........................................35

4 Viewing and Navigating ............................................37
4.1 Understanding the Screen Elements ...............................38
  4.1.1 Design Window ...............................................39
  4.1.2 Input Window .................................................39
  4.1.3 Project Tree ..................................................40
  4.1.4 Input Tree ....................................................40
  4.1.5 Thumbnail View ...............................................40
  4.1.6 Input File Source View ......................................40
  4.1.7 Status Bar ....................................................41
  4.2 Navigating in a Project .........................................41
    4.2.1 Jumping directly to a set or page .........................42
    4.2.2 Moving between sections .................................43

5 Input Design ..........................................................45
5.1 Before You Start ..................................................46
  5.1.1 Using the Create!form sample projects ......................46
  5.1.2 Selecting a different input file ................................46
  5.1.3 Viewing the input file source ................................47
  5.2 Key Input Design Concepts .................................47
    5.2.1 What is a DataMap? ......................................47
### 5.6.5.2 Changing the input data format

100

### 5.6.5.3 Copying data variables to the project

100

### 5.7 Designing for JDE PDF Input Files

101

#### 5.7.1 Input file properties

102

#### 5.7.2 Defining sets.

103

#### 5.7.3 Input sections

103

#### 5.7.4 Data variables

103

### 5.8 Creating Derived Variables

104

---

### 6 Project Settings

105

#### 6.1 Project Properties

106

##### 6.1.1 Collate options

106

##### 6.1.2 Duplicates

107

##### 6.1.3 About format styles

107

##### 6.1.3.1 Create a format style

107

##### 6.1.3.2 Customize a format style

109

##### 6.1.3.3 Set the default format style for new objects

109

#### 6.2 Project Page Style

109

#### 6.3 Project Background

110

### 7 Sections

111

#### 7.1 About Sections

112

#### 7.2 Data Sections

112

##### 7.2.1 Data section display options

112

#### 7.3 Header and Footer Sections

113

##### 7.3.1 Create set headers and footers with the wizard

114

##### 7.3.2 Continuation header and footer sections

114

##### 7.3.3 Create and remove header and footer sections

114

##### 7.3.4 Create header and footer sections from an input section

115

##### 7.3.5 The display order of header and footer sections

116

#### 7.4 Alternate Sections

116
7.5 Inserted Sections .................................................. 117
7.6 Setting Section Properties ........................................ 118
  7.6.1 Section positioning options .................................. 118
  7.6.2 Adjusting section size ........................................ 119
  7.6.3 Fixed and auto-expanding sections ......................... 119
  7.6.4 About parent/child relationships ......................... 120
  7.6.5 Keeping sections together .................................. 121
  7.6.6 Setting repagination control ............................... 121
  7.6.7 About sources .............................................. 122
  7.6.8 Adding sources .............................................. 123

8 Tables ................................................................. 125
  8.1 About Tables .................................................... 126
  8.2 Working with Tables ............................................ 126
    8.2.1 Create tables with the wizard ............................ 126
    8.2.2 Create a table ............................................. 126
    8.2.3 Change table properties .................................. 127
    8.2.4 Delete a table ............................................. 128
  8.3 Table Properties ................................................ 128
    8.3.1 Table levels ................................................ 129
    8.3.2 Sorting modes .............................................. 129
    8.3.3 Header and footer sections ............................... 130
    8.3.4 Grouping options ......................................... 130
    8.3.5 About parent/child relationships ....................... 131
    8.3.6 Setting table pagination controls ...................... 132

9 Objects .............................................................. 133
  9.1 About Objects .................................................. 134
    9.1.1 About objects and sections ............................... 134
    9.1.2 Selecting objects in different views .................... 135
    9.1.3 Using variables to control objects ...................... 135
    9.1.4 Window display preferences .............................. 136
9.2 Creating Objects ................................................................. 136
  9.2.1 Adding text to the page .............................................. 137
    9.2.1.1 Insert static text .............................................. 137
    9.2.1.2 Insert static text containing variables ................. 138
    9.2.1.3 Insert a single variable as text ......................... 138
    9.2.1.4 Copy data variables from an input section ............ 139
    9.2.1.5 Drag-and-drop variables from the input window ....... 139
  9.2.2 Adding subforms .................................................... 140
    9.2.2.1 About subforms .............................................. 141
    9.2.2.2 Insert an existing subform ............................... 141
    9.2.2.3 Convert text files into subforms ......................... 142
    9.2.2.4 Create subforms ............................................. 143
    9.2.2.5 Use a variable to display different subforms ........ 143
  9.2.3 Copying objects from other resources ......................... 144
    9.2.3.1 Copy objects from other projects ....................... 144
  9.3 Editing and Formatting Objects ...................................... 144
    9.3.1 Change the display properties of any object .......... 145
    9.3.2 Edit and format the contents of a text object ......... 145
    9.3.3 Format a variable text object ............................. 146
    9.3.4 Change number, date and currency formats ............... 147
    9.3.5 Edit the contents of a subform ............................ 148
  9.4 Setting Object Properties ........................................... 148
    9.4.1 Display and edit object properties ....................... 149
    9.4.2 Edit the properties of multiple objects ................ 149
    9.4.3 About property classes ...................................... 150
      9.4.3.1 General properties ..................................... 151
      9.4.3.2 Variable properties ................................... 152
      9.4.3.3 Text properties ........................................ 152
      9.4.3.4 Subform properties .................................... 154
      9.4.3.5 Subform variable properties .......................... 154
    9.4.4 Set default properties ...................................... 154
  9.5 Selecting and Editing Objects in Graphics View ................ 155
    9.5.1 Select objects .................................................. 155
### 9.5 Hidden Objects

- **9.5.2 Select hidden objects** .................................................. 155
- **9.5.3 Edit objects on sections that are not displayed** ............... 156
- **9.5.4 Move objects** ................................................................. 157
- **9.5.5 Resize objects** ................................................................. 157
- **9.5.6 Copy objects** ................................................................. 158
- **9.5.7 Align objects** ................................................................. 158
- **9.5.8 Advanced object selection** ............................................ 160

### 10 Variables

- **10.1 About Variables** ............................................................ 164
- **10.2 Data Variables and Derived Variables** ........................... 164
- **10.3 User Variables** ............................................................... 164
- **10.4 Global User Variables** ................................................... 166
- **10.5 System Variables** ........................................................... 167
- **10.6 Lookup Variables** .......................................................... 168
- **10.7 Environment Variables** .................................................. 169
- **10.8 Job Ticket Variables** ...................................................... 170

### 11 Expressions and Conditions

- **11.1 Expressions** ................................................................. 172
  - **11.1.1 Building Expressions** ............................................. 172
  - **11.1.2 Using the Expression Builder** ................................. 172
  - **11.1.3 Navigating and Editing Expressions** ........................ 173
- **11.2 Conditions** ................................................................. 173
  - **11.2.1 Building Conditions** ............................................. 174
  - **11.2.2 Using Expressions as Conditions** .......................... 174
- **11.3 Building Blocks** .......................................................... 174
  - **11.3.1 Boolean Operators** .............................................. 175
  - **11.3.2 Operators** ............................................................. 175
  - **11.3.3 Functions** ............................................................. 177
11.3.4 Variables .................................................. 181
11.3.5 System Conditions ......................................... 182
11.4 Syntax and Order of Precedence Rules ...................... 182
  11.4.1 Mixed Data Types ........................................ 183
  11.4.2 Addition Operator ....................................... 183
  11.4.3 Subtraction Operator .................................... 183
  11.4.4 Comparing Text Strings .................................. 184
  11.4.5 Order of Precedence ..................................... 184
  11.4.6 Nested Functions ........................................ 184
  11.4.7 Parsing .................................................. 185
  11.4.8 Names of Variables ...................................... 185
11.5 Summarizing Data in Headers and Footers .................. 185

12 Customizing .................................................. 189
  12.1 User Preferences ........................................... 190
    12.1.1 General Preferences ................................... 191
    12.1.2 Section Borders Preferences .......................... 192
    12.1.3 Input Source Preferences ............................... 192
    12.1.4 Design Window Preferences ............................. 193
    12.1.5 Design Window Sections Preferences ................. 194
    12.1.6 Thumbnails Preferences ................................ 194
    12.1.7 Input (CSV) Window Preferences ....................... 195
    12.1.8 Input (JDE PDF) Window Preferences ................... 196
    12.1.9 Input (JDE PDF) Window Sections Preferences ......... 197
    12.1.10 Input (Text) Window Preferences ..................... 197
    12.1.11 Input (Text) Window Sections Preferences .......... 199
    12.1.12 Input (XML) Window Preferences ..................... 199
  12.2 Spelling and Language ..................................... 200
    12.2.1 Checking Spelling ..................................... 200
    12.2.2 Languages ............................................ 201
    12.2.3 User Dictionaries ...................................... 202
  12.3 Colors ................................................... 202
Appendices

A  Glossary  207
   A.1 Glossary .............................................................. 208

B  Quick Guide  211
   B.1 Quick Guide to Menu Commands  ... 212
   B.2 Quick Guide to Keyboard Shortcuts  ........ 218
      B.2.1 Menu Commands  .................. 218
      B.2.2 Dialog Options  ................. 219
   B.3 Quick Guide to Tutorials  ...................... 221
Introducing Create!form Transform

Create!form Transform is a design component of the Create!form™ Distributed Output Management (DOM) system, that allows you to build customized data transformations, which can be applied to text, CSV, XML and other input file formats.

- Who is this Guide For?
- Create!form Transform Features
- Design Prerequisites
- Production Prerequisites
- Getting Help
- User Guide Conventions
1.1 Who is this Guide For?

This user guide is for people who design projects with Create!form Transform.

1.2 Create!form Transform Features

With Create!form Transform you can customize your documents without making modifications in your enterprise server. The following examples introduce some of the features that Create!form Transform offers.

Text files
You can add, remove and reposition data on a page; change the page length and width; or remove pagination altogether.

Transactional data
You can convert XML transactional data, as formatted text or CSV files, into a format suitable for processing by other system components.

Database updates
You can update any database with the contents of a text or XML file, by converting the data into a CSV data file.

Database extraction and merging
You can extract the contents of any database, either by looking up the records directly in the database, or by processing an extracted CSV data file, and merge the extracted data with the contents of other documents.

1.3 Design Prerequisites

The following are required before you can begin designing your project:

- basic familiarity with Windows concepts
- basic familiarity with word-processing or publishing software
- Adobe Reader version 4 or greater for viewing the user guide.

For information on system and hardware requirements, see Section 2.1.
1.4 Production Prerequisites

Before you can commence printing you must install and configure Create!form Server on the network e-forms server. For more information see the Create!form Server user guide. Depending on your preferred output format, you can also install the Create!archive, Create!fax or Create!email output modules.

1.5 Getting Help

Help is available in both Portable Document Format (PDF) and as online help.

To view and print the PDF user guide:

• Open the file Create!form Transform 6.1.pdf with Adobe Reader from the Manuals directory on the installation CD, or
• From the Windows Start menu, select:
  Programs\Create!form\Manuals\Create!form Transform

To view the online help:

• Start Create!form Transform and select Help\Help Topics from the Main menu.

1.6 User Guide Conventions

Section headings are listed at the front of each chapter to assist you in finding the relevant information. In the electronic version of this document, clicking on these and other cross references will take you to the appropriate section. The following typographical conventions are used:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Convention Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bold</td>
<td>Screen items and buttons.</td>
</tr>
<tr>
<td>Notes</td>
<td>Short-cut or reference to the document, chapter or section described after the symbol.</td>
</tr>
<tr>
<td>Courier Font</td>
<td>Commands entered by a user.</td>
</tr>
<tr>
<td>Note</td>
<td>Additional information relating to the topic.</td>
</tr>
<tr>
<td>Important</td>
<td>Critical information relating to the use of the product.</td>
</tr>
<tr>
<td>Item\Sub-Item</td>
<td>Menu selection; i.e. select the menu Item and then select the Sub-Item from the drop-down list, e.g., File\New.</td>
</tr>
</tbody>
</table>
## Convention

<table>
<thead>
<tr>
<th>Convention</th>
<th>Convention Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numbering (1.)</td>
<td>Step-by-step procedures. To perform an action, follow these instructions in the sequential order given.</td>
</tr>
<tr>
<td>✔️ Try this...</td>
<td>Step-by-step instructions on applying the procedures described in the current topic to a specific example.</td>
</tr>
</tbody>
</table>
Installing

Create!form Transform is installed on a Windows workstation from the Create!form installation CD.

The following topics are covered:

- Installation Requirements
- Installing Create!form Transform
- Starting Create!form Transform
2.1 Installation Requirements

Create!form Transform requires the following:

- Microsoft Windows 2000/2003/XP
- Microsoft Internet Explorer 5.0 or greater
- Adobe Reader 4 or greater (for viewing the User Guide)
- approximately 30 MB of free hard disk space
- a PostScript level 2 or greater printer is recommended

2.2 Installing Create!form Transform

Create!form Transform is installed automatically when you install Create!form Director.

2.3 Starting Create!form Transform

From the Windows Start menu

- click the Start menu and select Programs>Create!form>Create!form Transform

For information on screen elements and toolbars, see Chapter 4.

For information on getting started with Create!form Transform, see Chapter 3.
Getting Started

Your design is recorded in a project, which contains information about how the input data will be mapped to the output, and the layout of text on the output page. Each project consists of several files, located in a number of folders, which are required whenever the project is opened, saved, moved or transferred.

The following topics are covered:

- What is a Project?
- Project Directories
- Creating a Project
- Design Flowcharts
- Saving a Project
- Opening a Project
- Printing and Previewing Projects
- Packing Projects
- Transferring Projects to the e-forms Server
- Managing Project Files
3.1 What is a Project?

A transform project (or project) is a record of the way you want to reformat and reorganize input data files. Because your enterprise server may output a variety of different documents, you will need to create a project for each of these variations.

A project consists of an input design, which describes the structure of the input file, and an output design, which describes the format and layout of the output file. The input design will be displayed in the Input Window, and the output design will be displayed in the Design Window.

3.2 Project Directories

The project files are stored in the following folders:

- Project Directory
- Common Project Directory

The project also uses resources from a number of system folders.

3.2.1 Project Directory

Each project has a project directory. This is the location where you save the project and where its components are stored. When Create!form Transform is installed, a default project directory is created: `<install dir>\WorkDir`. This is where you will be prompted to save all new projects. During the installation or at a later time, you can change this default project directory.

To change the default project directory:

1. From the Main menu, select Tools>Preferences.
   The Preferences dialog will be displayed.
2. From the General pane, set the Default Project directory.
3. Click OK.

It is recommended that you use separate directories for design projects and production projects. For example, use `\WorkDir` for your design work and create a new directory `\ProdDir` for completed, live projects.
3.3 Creating a Project

3.2.2 Common Project Directory

Project components that are shared between projects are stored in the common project directory. When Create!form Transform is installed a default common project directory is created: <Install dir>\CommonProject. You can specify another location for the common project directory at any time.

To change the common project directory:

1. From the Main menu, select Tools ▶ Preferences. The Preferences dialog will be displayed.
2. From the General pane, set the Common project directory.
3. Click OK.

3.3 Creating a Project

Before creating a project, you must choose a suitable input file. The input file should be representative of the files likely to be encountered in a production environment and should contain all possible variations of content and format. The elements and structures of the input file, which you identify during the input design process, are recorded in a DataMap. For information on creating a DataMap, see Chapter 5.

The New Project wizard (Section 3.3.1) will guide you through the process of creating a new project. You will be given a number of options by the wizard, which may include:

- Input file type (Section 3.3.2)
- Using an existing project as a template (Section 3.3.3)
- DataMap options (Section 3.3.4)
- Project type (Section 3.3.5)
- Output Structure Options (Section 3.3.5)

To complete your design, follow the steps described in the design flowchart (Section 3.4) relevant to the input file type you have selected.

3.3.1 New Project wizard

The New Project wizard will guide you through the process of creating a new project.

To start the New Project wizard:

1. From the Main menu, select File ▶ New, or click the New Project button ( ). The New Project Wizard dialog will be displayed.
2. From the **Type** drop-down, select the type of source file from the available file types (Section 3.3.2).

3. From the **File/Data source** drop-down, select the input source file you wish to open, from the current project directory. If the file you want is not listed, click ![Open button] to display the **Select Input File** dialog. Locate the input file from this dialog and click **Open** to select the file.

4. Click **Next**.

   The sequence and content of dialogs displayed by the wizard will vary depending on the type of input file selected in the first dialog.

5. Continue to follow the wizard instructions and click **Finish** when you have completed all options.

For descriptions of these options, refer to:

- Using an existing project as a template (Section 3.3.3)
- DataMap options (Section 3.3.4)
- Project type (Section 3.3.5)
- Output Structure Options (Section 3.3.5)

---

**Note**

The project created by the wizard will be configured according to the options you select; you can however change the project by adding or removing features during the design process.

---

### 3.3.2 Input file type

The New Project wizard will give you the option of selecting from the available input file types. The input file types supported by Create!form include text, CSV, XML and JDE-PDF. For more information, refer to Chapter 5.

### 3.3.3 Using an existing project as a template

You can use another project as a template when you create the project with the New Project wizard, or later by selecting the **Project > Apply Design Template** command. The new project will be an exact copy of the template project including the DataMap and the output design. To successfully apply a template, the input files in both projects must be the same type and have the same structure. When you apply a template to an existing project, the output design will be lost.
3.3 Creating a Project

3.3.4 DataMap options

The New Project wizard gives you the option of creating a new DataMap, copying the DataMap from another project, or using a shared DataMap.

Create a new DataMap
• Choose the Create a new DataMap option if you have not previously created a DataMap for an input file of this type and structure.

If a suitable DataMap does exist, you can choose to either copy the DataMap, or, if it has already been shared, you can share the DataMap with other projects.

Copy a DataMap from another project
• Choose the Copy a DataMap from another project option if you need to modify the DataMap in any way.

Select a shared DataMap
• Choose the Select a shared DataMap option if the DataMap can be used without modification.

For a full description of these options, refer to Section 5.3.

3.3.5 Project type

When you are working with a text input file, you can choose the type of project you want to create. The New Project wizard gives you the option to:

• create a Standard project (Section 3.3.5.1)
• create a Simple project (Section 3.3.5.2)
• create an Overlay project (Section 3.3.5.3)

3.3.5.1 Standard project

Create a standard project when the input file structure is variable or requires repagination, table sorting, sub-totalling, substantial reformatting, or summary reporting. In a standard project you can utilize Create!form Transform’s many features to map any data from the input file to the project, and then reformat and manipulate the input data in the project.

Output Structure Options

In a standard project you have the option of creating sets and tables in the project.
Set headers and footers
- Choose the Set headers and footers option if the input file contains groups of pages. For example, a print run might contain multiple invoices for the same customer. The invoices for one customer would form a set. For more information, see Section 5.5.5.

Table with headers and footers
- Choose the Table with headers and footers option if the body of each page contains repeating line items that you want to sort and subtotal in the project, or if you require floating headers and footers. For more information, see Chapter 8.

3.3.5.2 Simple project
Create a simple project if only minimal reformatting of the input data is required. In a simple project, one output page is generated for every input page, but you can still selectively map the contents of the input page and then re-position, re-order and reformat as required on the output page.

3.3.5.3 Overlay project
Create an overlay project if you don’t want to change the layout, design and formatting of the input file, and need only to add text to the output pages. In an overlay project the entire input page is mapped directly to the output page.

3.4 Design Flowcharts
The following flowcharts list the steps required to create projects for the main input file types.
3.4 Design Flowcharts

CSV

Figure 3-1: An overview of the design processes for CSV input files.
Figure 3-2: An overview of the design processes for text input files.
3.4 Design Flowcharts

XML

Figure 3-3: An overview of the design processes for XML input files.

- Define sets
  - Section 5.6.3

- Define input sections
  - Section 5.6.4

- Copy data variables to the output
  - Section 5.6.5

- Set the project properties and format styles
  - Chapter 6

- Organize sections in the project
  - Chapter 7

- Create tables for repeated data sections
  - Chapter 8

- Add and format text
  - Chapter 9
3.5 Saving a Project

You should save your work regularly. You can store projects in any directory, but Create!form Transform looks in only two directories for the files used in each project:

- the `<Install dir>`\CommonProject directory, which typically contains items common to various projects; or
3.6 Opening a Project

To view, edit, print or transfer a project, you must open the project with Create!form Transform.

To open a project:

1. From the Main menu, select File ➤ Open or click ( ) from the Main toolbar. The Open dialog will be displayed and show files from the current project directory. To display files in the \CommonProject directory, select the Look in common project directory checkbox.

2. Select the project and click Open.

3.7 Printing and Previewing Projects

Projects can be printed locally, using the data from the input file. The output can also be previewed on-screen in Notepad, or another text editor.

To print a project:

1. From the Main menu, select File ➤ Print. The Print dialog will be displayed.
2. Select the required printing options and/or view an on-screen preview.
   • Select a printer.
   • Select the page range.
   • Choose whether to show variable values or names, or whether to hide them.
   • Choose whether you want duplicate copies, or whether you want to use the project settings for duplicates (set in \textit{Section 6.1.2}).

3. Click \textbf{Print}.

Local print settings will not affect how the project behaves in production printing.

\textbf{To preview a project:}

• From the Main toolbar, click ( ) on the Main toolbar or select \textbf{File} \textbf{Print Preview} from the Main menu.

The settings made on the \textbf{Print} dialog are used when previewing the project. If the preview does not appear as expected, check the settings on the \textbf{Print} dialog.

\section*{3.8 Packing Projects}

Because projects are made up of several different files (the project file, input data file, DataMap and subforms), \textit{Create!form Transform} allows you to save all associated files into a single packed project. The entire project can then be easily moved and opened on any computer.

\textbf{To save a project as a packed project:}

1. Open the project.

2. From the Main menu, select \textbf{File} \textbf{Save As}
   The \textbf{Save As} dialog will be displayed.

3. In the \textit{Save as type} field, select \textbf{Packed project files (*.tpp6)}.

4. Click \textbf{Save}.
   The packed project is saved with the specified name, with an .tpp6 extension. To send the packed project, simply email it as an attachment.

To open a packed project, from the \textbf{Open} dialog, select \textbf{Packed project files (*.tpp6)} from the \textit{Files of type} field.
3.9  Transferring Projects to the e-forms Server

Before the project can be used for production printing, you must transfer the project and its associated resources to the e-forms server.

To transfer a project to the e-forms server:

1. From the Main menu, select **Tools**  ➤ **Transfer**
   The **Transfer to Server** dialog will be displayed.

2. Select the project to be transferred and its destination.

You will find a complete discussion on transferring projects and the options displayed on the **Transfer to Server** dialog in the Create!form Server user guide.

3.10 Managing Project Files

Create!form Transform creates and uses a number of different file types that can be identified by their icons and filename extensions:

<table>
<thead>
<tr>
<th>File Type</th>
<th>Icon</th>
<th>Extensions</th>
<th>Created By</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td><img src="image" alt="Icon" /></td>
<td>.ctp6</td>
<td><strong>File</strong> ➤ <strong>Save</strong> and <strong>File</strong> ➤ <strong>Save As</strong></td>
</tr>
<tr>
<td>Project backup</td>
<td><img src="image" alt="Icon" /></td>
<td>.ctp6bak</td>
<td>See <a href="#">Section 12.1.1</a></td>
</tr>
<tr>
<td>DataMap</td>
<td><img src="image" alt="Icon" /></td>
<td>.dmp6 (text), .cmp6 (CSV), .xmp6 (XML), .jmp6 (JDE PDF)</td>
<td><strong>File</strong> ➤ <strong>Save</strong> and <strong>File</strong> ➤ <strong>Save As</strong></td>
</tr>
<tr>
<td>DataMap backup</td>
<td><img src="image" alt="Icon" /></td>
<td>.dmp6bak etc</td>
<td>See <a href="#">Section 12.1.1</a></td>
</tr>
<tr>
<td>Packed project</td>
<td><img src="image" alt="Icon" /></td>
<td>.tpp6</td>
<td>See <a href="#">Section 3.8</a></td>
</tr>
<tr>
<td>Subform</td>
<td><img src="image" alt="Icon" /></td>
<td>.cff6</td>
<td><strong>Tools</strong> ➤ <strong>Create Subform</strong> ➤ <strong>Design New</strong></td>
</tr>
<tr>
<td>Input file</td>
<td><img src="image" alt="Icon" /></td>
<td>various</td>
<td>external source</td>
</tr>
</tbody>
</table>

The project files may be located in either the project directory or the common project directory.

Care should be taken when deleting, moving or renaming project files as this may corrupt the project. Whenever you move, copy or send a project to another location, you should pack the project first.
Create!form Transform allows you to interact with and navigate through your projects with a familiar, graphical and easy to use interface, which can be customized to suit your needs.

The following topics are covered:

- **Understanding the Screen Elements**
- **Navigating in a Project**
4.1 Understanding the Screen Elements

The Create!form Transform interface can be customized to suit your needs. The following graphic shows the basic screen elements.

Note that the appearance and position of each element may vary, depending on screen size and the current view and customization settings.

Figure 4-1: The Create!form Transform screen.
The following topics describe the function of the different windows and views in detail.

### 4.1 Understanding the Screen Elements

#### 4.1.1 Design Window

The Design Window has two viewing modes, the Graphics View and the Listing View. To switch between them, click the Graphics View ( ) or the Listing View ( ) on the Main toolbar. The sections of the project are displayed as tabs at the bottom of the Design Window.

- **The Graphics View** displays what the project will look like when you print it, and allows you to select, edit and manipulate objects. The Graphics View of the Design Window is where you will perform most of your project design tasks.

- **The Listing View** is a non-graphical representation of all objects in the selected section, along with key information about each object. The Listing View enables you to easily identify objects when the Graphics View is crowded. You can select, edit and work with all objects in the Listing View in the same way as the Graphics View.

  You can change the sort order of the Listing View by clicking on the column heading you wish to sort by. If the X-Origin or Y-Origin is displayed in red text, it indicates that part (or all) of the object is off the page layout area.

You cannot hide the Design Window, but you can customize its look and feel (Sections 12.1.4 and 12.1.5).

#### 4.1.2 Input Window

The Input Window displays the input file. When you float the cursor over data in the input file, the name of the data variable will be displayed (Text and JDE PDF input files only). This can help you to select the correct data variable when copying to the Design Window.

To show or hide the Input Window, click the Input Window ( ) on the Main toolbar or select **View > Input Window** from the Main menu. Several task shortcuts are available by right-clicking on items in the Input Window.

You can customize the look and feel of the Input Window (Section 12.1).
4.1.3 Project Tree

The Project Tree displays the logical structure of the project. It contains a list of all sections and all objects, including their properties. Several task shortcuts are available by double-clicking and right-clicking on items in the Project Tree.

To show or hide the Project Tree, click the Project Tree ( ) on the Main toolbar or select View → Project Tree from the Main menu.

4.1.4 Input Tree

The Input Tree displays the logical structure of the input file. It provides details about the input file as a whole and individual sections within it. Several task shortcuts are available by double-clicking and right-clicking on items in the Input Tree.

To show or hide the Input Tree, click the Input Tree ( ) on the Main toolbar or select View → Input Tree from the Main menu.

4.1.5 Thumbnail View

The Thumbnail View provides a thumbnail of each page of the project, allowing you to scan the formatting of multiple pages.

To show or hide the Thumbnail View, click the Thumbnails ( ) on the Main toolbar or select View → Thumbnails from the Main menu.

You can customize the look and feel of the Thumbnail View (Section 12.1.6).

4.1.6 Input File Source View

The Input File Source View displays the raw input file.

To show or hide the Input File Source View, select View → Input File Source from the Main menu.
4.1.7 Status Bar

The Status bar provides a variety of information about the project as a whole and the current location within the project.

Shortcuts are available by double-clicking on the section and background tabs.

4.2 Navigating in a Project

The section tabs on the Status bar only appear for sections that are displayed on the current page of the active window. To locate occurrences of some sections using the tabs, you may need to browse the pages of the project.
To navigate to other pages or sections of the project:

- Click the appropriate button from the Navigation toolbar.
  The action will be applied to the active window:

![The buttons on the Navigation toolbar.](image)

These commands are also available through the main menu.

By default, when you browse to a particular section in the Design or Input Window, the corresponding section in the other window will be displayed. This synchronization can be turned off and on (Section 12.1.7).

### 4.2.1 Jumping directly to a set or page

**To go to a specified page number:**

1. From the Main menu, select Page ➤ Go To.
   The Go To Page dialog will be displayed.
2. Enter the page number to go to, or select the page number from the drop-down.
3. Click OK.
   The page selected will be displayed.

**To go to a specified set number:**

1. From the Main menu, select Set ➤ Go To.
   The Go To Set dialog will be displayed.
2. Enter the set number to go to, or select the set number from the drop-down.
3. Click OK.
   The first page of the set selected will be displayed.
4.2 Navigating in a Project

4.2.2 Moving between sections

In order to edit and format sections, they must be selected. You can select sections in the following ways:

- Click on the section.
- Press the arrow keys to move up or down one section in the current window.
- Press the TAB key to move to the next occurrence of that section and SHIFT+TAB to move to the previous occurrence of that section.
- Click on the section tab on the Status bar.
  All sections that are displayed on the current page will have tabs.
- Select the section in the Project Tree and the Input Tree.
  This is the only way you can edit the properties of a section if it is not set to display.
- Use the buttons on the Navigation toolbar.
Input Design

The input design process involves identifying and labelling the different parts or elements of the input file. The way in which you identify, label and organize the data in the input file is recorded in the DataMap.

The following topics are covered:

- **Before You Start**
- **Key Input Design Concepts**
- **DataMap Types and Options**
- **Designing for CSV Input Files**
- **Designing for Text Input Files**
- **Designing for XML Input Files**
- **Designing for JDE PDF Input Files**
- **Creating Derived Variables**
5.1 Before You Start

The following topics contain information about some basic procedures, which you may find useful:

- Using the Create!form sample projects (Section 5.1.1)
- Selecting a different input file (Section 5.1.2)
- Viewing the input file source (Section 5.1.3)

Before proceeding to the details of designing for a particular file type, you must also be familiar with:

- Key Input Design Concepts (Section 5.2)
- DataMap Types and Options (Section 5.3)

5.1.1 Using the Create!form sample projects

Sample input files have been provided with the installed software so that you can practice performing the tasks described in this user guide. The samples can be found in the `<install dir>\CF6Samples\Tutorials` folder. You can view the text, CSV and XML sample files with any text editor. You should complete each task in sequence as later tasks may assume that earlier tasks have already been completed.

Tutorials are provided for selected topics and appear as:

![Try this...](image)

The exercises in these sections of the chapter will help you to apply what you have learnt, step by step, on a sample project.

You will find a complete index to tutorials in Appendix B.

5.1.2 Selecting a different input file

If the input file you chose for the input design is unsuitable, or if you want to test your input design on another sample file, you can select a new input file at any time.

To select a different input file:

1. From the Main menu, select Project > Select Input File.
   The Select Input File dialog will be displayed.
2. Navigate to the folder where the file is located and select the new input file.
3. Click Open.
The project will refresh with the new data.

5.1.3 Viewing the input file source

To view the input file source, select the View/Input File Source command from the main menu.

5.2 Key Input Design Concepts

Before you begin your design, you must be familiar with the following key concepts:

- What is a DataMap? (Section 5.2.1)
- What is a set? (Section 5.2.2)
- What is a section? (Section 5.2.3)
- What is a data variable? (Section 5.2.4)
- What is a derived variable? (Section 5.2.5)

5.2.1 What is a DataMap?

The input design process involves identifying and labelling the different parts or elements of the input file such as header data, line items and footer text. The input file may be an unstructured text file, for which you must create rules that identify each part of the file, or the file may be structured, like the CSV and XML formats, which are largely self-defining, but may still require some editing or relabelling. The way in which you identify, label and organize the data in the input file is recorded in the DataMap. You must define a DataMap for every project; however, where the input file structure is the same, you can share the same DataMap between several projects.

The information recorded in the DataMap will include:

- the properties of the input file
- how sets are defined
- how sections are defined
- how data variables are defined
- how derived variables are defined

For more information about creating, editing and sharing DataMaps, refer to Section 5.3.
5.2.2 What is a set?

Where the input file contains sequential related pages or blocks of data, such as account statements for multiple customers, you can group these pages or data together as sets.

During production printing, each set can be treated as a self-contained document. Defining sets also allows you to make use of various pre-defined system conditions, which can be used to apply different formats, or insert extra pages at the start and end of each set.

For more information about defining sets, refer to Section 5.4.3 (CSV), Section 5.5.5 (text), Section 5.6.3 (XML) and Section 5.7.2 (JDE-PDF).

5.2.3 What is a section?

A section describes a repeated group of data in the input file. For example, in a text file, the rows that contain header text at the top of each page will form a section, while in a CSV file, each row will form a section.

By describing all the different types of data in the input file in this way, the information contained in each section can be more easily reorganized and reformatted in the project.

For more information about defining sections, refer to Section 5.4.4 (CSV), Section 5.5.1 (text), Section 5.6.4 (XML) and Section 5.7.3 (JDE-PDF).

5.2.4 What is a data variable?

Input sections can be composed of several different types of information, for example, the input header sections might contain names, addresses, dates and various reference numbers. A data variable specifies a particular field of data in each section.

Input data cannot be displayed or used in the project unless the data has been identified as a data variable in the input file. The process of defining data variables will vary according to the type of input file you are processing. For more information, refer to the relevant input format type in the sections that follow.

For more information about creating data variables, refer to Section 5.4.5 (CSV), Section 5.5.6 (text), Section 5.6.5 (XML) and Section 5.7.4 (JDE-PDF).
5.2.5 What is a derived variable?

You can combine and manipulate data variables, in user defined expressions that are called derived variables. Examples of how derived variables can be used include:

• to retrieve a substring of characters from a data variable
• to concatenate two or more data variables into one string of characters
• to convert the format of dates
• to find abbreviations and substitute with the expanded text
• to derive values from an arithmetic formula using one or more data variables

Derived variables will appear as data variables in the project.

The properties of derived variables and how they are defined are described in Section 5.8.

5.3 DataMap Types and Options

You can use either an embedded DataMap, or share a DataMap with other projects, which can be selected when you create the project, or later while designing the project.

What would you like to know about?

• Embedded and shared DataMaps (Section 5.3.1)
• DataMap options when creating a new project (Section 5.3.2)
• DataMap options while designing a project (Section 5.3.3)

5.3.1 Embedded and shared DataMaps

A DataMap can be either:

• Embedded
  An embedded DataMap is associated with a single project, and can be edited and saved without reference to other projects. When you create a new DataMap, or copy a DataMap from another project, the DataMap status is set to embedded.
• Shared
  A shared DataMap can be used by multiple projects. Any changes made to a shared DataMap will affect every project that uses it.

To confirm the status and name of a DataMap:

1. From the Main menu, select the Project ➤ Change DataMap command. The Change DataMap dialog will be displayed.
2. The status of the DataMap is displayed, and
   • if the status is **shared**, the name of the shared DataMap is also shown.
   • if the status is **embedded**, the name of the DataMap can be determined by
     adding the appropriate extension to the stem of the project name e.g. if the
     input file type is CSV, the project MyProject.ctp6 creates an embedded CSV type
     DataMap with the name MyProject.cmp6. For information about file names,
     refer to [Section 3.10](#).

### 5.3.2 DataMap options when creating a new project

The New Project wizard gives you the option to:

- **Create a new DataMap**
  Choose this option if you have not previously created a DataMap for the type of
  input file you are working with.

- **Copy a DataMap from another project**
  Choose this option if you have previously created a DataMap in another project
  for the same input file, which you want to modify for the new project.

- **Select a shared DataMap**
  Choose this option if you are able to use a DataMap from another project without
  modification. You must share the DataMap ([Section 5.3.3.3](#)) before it can be
  selected.

### 5.3.3 DataMap options while designing a project

In an existing project, you can edit the DataMap or change the status of the DataMap
in several ways:

**What do you want to do?**
- Edit an embedded DataMap ([Section 5.3.3.1](#))
- Copy a DataMap from another project ([Section 5.3.3.2](#))
- Share a DataMap with other projects ([Section 5.3.3.3](#))
- Edit a shared DataMap ([Section 5.3.3.4](#))
- Use a shared DataMap ([Section 5.3.3.5](#))
- Embed a DataMap ([Section 5.3.3.6](#))
5.3 DataMap Types and Options

**5.3.3.1 Edit an embedded DataMap**
You can edit an embedded DataMap at any time. The DataMap will change whenever the input design is changed.

**5.3.3.2 Copy a DataMap from another project**
1. From the Main menu, select the **Project > Change DataMap** command. The **Change DataMap** dialog will be displayed.
2. Select the **Copy from another DataMap** option.
3. Click **Select** and locate the DataMap file you want to copy.
4. Click **OK**.
The existing DataMap will be overwritten and the status remains embedded.

**5.3.3.3 Share a DataMap with other projects**
1. From the Main menu, select the **Project > Change DataMap** command. The **Change DataMap** dialog will be displayed.
2. Select the **Share this DataMap as** option.
3. The name of the shared DataMap is displayed, and can be edited.
4. Click **OK**.
The DataMap will be moved to the \CommonProject directory and renamed. The DataMap status will be changed to shared.

**5.3.3.4 Edit a shared DataMap**
You can edit a shared DataMap from any project that uses the DataMap. A warning will be displayed when you open additional projects that share the same DataMap.

When you edit a shared DataMap, the changes will be applied to all projects that use the same DataMap. A warning will be displayed before the edited DataMap is applied to other open projects.
5.3.3.5 Use a shared DataMap

Before you can use a shared DataMap, you must first share the DataMap from the original project. For information on how to share a DataMap, refer to Section 5.3.3.3.

To use a shared DataMap:
1. From the Main menu, select the **Project** ➤ **Change DataMap** command. The **Change DataMap** dialog will be displayed.
2. Select the **Select a shared DataMap** option.
3. Click **Select** and locate the DataMap file you want to use. Shared DataMaps are stored in the \CommonProject directory.
4. Click **OK**.
The DataMap status will be changed to shared.

To select a different shared DataMap:
1. From the Main menu, select the **Project** ➤ **Change DataMap** command. The **Change DataMap** dialog will be displayed.
2. Select the **Select a different shared DataMap** option.
3. Click **Select** and locate the DataMap file you want to use. Shared DataMaps can be stored in project directories and the common project directory.
4. Click **OK**.

5.3.3.6 Embed a DataMap

In a project that uses a shared DataMap, you can embed either the shared DataMap, or a DataMap from another project.

To embed a DataMap:
1. From the Main menu, select the **Project** ➤ **Change DataMap** command. The **Change DataMap** dialog will be displayed.
2. Select one of the following options:
   - **Embed DataMap into project**
   - **Embed copy of another DataMap into project**
3. If selecting another DataMap, click Select and locate the DataMap file you want to use.

4. Click OK.

The DataMap is copied into the Project Directory and named according to the project stem and DataMap type (Section 3.10). The DataMap status will be changed to embedded.

5.4 Designing for CSV Input Files

The input design process for CSV files involves identifying the row and column structure of the CSV input file; this information is recorded in the DataMap. A DataMap for an CSV input file is denoted by the .cmp6 file name extension.

What would you like to know about?
- CSV file structure (Section 5.4.1)
- Input file properties (Section 5.4.2)
- Defining sets (Section 5.4.3)
- Defining sections (Section 5.4.4)
- Working with data variables (Section 5.4.5)

To create the CSV sample project:

1. To start the wizard, click the New Project button ( ) on the Main toolbar.
2. From the Type drop-down, select CSV from the available file types.
3. Click ( ) to display the Select Input File dialog and select the file Chapter5Sample.csv from the <install dir>\CF6Samples\Tutorials folder.
4. Click Next.
5. Select the Create a new DataMap radio button and click Next. A new DataMap will be created.
6. Ensure the Set headers and footers checkbox and the Table with headers and footers checkbox are cleared and click Next.
7. Click Finish.

The new sample project will be displayed in the Input Window.
5.4.1 CSV file structure

A CSV file contains repeated rows of data, with each field in a row separated by a particular character called the delimiter. Typically fields are separated by a comma, but other characters can be used, including spaces and tabs. The file may also contain a header section, and a column heading row. The important elements of a CSV file are demonstrated in sample comma delimited file below:

- **Header rows**
  - Extract from contacts database 9/18/05
  - **********************************************
- **Column headings**
  - Autodial,Name,Tel2,Div,Address,City,State,Zip,Country
  - 26002,A&E Job,,Division 1,345 Main Street,Birmingham,AL,65034,USA
  - 5577,"Abbot, Clarice",,Clarice Abbot,1407 West Chenango,Englewood,C
  - 5619,"Abbot, Jamie",,Jamie Abbot,1407 West Chenango,Englewood,C
  - 5532,"Abbot, Mark",,Mark Abbot,1407 West Chenango,Englewood,CO
  - 6002,"Abbott, Dominique",,Dominique Abbott,1407 West Chenango,Er
  - 1538,Abernathy & Hitch,212 234-1887,Abernathy & Hitch,5907 Fifth A
  - 7392,Accounts Receivable Department,,Accounts Receivable Depart
  - 17002,Acme Department Store #68,,Acme Department Store #68,1790

**Figure 5-1:** CSV input file structure

Note that text values that contain the delimiter character are enclosed by quotation marks, which is referred to as the text qualifier.

**Paginated CSV Files**

If the CSV input file is paginated and contains page header and footer sections, you must treat the input as a text file, in which you will be able to define the page length and separately identify the detail sections from the header and footer sections. In a text input file, you can create a data variable that contains the entire data row, and then use the `retrieve` function to extract the delimited field values as user variables. For more information, refer to Section 5.5.

5.4.2 Input file properties

The input file properties determine how the structure of the CSV input file will be interpreted.

**To change the input file properties:**

1. Select the Input Window by clicking anywhere inside the input window frame.
2. From the Main menu, select **Input ➤ File Properties**. The **Input File Properties** dialog will be displayed.

![CSV Input File Properties dialog]

**Figure 5-2: CSV Input File Properties dialog**

3. In the **Delimiter** section, select a delimiter character and a text qualifier.

4. In the **Header** section, select the number of header rows and indicate whether the file contains a column heading row.

5. In the **Data format** section, select the format used for date and decimal values in the input file.

6. Click **OK**.

The changes will be applied to the input file and the input window will be refreshed.
5.4.3 Defining sets

You can use the following means to determine when a new set starts:

- the entire input file can be treated as one set (default setting)
- when the value in one or more columns changes
- or, one set per row

Try this...

Follow the instructions in Section 5.4 and create a new sample CSV input project. In the sample project, the wizard applies the default input file property settings, and the input window should appear as shown below:

If you need to change how the input file is interpreted:
1. Click anywhere inside the Input Window.
2. From the Main menu, select Input→File Properties. The Input File Properties dialog will be displayed.
3. In the Delimiter section, select the comma (,) as the field delimiter, and the double quotation mark (“”) as the text qualifier.
4. In the Header section, set the number of header Lines to 2, and select the Column headings checkbox.
5. Click OK.
5.4 Designing for CSV Input Files

To define sets:

1. From the Main menu, select Set>Define. The Define Set dialog will be displayed.

2. Select one of the radio button options to determine when a new set starts:
   • Treat entire spool file as one set.
   • When values change in marked columns:
     Select one or more columns.
   • One set per row

3. Click OK.

5.4.4 Defining sections

Sections are created automatically from the structure defined by the Input File Properties dialog (Section 5.4.2):

• InputFileHeader
  The optional section at the top of the file that contains header text
• Detail
  The repeating rows in the body of the file that contain the delimited data

The sections created are displayed in the Input Window.
5.4.5 Working with data variables

Data variables are created automatically from the structure defined by the Input File Properties dialog (Section 5.4.2). The data variables created are shown in the Input Tree view:

![Input Tree](image)

Figure 5-3: Data variables created from the sample CSV input file

What would you like to know about?

- Data variables in the input header section (Section 5.4.5.1)
- Data variables in detail sections (Section 5.4.5.2)
- Changing the properties of columns (Section 5.4.5.3)
- Changing the input data format (Section 5.4.5.4)
- Copying data variables to the project (Section 5.4.5.5)

5.4.5.1 Data variables in the input header section

In the input header section, a data variable will be created for each header line.

5.4.5.2 Data variables in detail sections

In detail sections, a data variable will be created for each column. The column heading will be used as the data variable name.

5.4.5.3 Changing the properties of columns

You can change the name, type and other properties of columns that define data variables.
To change the properties of a column:

1. In the Input Window, click the column heading you wish to edit. The Column Properties dialog will be displayed.
2. To change the column heading and the name of the data variable, type a new name in the Name field.
3. In the Type field, select text, numeric or date as the data type.
4. To repeat the last non-blank value in blank fields, select the Repeat previous value if blank checkbox.
5. Click OK.

5.4.5.4 Changing the input data format

Create!form Transform interprets dates and numeric data in the input file using the data formats defined by the input file properties. For example, you can change the decimal separator to a comma, or the date order from year-month-day to month-day-year.

To change the input data format:

1. From the Main menu, select Input→File Properties. The Input File Properties dialog will be displayed.
2. From the Date order drop-down, select the required date format.
3. From the Decimal symbol drop-down, select the required numeric format.
4. Click OK.

5.4.5.5 Copying data variables to the project

Data variables defined in the Input Window can be displayed in the project by dragging the data variable into the Design Window.

To copy a data variable to the project:

1. In the Input Window, select the data variable.
2. Drag-and-drop the selected variable from the Input Window across into the Design Window.

For more information on adding text variables and other objects to the project, see Section 9.2.
5.5 Designing for Text Input Files

The input design process for text files involves identifying and labelling the different parts or sections of the input file such as header text, line items and footer text. You must create rules that identify each of these parts of the input. These rules are recorded in the DataMap. A DataMap for a text input file is denoted by the .dmp6 file name extension.

The information recorded in the DataMap includes:

- classification of rows on each page into input sections
- input file page properties
- grouping of pages as sets
- mapping of the contents of input sections as data variables
- definition of derived variables

What would you like to know about?

- Input sections (Section 5.5.1)
- Examining how sections are populated (Section 5.5.2)
- Design principles (Section 5.5.3)
- Setting the input page size (Section 5.5.4)
- Defining sets (Section 5.5.5)
- Working with data variables (Section 5.5.6)
- Common input section tasks (Section 5.5.7)
- Working with input header sections (Section 5.5.9)
- Working with detail sections (Section 5.5.10)
- Working with input footer sections (Section 5.5.11)

To create the text sample project:

1. To start the wizard, click the New Project button ( ) on the Main toolbar.
2. From the Type drop-down, select Text from the available file types.

Try this...

In the sample project, drag-and-drop the first cell of each column into the output.
Your input design is now complete.
If you haven’t already done so, save your project.
5.5 Designing for Text Input Files

3. Click ( ) to display the Select Input File dialog and select the file Chapter5Sample.txt from the <install dir>\CF6Samples\Tutorials folder.

4. Select the Create a New DataMap radio button and click Next. A new DataMap will be created.

5. Click Next.

6. Select the Standard radio button and click Next. This option will permit you to fully define the structure of the input file.

7. Ensure the Set headers and footers checkbox is selected, and the Table with headers and footers checkbox is cleared and click Next. Sets will be automatically created in the project. No tables will be created.

8. Click Finish. The new sample project will be displayed in the Input Window.

Figure 5-1: Input Window showing the sample project input file.

For more information about screen elements in the Input Window, see Section 4.1.

5.5.1 Input sections

There are four types of sections used to define the structure of the input file:

- **Input file header section (Section 5.5.8)**
  A non-repeating section at the start of the file that contains header text.

- **Input header section (Section 5.5.9)**
  A section at the top of each page that contains header text.
Chapter 5: Input Design

- **Detail section** (Section 5.5.10)
  Used to define repeating rows within the body of the page.

- **Input footer section** (Section 5.5.11)
  A section at the bottom of each page that contains footer text.

The following graphic displays the way you would define the sections of the sample file that are repeated on each page of the input file. The input file header section occurs once, at the start of the file.

![Figure 5-2: Input page structure](image)

Each section in the input file has a different set of properties, as described in Section 5.5.7.1.

**Try this...**

Follow the instructions in Section 5.5 and create a new sample text input project. In the sample project, the wizard creates input header, detail and input footer sections:

1. In the Input Window, click inside the first five rows to select the input header section created by the wizard.
2. Click below the input header section to select one of the repeated detail sections in the body of the file.
3. Scroll down to the bottom of the page and click anywhere inside the last five rows of the page to select the input footer section.

Note how the active section border and section tabs on the Status Bar change to indicate the selected section.
5.5.2 Examining how sections are populated

To assist you in the design process, you can examine all the instances of a section in the input file with the **Input> Show All Instances** command. In this view mode, all non-blank characters in all instances of the selected section will be marked with an “X”.

![Figure 5-3: Examining all instances of a section in the input file](image)

Populated column positions are marked with an X

5.5.3 Design principles

Create!form Designer can handle a variety of input file structures, from simple tables to complex and irregular forms. This section introduces the design principles that form the foundation for the procedures described later in this chapter.

5.5.3.1 Designing from the top-down

The most important principle, when preparing the input file, is to approach the design from the top-down. As depicted in the Design flowchart, start with the input header section on the first page and make any adjustments needed to the section before looking at the next section. Scan the page from the top-down and locate the first anomaly. Resolve the anomaly, then move down the page to the next anomaly. When the design is correct on the first page, view the second page, again from the top-down. Proceed through the file until the design is complete from start to finish — from top to bottom.
5.5.3.2 Design flowchart
The following flowchart describes the main tasks associated with input design.

![Input design flowchart](image)

5.5.4 Setting the input page size
Create!form Designer can handle fixed or variable page lengths, and widths of up to 2048 characters. You can define the input file page length in the following ways:

- by a form feed character (default setting)
- the page has a fixed number of lines
- by a page number
- by a specified character
The input page size is set from the **Input File Properties** dialog:

---

**Figure 5-5:** Input page size options on the Input File Properties dialog.

---

**To define the input file page width and length:**

1. From the Main menu, select **Input** > **File Properties**. The **Input File Properties** dialog will be displayed.

2. In the **Lines** field, type the page length in lines.

3. In the **Columns** field, type the line width in characters. If you are unsure of the page length or line width, click the **Recalculate** button to scan the input file and find the maximum page length and line width in the file. Refer to **Section 5.5.4.1** for details of how the page size is calculated.

4. In the **End of page** section, select one of the following radio button options:
   - **Form feed**
     A new page will be generated whenever a form feed character is encountered.
   - **Fixed number of lines**
     A new page will be generated according to the **Lines** field above.
• **Characters**
  A new page will be generated when the specified characters or character string is encountered.
  In the **Characters** field, type the character or character string.
  If the string is in a specific location, type the starting position of the string in the **at column** field; all other occurrences of the string will be ignored.
  Select the appropriate radio button to determine how the character string will be applied:
  select **define the first line of a page** if the new page begins at the start of the line containing the string; 
  select **define the last line of the page** if the new page begins at the start of the next line following the line containing the string; 
  select **terminate the page** if the new page begins at the next character following the string.

  5. Click **OK**.

  ![Try this...](image)

  In the sample project,
  1. From the Main menu, select **Input ➤ File Properties**.
     The **Input File Properties** dialog will be displayed.
     By default, the **Form feed** option is selected to control page length. Using a text editor like Notepad, open the input file **Chapter5Sample.txt** and verify that a form feed character occurs at the end of each page. Count the rows between each form feed.
  2. Return to Create!form Transform, and click the **Recalculate** button on the **Input File Properties** dialog to determine the maximum page width and length in the sample file.
  3. Click **OK**.

  In this example, the maximum page length is 27 lines. If you use a different page length, the input footer section will always ‘sink’ to the bottom of the page.

  **5.5.4.1 How the page size is calculated**

Once Create!form designer has calculated the page size for a project, it remains unchanged until the **Page size** fields are edited, or the **Recalculate** button is selected.

**When creating standard and simple projects**

When creating a new standard or simple project, Create!form designer will read the first 10 pages of the file to determine the maximum lines and columns per page. If the number of lines is less than the current default value, the default will be used. If the number of columns is less than the current default value, the default will be used.
5.5 Designing for Text Input Files

When creating an overlay project

When creating a new overlay project, Create!form designer will read the first 10 pages of the file to determine the maximum lines and columns per page. The calculated page size is always used and the default values are ignored.

When recalculating the page size

When you click the Recalculate button, Create!form designer will read the first 10 pages of the file to determine the maximum lines and columns per page. The calculated page size is displayed and the default values are ignored.

To change the default number of lines and columns

1. In the Lines and Columns fields, type the new default values.
2. Review the other settings in the Input File Properties dialog and set default values as appropriate.
3. Click the Set as Default button.

The default settings may be applied when a new standard or simple project is created (see above) or whenever the Reset from Default button is selected.

5.5.4.2 Other input file settings

In addition to page size settings, the following properties are also set from the Input File Properties dialog:

- **Data format**
  Sets the formats used by data variables (Section 5.5.6.1).
- **Ignore leading blank pages**
  Select to ignore blank pages at the start of the input file.
- **File starts with document header**
  Creates an input file header section (Section 5.5.8).
• **Ignore bold and underline formatting**
  When processing OS/400 spool files (typically in overlay projects), if the spool file contains bold and underline overstrike control characters, you can choose to either ignore overstriking or convert the overstrike characters into PostScript. Previously this option could only be set in the writer properties on the server.

![Input File Properties dialog](image)

**Figure 5-6: The Input File Properties dialog.**

**To define input file properties:**

1. From the Main menu, select **Input ➤ File Properties**. The **Input File Properties** dialog will be displayed.
2. Select options as required.
3. Click **OK**.
5.5.5 Defining sets

Defining sets involves specifying when a new set will commence. Typically a new set commences every time the page numbering restarts, but you can also specify that a new set commences when data such as the customer number or invoice number changes.

You can use the following means to determine when a new set starts:

- at the start of each page in the input file (default setting)
- when the page number in the input file is equal to 1
- when the value of a specified character string on the input page changes
- when a specified condition tests TRUE
- or, the entire input file can be treated as one set

To define sets:

1. From the Main menu, select Set>Define.
   The Define Set dialog will be displayed.

2. Select one of the radio button options to determine when a new set starts:
   - One set per input page
   - When page number = 1 at:
     In the Input Window, click-and-drag a box around the page number. The character position and length of the page number field will be displayed in the Define Set dialog.
   - When value changes at:
     In the Input Window, click-and-drag a box around the value. The character position and length of the value field will be displayed in the Define Set dialog.
   - On condition:
     Click the Edit button to display the Condition Builder dialog (Section 11.2). A new set will start when the condition tests TRUE.
   - Treat entire spool file as one set.

3. Click OK.
5.5.6 Working with data variables

5.5.6.1 Set input data formats

Before you start creating data variables, you must set the data formats that Create!form Designer will use to recognize dates and numeric data in the input file. For example, you can change the decimal separator to a comma, or the date order from year-month-day to month-day-year.

To change the input data format:

1. From the Main menu, select Set » Define. The Define Set dialog will be displayed (Figure 5-5).
2. From the Date order drop-down, select the required date format.
3. From the Decimal symbol drop-down, select the required numeric format.
4. Click OK.

5.5.6.2 Create data variables

To selectively map particular data from the input, for use in the project, you must create data variables around the data.
To create a data variable:

1. In the Input Window, select the section containing the data you want.
2. Click-and-drag a box around the area that contains the information you want to store in the variable.
   The Data Variable Properties dialog will be displayed.

3. In the Name field, type a name for the data variable.
   The variable name must start with an alphabetic character and then be followed by any combination of alphabetic (a-z, A-Z) and numeric (0-9) characters and underscores. Names are not case sensitive. Naming of variables is entirely at the discretion of the user; however the convention of naming variables to represent the data they contain is recommended. You should think of a name that you will be able to recognize and distinguish from all other data variables in the section.

4. Select a data type from the Type drop-down list.
5. Click OK.
   The area defining the variable will appear highlighted.

Use the Show All Instances command (Section 5.5.2) to examine the extents of a data variable in all instances of the section in the input file.

To change the size and position of a data variable:

1. In the Input Window, select the data variable.
2. Click-and-drag the resize handles to change the box size, or click-and-drag inside the box to reposition the box.

For more information on editing graphic objects, see Section 9.5.
To display and edit data variable properties:

1. Display the **Data Variable Properties** dialog by doing one of the following:
   - Double-click the data variable in either the Input Window or the Input Tree.
   - Right-click the data variable in either the Input Window or the Input Tree and select **Properties** from the shortcut menu.

![Data Variable Properties dialog](image)

*Figure 5-8: The Data Variables Properties dialog.*

2. With the **Data Variable Properties** dialog you can:
   - Change the name of the data variable by typing a new name in the **Name** field.
   - Select the data type from the **Type** drop-down. See also [Section 5.5.6.1](#) for information on how Create!form Designer recognizes date and numeric formats.
   - Change the position and size of the data variable in the **Row**, **Column**, **Depth** and **Length** fields. Row and column positions are defined relative to the origin (top left corner) of the selected section.
   - Select the **Trim spaces** checkbox to remove leading and trailing spaces from the text string.
   - Select the **Expandable** checkbox to make the data variable expand with the section ([Section 5.5.6.5](#)).

To delete a data variable:

1. Select the data variable in the Input Window or Input Tree.
2. Press the DEL key.
5.5 Designing for Text Input Files

Try this...

In the sample project,
1. Double-click the input header section to display the **Input Section Properties** dialog.
2. Type 9 (rows) in the **Size** field.
3. Click **OK**.
   This will increase the size of the input header section, allowing you to identify all the data variables it contains.
4. Create the data variables in the input header section as shown in the graphic below

The variables in this example are used to define data values in the input header section of each invoice:

- **BillTo** — text type variable containing the billing address
- **OrderNo** — text type variable containing the shipping order number
- **InvNo** — text type variable containing the invoice number
- **InvDate** — date type variable containing the invoice date
- **PageInfo** — text type variable containing the information about page numbers
- **CustRef** — text type variable containing the customer number

The names used above have no special significance, for example, “PageNo” could be changed to “PgNum” or perhaps “Page_Number”.

In the figure above, observe how some of the data variables have been defined larger than the data appearing in the input header section. This anticipates where the data can expand in size in other occurrences of the section, for example, the PageInfo data variable could be “Page 12 of 32”. You can use the **Trim spaces** option (see below) to remove any extra spaces.

You can display the name of a data variable by moving the cursor over the data variable box. Data variables are also listed in the Input Tree (**Section 4.1.4**).

5.5.6.3 Copy data variables to the project

Data variables defined in the Input Window can be displayed in the project by dragging the data variable into the Design Window.
To copy a data variable to the project:

1. In the Input Window, select the data variable.
2. Drag-and-drop the selected variable from the Input Window across into the Design Window.

For more information on adding text to the project, see Section 9.2.1.

Try this...

In the sample project, select the data variables you have defined in the input header section (Section 5.5.6.2) and drag-and-drop into the project set header.

5.5.6.4 Create data variables in a detail section

Any data variables created in one occurrence of the detail section will be applied automatically to all other occurrences of the same detail section.

For information on creating data variables, see Section 5.5.6.2.

The area defining the variable will appear highlighted and will be displayed in each occurrence of the same detail section:

Use the Show All Instances command (Section 5.5.2) to examine the extents of a data variable in all instances of the section in the input file.
5.5 Designing for Text Input Files

5.5.6.5 Make data variables expandable

Data variables defined in expandable sections (Section 5.5.9.2 and 5.5.10.3) can be made to expand as the section expands. For example, you could define an expandable data variable on the first row of an expandable detail section so that when the section expands to two or more rows, the data variable will contain the extra rows. An expandable data variable will always expand to the bottom of the section.

Try this...

In the sample project, create the data variables in the detail section as shown:

- **ProdNo** — text type variable containing the product number
- **ProdDesc** — text type variable containing the product description
- **UnitPrice** — numeric type variable containing the unit price
- **Qty** — numeric type variable containing the quantity
- **TotalPrice** — numeric type variable containing the total price

The variables in this example are used to define the following values in each line item:

- **ProdNo** — text type variable containing the product number
- **ProdDesc** — text type variable containing the product description
- **UnitPrice** — numeric type variable containing the unit price
- **Qty** — numeric type variable containing the quantity
- **TotalPrice** — numeric type variable containing the total price

![Figure 5-10: Creating a data variable in a detail section.](image)
To define an expandable data variable:

1. Create a data variable (Section 5.5.6.2) in an unexpanded occurrence of an expandable input section. The Data Variable Properties dialog will be displayed.

2. Type the data variable name in the Name field, and edit other properties as required.

3. Select the Expandable checkbox.

4. Click OK.

5.5.7 Common input section tasks

The following tasks are applicable to all input sections (header, detail and footer).

What do you want to know about?

• Display and edit input section properties (Section 5.5.7.1)
• Controlling when an input section is generated (Section 5.5.7.2)
• Working with multiple input sections (Section 5.5.7.3)
• Deleting input sections (Section 5.5.7.4)

For tasks specific to each section type, refer to:

• Working with the input file header section (Section 5.5.8)
• Working with input header sections (Section 5.5.9)
• Working with detail sections (Section 5.5.10)
• Working with input footer sections (Section 5.5.11)

5.5.7.1 Display and edit input section properties

Input section properties can be displayed so that you can:

• change the name of the section
• change the size of the section
• control where this type of section is applied in the input file
• control when this type of section is generated
• define which sections can follow this section
• control the order in which following sections are tested

When you change the properties of a section, the changes are applied to all occurrences of that section in the file.
5.5 Designing for Text Input Files

To display the Input Section Properties dialog:

Do one of the following:

- With the section active, select **Input > Section Properties** from the Main menu.
- Double-click the selected section.
- Right-click the selected section and select **Section Properties** from the shortcut menu.
- Double-click the section tab on the Status bar.

Input Section Properties dialog controls:

For examples of how to use conditions, see Section 5.5.9.3, 5.5.10.2 and 5.5.11.2.

For information on creating expandable sections, see Section 5.5.9.2 and 5.5.10.3.

For information on using the Condition Builder, see Chapter 11.

For information on controlling when a section is generated, see Section 5.5.7.2.

Use these controls to add or remove sections from the Followed By list.

Use these controls to change the order in which sections are tested in the Followed By list.

Figure 5-11: The Input Section Properties dialog.
5.5.7.2 Controlling when an input section is generated

You can use the Generate Section options in the Input Section Properties dialog (Figure 5-11) to control when an input section is generated in the project. If the section is not generated, the position on the input page where the section would have been displayed is reserved. Changing the Generate Section options will not alter the sequence in which sections are applied to the input page.

To control when an input section is generated:

1. In the Input Window, double-click the section. The Input Section Properties dialog will be displayed.
2. Select one of the radio buttons in the Generate Section field:
   - Always — the section will always be generated if valid.
   - Never — the section will never be generated even if valid.
   - If not empty — the section will only be generated if it contains printable characters.

5.5.7.3 Working with multiple input sections

In cases where input pages begin with different header text, or end with different footer text, or contain different line items, you can define and use additional input sections that accommodate these anomalies. Note that:

- All input header sections are listed in the input headers control list (Section 5.5.9.3).
- All detail sections are listed in the Followed By list on the relevant Input Section Properties dialog.
- All input footer sections are listed in the input footers control list (Section 5.5.10.2).
5.5 Designing for Text Input Files

For each page, the logic of which section appears where, is summarized in the following flowchart:

Figure 5-12: This logic must be applied to each page in the input.

5.5.7.4 Deleting input sections
You can delete the selected input section with the Input ▶ Delete Section command. If the section is not visible on the current page, select the section in the Input Tree. You should exercise great care when deleting an input section, as this can disrupt the logic and continuity of how other sections are applied.

5.5.8 Working with the input file header section
You can create an input file header section to contain non-repeating data at the start of the input file.

To create an input file header section:
1. From the Main menu, select Input ▶ File Properties. The Input File Properties dialog will be displayed.
2. Select File starts with document header checkbox.
3. Click OK.
To adjust the input file header section size:

1. Select the input file header section.
2. Click-and-drag the resize handle on the bottom of the section to reduce or increase the section size.

You can also double-click the section in the Input Window and set the number of rows from the Input Section Properties dialog.

5.5.9 Working with input header sections

What do you want to do?

• Adjust the input header section size (Section 5.5.9.1)
• Make an input header section expandable (Section 5.5.9.2)
• Add input header sections (Section 5.5.9.3)

What do you want to know about?

• Rules for creating multiple input header sections (Section 5.5.9.4)

5.5.9.1 Adjust the input header section size

Typically when you start a new project, the default size of the input header section will be too big or too small for the header text in the input file. You must adjust the input header section so that it contains all the header text and only the header text.

Use the Show All Instances command (Section 5.5.2) to examine all instances of the section in the input file.

To adjust the input header section size:

1. Select the input header section.
2. Click-and-drag the resize handle on the bottom of the section to reduce or increase the section size.
5.5.9.2  Make an input header section expandable

By default, input header sections have a set size. That is, all instances of that section will have the same size regardless of their content. Where the size of the text in an input header section varies between instances, you can make the section dynamically expand around the relevant text. The section becomes “expandable”.

To make an input header section expandable:

1. In the Input Window, double-click the input header section. The Input Section Properties dialog will be displayed.
2. In the Size field, type the minimum size of the section in rows.
3. Select the Expandable checkbox.
4. Click OK. The section size will be reset to the minimum size.
5. Double-click the first detail section below the input header section. The Input Section Properties dialog will be displayed.
6. Click the ( ) button next to the Condition field. Use the Condition Builder (Chapter 11) to create a valid condition, using the identified variable, for this detail section.
7. Click OK.
8. Ensure that all sections that are permitted to follow the header section, contain a suitable condition that will distinguish header rows from detail rows.

The sections that are permitted to follow the input header section are listed in the This section can be followed by field on the input header Input Section Properties dialog — also referred to as the Followed By list. The input header section will not expand if any of the sections in the Followed By list do not contain a condition, or the condition always tests TRUE.
You can also make data variables expandable in an expandable section (Section 5.5.6.5).

Try this...

In the sample project, click anywhere inside the Input Window, then navigate to the second page of the input file. Observe that the input header contains 9 rows on the first page, and 6 rows on the second page; because of this, the mapping of the data variables is now incorrect. This is typical of a continuation header. Observe also that the invoice numbers on both pages are the same; they belong to the same set.

To make the input header section expandable, firstly set the minimum size:

1. In the Input Window, on page 2, adjust the input header section size to 6 rows.

Next, create a data variable that can be used to distinguish header rows from detail rows (if you have already created the TotalPrice data variable, you can skip to step 6):

2. Select the next detail section.

3. Create a data variable around the price field at the end of the row. Note that the header rows do not contain a decimal point in this position. There are many other possible tests, but we’ll use the decimal point.

   The Data Variable Properties dialog will be displayed.

4. Type TotalPrice in the Name field.

5. Click OK.

Next, create a condition in the detail section that tests for a decimal point:

6. Double-click the detail section, making sure that you select the section clear of any defined data variables.

   The Input Section Properties dialog will be displayed.

7. Click the ( ) button and use the Condition Builder to create the expression contains(TotalPrice,".") in the Condition field. This condition will be TRUE when the data variable TotalPrice contains the "." character.

8. Click OK.

Finally, make the input header section expandable:

9. Double-click the input header section to display the Input Section Properties dialog.

10. Select the Expandable checkbox.

11. Click OK.

Now navigate back to page 1 and observe that the input header section has expanded to 9 rows, but is still 6 rows on page 2.
5.5 Designing for Text Input Files

5.5.9.3 Add input header sections
In cases where the content of the input header section changes between pages, or an expandable header section cannot be used, you can add additional header sections. Only one input header section can be used on a page.

To create an additional input header section:

1. From the Main menu, select Input \(\Rightarrow\) Input Headers.
The Input Headers Control List dialog will be displayed.

2. Click the Add Row button (\(\frac{3}{4}\)).
The New Header Section dialog will be displayed.

3. In the Name field, type the new input header section name.

4. In the Size field, type the number of rows required.

5. Using the controls provided, add one or more detail sections to the This section can be followed by list.
If there are no sections in the Followed By list, or if the sections listed are not valid when tested, no detail sections will be created on the page.

6. Click OK to close the New Header Section dialog. Do not close the Input Headers Control List dialog.

   The input headers control list will now have two input header sections defined. Both input header sections are unconditional (the Condition field is empty on the Input Section Properties dialog). If the first input header in the list is unconditional, it will always be created. So the first input header must be conditional. If the condition tests TRUE then the first input header is applied, if the test fails then the second input header is applied.

7. Click the (\(\ldots\)) button next to the first input header in the control list.
The Input Section Properties dialog will be displayed.

8. Click the Edit button next to the Condition field.
Use the Condition Builder (Chapter 11) to create a valid condition for this input header section. If you wish to use a data variable in the condition, you must create the data variable first.

9. Check that the detail sections in the This section can be followed by list are valid and in the correct order.

10. Click OK to close the Input Section Properties dialog.
11. Click the button on the second (unconditional) input header to display the Input Section Properties dialog.

12. Check that the detail sections in the This section can be followed by list are valid and in the correct order.

13. Click OK to close the Input Section Properties dialog.

14. Click OK to close the Input Headers Control List dialog.

The definition of the new input header section will be applied to the entire file.

The Condition Builder makes available a number of useful system variables and conditions, for example the condition Sys.FirstPageOfDoc will test TRUE on the first page of the document.

To delete an input header section:

Care should be taken whenever a input header section is deleted as this will alter not only which input header section is applied, but may also change the way detail sections are applied to the rest of the page. Always review the Followed By lists on the other input header sections, and the order of the input headers control list, before deleting a input header section.

1. From the Main menu, select Input→Input Headers.

   The Input Headers Control List dialog will be displayed.

2. Select the input header section in the control list.

3. Click the Delete Row button ( ).

4. Confirm that you wish to delete this section.

5. Click OK.

   The change will be applied to the entire file.

5.5.9.4 Rules for creating multiple input header sections

The procedures detailed in Section 5.5.9.3 can be applied when adding further input header sections. The rules applying to multiple input header sections are:

• The last header in the input headers control list should be unconditional.
• All other headers in the list should be conditional.
• The first header that tests TRUE in the list is applied.
• If none of the headers in the control list test `TRUE` on a page, then no header will be created on that page.
• Input header sections can be expandable (Section 5.5.9.2).

The detail section that follows the input header section, will be the first valid detail section in the Followed By list of the input header section.

5.5.10 Working with detail sections

What do you want to do?
• Add detail sections
• Make a detail section expandable

What do you want to know about?
• About defining detail sections
• Rules for creating multiple detail sections

5.5.10.1 About defining detail sections

You can create additional detail sections when the content of line items varies within the page. For example, an invoice may contain subtotal lines, blank lines or description lines in addition to the regular line items. Three different detail section types are shown in the following example.

<table>
<thead>
<tr>
<th>type</th>
<th>APT103 Bamboo Blinds</th>
<th>1020.00</th>
<th>10</th>
<th>10200.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>AT1C39 Umbrellas</td>
<td>30.00</td>
<td>80</td>
<td>2400.00</td>
</tr>
<tr>
<td>type</td>
<td>*** 9 units shipped, balance with next order</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>APT103 Black Dragon Vase</td>
<td>199.00</td>
<td>10</td>
<td>1990.00</td>
</tr>
<tr>
<td>type</td>
<td>*** hold for collection at Nunawading warehouse</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>Subtotal</td>
<td>12590.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>APT103 Tea Chests</td>
<td>799.00</td>
<td>4</td>
<td>3196.00</td>
</tr>
<tr>
<td>type</td>
<td>APT103 Black Dragon Vase</td>
<td>199.00</td>
<td>10</td>
<td>1990.00</td>
</tr>
<tr>
<td>type</td>
<td>*** *** replacement of returned items *** ***</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>AT1C78 Cane Baskets</td>
<td>50.00</td>
<td>5</td>
<td>250.00</td>
</tr>
<tr>
<td>type</td>
<td>AT1C38 Black Dragon Set</td>
<td>1600.00</td>
<td>1</td>
<td>1600.00</td>
</tr>
<tr>
<td>type</td>
<td>Subtotal</td>
<td>7036.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 5-13: An example of irregular line items.*
The details of lines are as follows:

- type 1 is the primary line item containing the item details, quantity and price
- type 2 is a descriptive line relating to the previous type 1 line
- type 3 is a subtotal line

In particular note that:

- type 1 lines are not always followed by type 2 lines
- type 2 lines can only follow type 1 lines
- type 2 lines always contain “***” at the start of each line
- type 2 can be one or more lines
- type 3 is always two lines
- type 3 lines can follow either type 1 or 2 lines
- type 3 lines can occur at any position
- type 3 lines always contain the text string “Subtotal”

These observations form a set of rules that define the input file structure. The process of building a DataMap involves translating these observations or rules into section properties. You must create a new section for each of the line types you observe in the file.

*NOTE*

Use the Show All Instances command (Section 5.5.2) to examine all instances of the section in the input file.

### 5.5.10.2 Add detail sections

There are many possible ways of creating DataMaps for the same input file. The type of sections defined will depend not only on the content of the input file, but also on how the input data will be used in the project.

**To define additional detail sections:**

1. Identify the first instance where a new section is required.
2. Select the section immediately before (above) where the new section is required.
3. From the Main menu, select Input→New Section After. The New Data Section dialog will be displayed.
4. In the Name field, type the new section name.
5. In the Size field, type the size of the section in rows.
6. Click **OK**.
The new section will be displayed in the Input Window.

7. Select the new section.

8. Create the data variables ([Section 5.5.6.4](#)) required to define the data in the new section.

9. Create any additional data variables that are required to assist in distinguishing this section from all other detail sections. You only need to create additional data variables if the data variables created in the previous step are not suitable.
For example, you might test when a particular character position is blank, or when a character string contains a particular substring, or a combination of tests using boolean operators ([Section 11.3.1](#)).

10. From the Main menu, select **Input Section Properties**.
The **Input Section Properties** dialog will be displayed.

11. Click the **Edit** button to display the **Condition Builder** dialog and create a condition that tests **TRUE** when this section is valid.
For information on creating conditions, see [Chapter 11](#).

12. Define the sections that can follow this section in the **This section can be followed by** field by adding, removing and ordering sections with the controls provided ([Section 5.5.7.1](#)).
The order in which the sections appear in this list is very important: the first valid section in this list will be the next section created after this section.

13. Click **OK**.
The definition of the new section will be applied to the entire file and displayed in the Input Window.

---

**NOTE**

It is good design practice to create a positive condition for every detail section you create, even if the condition is not immediately required to identify the section.
Try this...

In the sample project, navigate to page 2 of the input file.

1. Select the first detail section below the input header section and, if you have not already done so, create data variables to define the contents of the row (Section 5.5.6.4).

<table>
<thead>
<tr>
<th>125-79</th>
<th>78944-98</th>
</tr>
</thead>
<tbody>
<tr>
<td>A73C8</td>
<td>1634.00</td>
</tr>
<tr>
<td></td>
<td>41</td>
</tr>
<tr>
<td>71D12</td>
<td>3244.00</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>71C32</td>
<td>12376.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>A71C78</td>
<td>450.00</td>
</tr>
<tr>
<td></td>
<td>15</td>
</tr>
<tr>
<td>9 units shipped, balance with next order</td>
<td></td>
</tr>
<tr>
<td>A71C78</td>
<td>250.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

   In Section 5.5.9.2 we set a condition for the Detail section such that the TotalPrice data variable must contain a decimal point. Observe that the condition fails at row 9, where a different type of line item is encountered. A new detail section must be created for this type of line.

2. Select row 8, the row immediately above where the new section will be created.

3. Right-click inside the section border, but away from any of the data variables, and select New Section After from the shortcut menu.
   The New Data Section dialog will be displayed.

4. Define a section named Comment, of one row in size, which can only be followed by a Detail section, and click OK.

   Now create a data variable in the Comment section that you can use to positively test the type of section:

5. Create a data variable named CommentVar1 in columns 2-7, and another named CommentVar2 in columns 9-53.

6. Double-click the Comment section to display the Input Section Properties dialog.

7. Using the Condition Builder, create the condition:
   CommentVar1="" AND CommentVar2="", which tests that CommentVar1 is blank and CommentVar2 is not blank; both conditions must be true for a Comment section to be created.

8. Click OK.

   The sections are now correctly applied from the top of the page down to row 11. The next line is blank, and neither of the two sections we have created are valid, so no further sections are created on this page.

In the Input Window, navigate back to page 1 and check that the changes you have made to the section properties have been correctly applied there as well.
5.5.10.3 Make a detail section expandable

In cases where the number of rows in a detail section is variable, e.g. repeated comment lines, you can make the detail section expandable.

To make a detail section expandable:

1. In the Input Window, double-click the detail section. The Input Section Properties dialog will be displayed.
2. In the Size field, type the minimum size of the section in rows.
3. Select the Expandable checkbox.
4. Click OK.

The section will now automatically expand in size, testing each row below the section, until a condition in a following detail section tests TRUE. Ensure that all sections that are permitted to follow the section contain a suitable condition.

You can also make data variables expandable in an expandable section (Section 5.5.6.5).

Try this...

In the sample project, navigate to page 3 of the input file. Observe that the input sections defined so far have been correctly applied down to row 15. Also observe that Comment sections can be larger than a single row. This can be handled in several different ways, depending on how you intend to use the data in the project:

You can change the Comment section Followed By list to include itself:

1. Double-click any Comment section to display the Input Section Properties dialog.
2. Add Comment to the This section can be followed by list.
3. Click OK to apply the change.

Alternatively, you can make the Comment section expandable:

1. Double-click any Comment section to display the Input Section Properties dialog.
2. Select the Expandable checkbox.
3. Click OK to apply the change.

Try both methods and see how these changes are applied on the previous pages.
5.5.10.4 Rules for creating multiple detail sections

The procedures detailed in Section 5.5.10.2 can be applied when adding further detail sections. The rules applying to multiple detail sections are:

- The detail section that follows the input header section, will be the first valid detail section in the Followed By list of the input header section.
- The first valid detail section in the previous detail section Followed By list is applied next.
- An unconditional detail section will always be applied if tested in the previous detail section Followed By list.
- If there are no valid sections in the last Followed By list, no further detail sections will be applied to that page.
- Detail sections can be expandable (Section 5.5.10.3).

5.5.11 Working with input footer sections

What do you want to do?
- Adjust the input footer section size
- Add input footer sections

What do you want to know about?
- Rules for creating multiple input footer sections

5.5.11.1 Adjust the input footer section size

As with the input header section, you must adjust the size of the input footer section so that it contains all footer text and no text that belongs in a detail section.

Use the Show All Instances command (Section 5.5.2) to examine all instances of the section in the input file.

To adjust the input footer section size:

1. Select the input footer tab on the Status Bar to select and display the input footer section.
2. Click-and-drag the resize handle on the top border of the input footer section to make the section bigger or smaller.
5.5 Designing for Text Input Files

5.5.11.2 Add input footer sections

In cases where the content of the input footer section changes between pages, you can add and use additional input footer sections. The additional input footer sections will be added to the input footers control list. Only one can be used on each page.

To define additional input footer sections:

1. From the Main menu, select **Input** > **Input Footers**. The **Input Footers Control List** dialog will be displayed.

2. Click the Add Row button ( ) to display the **New Footer Section** dialog.

3. In the **Name** field, type the footer section name.

4. In the **Size** field, type the number or rows required for the new input footer section.

5. Click **OK** to close the **New Footer Section** dialog. Do not close the **Input Footers Control List** dialog.

   The input footers control list now has two input footer sections defined. Both input footer sections are currently unconditional (the **Condition** field is empty). If the first in the list is unconditional, it will always be created. So the first footer must be conditional. If the condition tests **TRUE** then the first input footer is applied, if the test fails then the second input footer is applied.

6. Click the Edit button ( ) for the first input footer to display the **Input Section Properties** dialog.

Try this...

In the sample project,

1. Navigate to page 1 in the Input Window.
2. Drag the input footer border down so that the section size is 2 rows:

   The adjusted footer section should start at row 26, just below the last line item.
7. In the **Condition** section, click the ( ) button to display the **Condition Builder** dialog and create a condition that tests **TRUE** when this section is valid. For information on creating conditions, see [Chapter 11].

8. Click **OK** to close the **Input Section Properties** dialog.

9. Click **OK** to close the **Input Footers Control List** dialog.

   The definition of the new input footer section will be applied to the entire file.

---

*The Condition Builder makes available a number of useful system variables and conditions, for example the condition* `Sys.LastPageOfSet` *will test **TRUE** on the last page of each set.*

---

**Try this...**

In the sample project, navigate to page 2 of the input file. Observe that rows 23, 24 and 25 should be part of the input footer section, as they appear on pages 1 and 3. Pages 1 and 3 are the first pages of a new set, whereas page 2 is a continuation page. The existing input footer section is being applied correctly on the first page of each set, but incorrectly on the continuation pages.

Firstly, create a condition that tests **TRUE** on the first page of each set.

1. Double-click the input footer section.
   
   The **Input Section Properties** dialog will be displayed.

2. Click the ( ) button and use the **Condition Builder** to create the expression `Sys.FirstPageOfSet` in the **Condition** field.

3. Click **OK**.

   Now, create a continuation page footer which will be applied when the test for `InputFooter1` fails.

4. From the Main menu, select **Input** ➤ **Input Footers**.
   
   The **Input Footers Control List** dialog will be displayed.

5. Click the Add Row button to display the **New Footer Section** dialog.

6. Type `ContFooter` in the **Name** field.

7. Type 5 in the **Size** field.

8. Click **OK** to close the **New Footer Section** dialog.

9. Click **OK** to close the **Input Footers Control List** dialog.

   The input footers control list now has two input footer sections defined. The first footer in the control list (`InputFooter1`) is conditional. If the condition tests **TRUE** then `InputFooter1` (with 2 rows) is applied, if the test fails then `ContFooter` (with 5 rows) is applied.
5.5.11.3 Rules for creating multiple input footer sections

The procedures detailed in § 5.5.11.2 can be applied when adding further input footer sections. The rules applying to multiple input footer sections are:

- The first input footer that tests TRUE in the control list is applied.
- If none of the input footers in the control list test TRUE on a page, then no input footer will be created on that page.

✓ Try this...

After creating and copying data variables in the input header, detail and input footer sections, the sample project should now look like this:

1. Select View→Input Tree to display the Input Tree (§ 4.1.4).
2. Expand the input header, detail and input footer sections in the Input Tree to display the data variables you have defined.
3. Navigate through each page of the input file to verify that the sections you have defined have been correctly applied on every page.

Your input design is now complete.
If you haven’t already done so, save your project.
5.6 Designing for XML Input Files

The input design process for XML files involves identifying the different elements of the XML input file; this information is recorded in the DataMap. A DataMap for an XML input file is denoted by the .xmp6 file name extension.

What would you like to know about?
- XML file structure (Section 5.6.1)
- Input file properties (Section 5.6.2)
- Defining sets (Section 5.6.3)
- Defining sections (Section 5.6.4)
- Working with data variables (Section 5.6.5)

To create the XML sample project:

1. To start the wizard, click the New Project button ( ) on the Main toolbar.
2. From the Type drop-down, select XML from the available file types.
3. Click ( ) to display the Select Input File dialog and select the file Chapter5Sample.xml from the <install dir>\CF6Samples\Tutorials folder.
4. Click Next.
5. Select the Create a new DataMap radio button and click Next.
   A new DataMap will be created.
6. Ensure the Set headers and footers checkbox and the Table with headers and footers checkbox are cleared and click Next.
7. Click Finish.
   The new sample project will be displayed in the Input Window.
5.6.1 XML file structure

To be properly interpreted, the XML file should be “well-formed” and adhere to the published standards. An example of a well-formed XML file is shown below:

<table>
<thead>
<tr>
<th>XML Element Type</th>
<th>Line</th>
<th>XML Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML declaration</td>
<td>1</td>
<td><code>&lt;?xml version=&quot;1.0&quot; encoding=&quot;ISO8859-1&quot; ?&gt;</code></td>
</tr>
<tr>
<td>comment</td>
<td>2</td>
<td><code>&lt;!-- Member Updates--&gt;</code></td>
</tr>
<tr>
<td>root tag with attribute</td>
<td>3</td>
<td><code>&lt;update total=&quot;2&quot;&gt;</code></td>
</tr>
<tr>
<td>child element start tag with attribute</td>
<td>4</td>
<td><code>&lt;member count=&quot;1&quot;&gt;</code></td>
</tr>
<tr>
<td>subchild element tag</td>
<td>5</td>
<td><code>&lt;ID&gt;1984&lt;/ID&gt;</code></td>
</tr>
<tr>
<td>subchild element start tag</td>
<td>6</td>
<td><code>&lt;changes&gt;</code></td>
</tr>
<tr>
<td>nested subchild element tag</td>
<td>7</td>
<td><code>&lt;street&gt;1667 Elm Street&lt;/street&gt;</code></td>
</tr>
<tr>
<td>nested subchild element tag</td>
<td>8</td>
<td><code>&lt;email&gt;fred.krueger@craven.com&lt;/email&gt;</code></td>
</tr>
<tr>
<td>subchild element end tag</td>
<td>9</td>
<td><code>&lt;/changes&gt;</code></td>
</tr>
<tr>
<td>child element end tag</td>
<td>10</td>
<td><code>&lt;/member&gt;</code></td>
</tr>
<tr>
<td>child element start tag with attribute</td>
<td>11</td>
<td><code>&lt;member count=&quot;2&quot;&gt;</code></td>
</tr>
<tr>
<td>subchild element tag</td>
<td>12</td>
<td><code>&lt;ID&gt;1886&lt;/ID&gt;</code></td>
</tr>
<tr>
<td>subchild element start tag</td>
<td>13</td>
<td><code>&lt;changes&gt;</code></td>
</tr>
<tr>
<td>nested subchild element tag</td>
<td>14</td>
<td><code>&lt;title&gt;Mr&lt;/title&gt;</code></td>
</tr>
<tr>
<td>nested subchild element tag</td>
<td>15</td>
<td><code>&lt;first&gt;Edward&lt;/first&gt;</code></td>
</tr>
<tr>
<td>nested subchild element tag</td>
<td>16</td>
<td><code>&lt;last&gt;Hyde&lt;/last&gt;</code></td>
</tr>
<tr>
<td>nested subchild element tag</td>
<td>17</td>
<td><code>&lt;email&gt;henry.jekyll@stevenson.com&lt;/email&gt;</code></td>
</tr>
<tr>
<td>subchild element end tag</td>
<td>18</td>
<td><code>&lt;/changes&gt;</code></td>
</tr>
<tr>
<td>child element end tag</td>
<td>19</td>
<td><code>&lt;/member&gt;</code></td>
</tr>
<tr>
<td>end root tag</td>
<td>20</td>
<td><code>&lt;/update&gt;</code></td>
</tr>
</tbody>
</table>

Figure 5-14: Example of a well-formed XML file

The elements and structure of the XML file are interpreted according to an underlying set of rules that determine how the XML elements will be applied in the output project. These initial settings can be changed with the following commands:

<table>
<thead>
<tr>
<th>To do this</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change formats of date and numeric data</td>
<td>Input File Properties (Section 5.6.2)</td>
</tr>
<tr>
<td>Ignore selected XML tags in source file</td>
<td>Input File Properties (Section 5.6.2)</td>
</tr>
<tr>
<td>Change how sets are created</td>
<td>Set Define (Section 5.6.3)</td>
</tr>
<tr>
<td>Change how sections are created</td>
<td>Input Tag Properties (Section 5.6.4)</td>
</tr>
<tr>
<td>Rename a section</td>
<td>Input Tag Properties (Section 5.6.4)</td>
</tr>
</tbody>
</table>
5.6.2 Input file properties

The input file properties determine the input format for dates and numbers, and which tags will be ignored when the input file is processed.

To change the input file properties:

1. Select the Input Window by clicking anywhere inside the input window frame.
2. From the Main menu, select Input File Properties. The Input File Properties dialog will be displayed.

Figure 5-15: XML Input File Properties dialog

3. In the Data format section, select the format used for date and decimal values in the input file.
4. In the Ignored Tags section, use the toolbar buttons to add the tags you want to ignore or remove the tags you want to include.
5. Click OK.

The changes will be applied to the input file and the input window will be refreshed.
5.6.3 Defining sets
You can select one tag to indicate where a new set will be created:

To define sets:
1. From the Main menu, select Set►Define.
   The Define Set dialog will be displayed.
2. Select the tag that defines the start of a new set.
3. Click OK.

5.6.4 Defining sections
You can select one or more tags to indicate where a new data section will be created:

To define sections:
1. In the Input Window, select a tag that defines the start of a new section.
2. From the Main menu, select Input►Tag Properties.
   The Tag Properties dialog will be displayed.
3. Type the name of the section.
4. Click OK.
   Follow the same procedure to unselect section tags.

To rename a section:
1. In the Input Window, select the tag that defines the start of the section.
2. From the Main menu, select Input►Tag Properties.
   The Tag Properties dialog will be displayed.
3. Type the new name in the Name field.
4. Click OK.
Try this...

Follow the instructions in Section 5.6 and create a new sample XML input project.

In the sample project, the wizard applies a standard set of rules to interpret the XML input file and create sets, sections and data variables. These initial settings are displayed in the Input Window.

Firstly, look at how sets are defined:

1. From the Main menu, select the Set Define command.
   The Define Set dialog will be displayed with the <order> tag highlighted.
   A new set will be created wherever the <order> tag occurs in the input file, which is the setting we want.

Now, look at how sections are created:

In the Input Window, note that some tags are ticked, and others are not. The tick marks indicate which tags are being used to create sections in the output. In this case, we only want the <order> and <item> tags to create sections, so we need to turn off section creation on the <name> and <address> tags as these belong in the same section as the order number.

1. In the Input Window, click the <name> tag.
2. From the Main menu, select Input Tag Properties.
   The Tag Properties dialog will be displayed.
3. Clear the Create section checkbox and click OK.

Repeat steps 1 to 2 and turn off section creation for the <address> section.
5.6.5 Working with data variables

Data variables are created automatically from the tag attributes and data elements in the input file. The data variables created are shown in the Input Tree view:

![Input Tree View]

**What would you like to know about?**

- Changing the properties of data variables ([Section 5.6.5.1](#))
- Changing the input data format ([Section 5.6.5.2](#))
- Copying data variables to the project ([Section 5.6.5.3](#))

### 5.6.5.1 Changing the properties of data variables

To change the properties of data variables:

1. In the Input Window, double-click the data variable. The **Data Variable Properties** dialog will be displayed.

2. With the **Data Variable Properties** dialog you can:
   - Change the name of the data variable by typing a new name in the **Variable Name** field.
   - Select the data type from the **Data Type** drop-down.
   - See also [Section 5.6.5.2](#) for information on how Create!form Designer recognizes date and numeric formats.
3. Click OK.

---

**Figure 5-17: Data Variable Properties dialog**

### 5.6.5.2 Changing the input data format

Create!form Designer interprets dates and numeric data in the input file using the data formats defined by the input file properties. For example, you can change the decimal separator to a comma, or the date order from year-month-day to month-day-year.

**To change the input data format:**

1. From the Main menu, select **Input > File Properties**. The **Input File Properties** dialog will be displayed (Figure 5-15).
2. From the **Date order** drop-down, select the required date format.
3. From the **Decimal symbol** drop-down, select the required numeric format.
4. Click **OK**.

### 5.6.5.3 Copying data variables to the project

Data variables can be displayed in the project by dragging the data variable from either the Input Window, or the Input Tree, into the Design Window.

**To copy a data variable to the project:**

1. In the Input Window, select the data variable.
2. Drag-and-drop the selected variable from the Input Window across into the Design Window.
5.7 Designing for JDE PDF Input Files

For more information on adding text variables and other objects to the project, see Section 9.2.

Try this...
In the sample project,
1. Drag-and-drop the order number and customer details from the Input Tree into the corresponding output section.
2. Drag-and-drop the item details from the Input Tree into the corresponding output section.

Your input design is now complete.
If you haven’t already done so, save your project.

5.7 Designing for JDE PDF Input Files

The J.D. Edwards PDF format is a self-defining proprietary PDF format generated by PeopleSoft EnterpriseOne. The contents of the PDF file are pre-mapped, which Create!form Designer automatically translates into sections and data variables.

The input design process for JDE PDF files involves defining how sets are created and mapping data variables to the output; this information is recorded in the DataMap. A DataMap for an JDE PDF input file is denoted by the .jmp6 file name extension.
To view the input file elements mapped by the DataMap:

- Click the Input Tree ( ) on the Main toolbar or select View ► Input Tree from the Main menu.

What would you like to know about?

- Input file properties (Section 5.7.1)
- Defining sets (Section 5.7.2)
- Input sections (Section 5.7.3)
- Data variables (Section 5.7.4)

### 5.7.1 Input file properties

Non-breakable input sections and the date order format are defined by the input file properties.

To add and remove non-breakable input sections:

1. Select the Input Window by clicking anywhere inside the input window frame.
2. From the Main menu, select Input ► File Properties.
3. To add a section to the Non-breakable sections, click the Add Row button ( ), click the drop-down button on the new row, and select the section from the list.
4. To remove a section, select the section in the Non-breakable sections list, and click the Delete Row button ( ).
5. Click OK.

To change the input file date format:

1. Select the Input Window by clicking anywhere inside the input window frame.
2. From the Main menu, select Input ► File Properties.
3. Select the required format from the Date order field.
4. Click OK.
5.7.2 Defining sets

Defining sets involves specifying when a new set will commence. Typically a new set commences every time the page numbering restarts, but you can also specify that a new set commences when data such as the customer number or invoice number changes.

You can use the following means to determine when a new set starts:

- when the value of a page number variable is equal to 1
- when the value of a variable changes
- one set per page
- no sets

To define sets:

1. From the Main menu, select Set>Define. The Define Set dialog will be displayed.

2. Select one of the radio button options to determine when a new set starts:
   - This project does not contain sets
   - New set occurs when:
     - From the Section drop-down, select the name of the section containing the variable you want to use to define sets, then an option and the required variable from the Condition field:
       - Page number variable ... equals 1
       - Value of variable ... changes
     - New set starts on every page.

3. Click OK.

5.7.3 Input sections

Input sections are created automatically from the pre-mapped contents of the input file. The sections defined in the input file are shown in the Input Tree view. The properties of input sections in a JDE PDF input file cannot be changed.

5.7.4 Data variables

Data variables are created automatically from the pre-mapped contents of the input file. The data variables are shown in the Input Tree view. The properties of data variables in a JDE PDF input file cannot be changed.
To copy a data variable to the project:

1. In the Input Window, select the data variable.

2. Drag-and-drop the selected variable from the Input Window across into the Design Window.

For more information on adding text variables and other objects to the project, see Section 9.2.

### 5.8 Creating Derived Variables

Derived variables are expressions that are used to combine and manipulate the values of data variables. To create a derived variable, select **Input > Derived Variables** from the Main menu to display the Derived Variables dialog. For information on creating expressions, see Section 11.1.

#### Examples:

<table>
<thead>
<tr>
<th>Derived Variable</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>TotalPrice</td>
<td>(UnitPrice+SalesTax)*Qty</td>
</tr>
<tr>
<td>LastChar</td>
<td>substring(ProdDesc,length(ProdDesc),1)</td>
</tr>
<tr>
<td>AddressLine1</td>
<td>Title+FirstName+LastName</td>
</tr>
<tr>
<td>DaysInMonth</td>
<td>Day(date(year(Sys.Today),month(Sys.Today)+1,1)-1)</td>
</tr>
</tbody>
</table>

For information on how to use derived variables in the project, see Section 9.2.
Project Settings

Before you start designing your project, you should review and edit the project properties, the project page style and the project background; otherwise, you may have to redesign your project if these settings are changed later.

The following topics are covered:

- Project Properties
- Project Page Style
- Project Background
6.1 Project Properties

Project properties include various print options and default styles, which apply to the whole project.

What do you want to know about?
• Collate options
• Duplicates
• About format styles

To edit project properties
1. From the Main menu, select Project Properties. The Project Properties dialog will be displayed.
2. Define the collate options (Section 6.1.1).
3. Define the duplicate options (Section 6.1.2).
4. Create format styles (Section 6.1.3).
5. In the Pages per sheet field, select the number of pages you want to be printed on each sheet. For information about setting the page size and orientation, see Section 6.2.
6. Click OK to save the project properties.

6.1.1 Collate options

There are three ways to order the output of pages:
• Uncollate
• Collate
• (collate) By Set

The dynamic image on the Project Properties dialog gives a graphical representation of the selected collate option.
6.1 Project Properties

6.1.2 Duplicates

By specifying a value greater than 1 the **Number of duplicates** field, you can set the project to process the input file multiple times. By using this option, you can generate multiple copies at the printer, rather than sending all copies from the server.

---

**You can also specify that the spool file is processed multiple times using Create!stream. For more information see the Create!stream User Guide and the Create!form Server User Guide. The collate options apply to multiple copies, regardless of how the number of copies is specified.**

---

6.1.3 About format styles

You can create format styles that customize the way numbers, dates and currencies are displayed. Format styles allow you to specify such things as:

- whether a period (.) or comma (,) is used as a decimal symbol
- the currency symbol
- whether the date will be displayed as day-month-year or month-day-year

Format styles are global; once you have created a style you can apply it in any project.

When you install Create!form Transform, some format styles are automatically created (including the appropriate one for the regional settings of your computer). You can customize these format styles or create new ones.

**What do you want to do?**

- Create a format style
- Customize a format style
- Set the default format style for new objects

6.1.3.1 Create a format style

1. In the **Project Properties** dialog, select **Manage Formats** from the **Default format style** drop-down.
   The **Manage Format Styles** dialog will be displayed.

2. Click **Create**.
   The **Create Format Style** dialog will be displayed.

3. Type a name for the style.
4. Choose whether to base the initial style settings on:
   - a Windows locale, and choose a locale or;
   - a previously defined format style.
   The locale selected will determine the regional sort mode (Section 8.3.2).

5. Click OK.
   The <name of format style> dialog will be displayed.

6. Make all required settings for numbers, currency and dates and click OK.
   The Manage Format Styles dialog will be displayed.

7. When you have finished managing your styles, click Close to return to the
   Project Properties dialog.

8. Click OK to return to the project.

You can also manage format styles with the Tools►Format Styles command.
6.2 Project Page Style

6.1.3.2 Customize a format style
1. In the Project Properties dialog, select Manage Formats from the Default format style drop-down.
   The Manage Format Styles dialog will be displayed.
2. Select the format style you want to customize and click Edit.
3. Make the necessary changes and return to the project.
   All objects using the edited format style will be updated.

6.1.3.3 Set the default format style for new objects
When you set the default, it will be the default for all objects created in that project. When you change the default, all objects using the (Project Default) format style will be updated.
1. In the Project Properties dialog, select the format you want as the default for all new objects in the Default format style field.
2. Click OK.
For information on applying different format styles to particular objects, see Section 9.4.3.3.

6.2 Project Page Style
The Page Setup dialog provides the page settings for the following options:

- page size
- page control

These settings will apply to all pages in the project.

To set the page style:
1. From the Main menu, select File→Page Setup.
   The Page Setup dialog will be displayed.
2. In the Page size field, type the page Width and Height in columns and rows.
3. In the Page control field, select one of the following options:
   - Form feeds - a form feed character will be used
   - Fixed height - blank lines will be inserted to give a fixed page height
   - None - the output file will not be paginated
Chapter 6: Project Settings

6.3 Project Background

The page view consists of two superimposed layers:

- **Background layer**
  Contains fixed content.
- **Design layer**
  Contains all variable content i.e. variable text and variable subform objects.

The background layer will appear on every page, and is overprinted by the design layer.

To add text to the background:

Click the Project Background tab and edit the page. For information on inserting objects, see Chapter 9.

When you double-click the Project Background tab, the Page Setup dialog will be displayed with which you can alter the page size and other properties.
Sections

Sections are defined both in the input and the output. The following topics explain how to create different types of sections in the output.

The following topics are covered:

▷ About Sections
▷ Data Sections
▷ Header and Footer Sections
▷ Alternate Sections
▷ Inserted Sections
▷ Setting Section Properties
Chapter 7: Sections

7.1 About Sections

Each type of section displays a particular type of information or occupies a particular space on the page. Sections typically have many occurrences, and where you make layout or format changes to one occurrence, the changes are automatically made to all other occurrences.

In Chapter 5 you were introduced to the types of input sections:

- detail sections, which typically contain information such as line item data
- input header and input footer sections, which typically bracket detail sections

These sections are used to define the input data.

In the project (viewed in the Design Window), you can use the following types of sections to manipulate the display of data:

- Data Sections (Section 7.2)
- Header and Footer Sections (Section 7.3)
- Alternate Sections (Section 7.4)
- Inserted Sections (Section 7.5)

7.2 Data Sections

When you create a project, all sections in the input file are automatically copied across to the project. This type of section is called a data section and each data section has the same name as the corresponding section in the input file.

You cannot delete a data section but you can hide it so it does not display. This is done from the Section Properties dialog.

What do you want to know about?

- Data section display options
- Setting Section Properties

7.2.1 Data section display options

You can specify a data section to display:

- always
- never
- on condition

By default, data sections are set to always display.
To set the display properties of a data section:

1. From the Main menu, select **Section Properties**. The **Section Properties** dialog will be displayed.
2. Click the **Display** tab.
3. Select one of the radio button options:
   - **Always** (to display the section unconditionally)
   - **Never** (to hide the section unconditionally)
   - **On Condition** and create the condition (to display the section conditionally)
     For information on creating conditions, see Chapter 11.

To prevent data from the section being used in calculations when the section is hidden, select the **Ignore data from this section when hidden** checkbox.

**Viewing the properties of sections that are not displayed**

If a section is set to never display, or conditionally, and the conditions are never met, the section properties can only be displayed by double-clicking on the section in the Project Tree. For information on the Project Tree, see Section 4.1.3.

### 7.3 Header and Footer Sections

Creating header and footer sections provides the basic page layout structure and makes designing your project easier.

You can create header and footer sections around the following units:

- the project
- a set
- a page
- a table

Once created, you can format header and footer sections like any other section.

**What do you want to know about?**

- Continuation header and footer sections
- The display order of header and footer sections

**What do you want to do?**

- Create set headers and footers with the wizard
- Create and remove header and footer sections
- Create header and footer sections from an input section
7.3.1 Create set headers and footers with the wizard

When a standard project (Section 3.3.5.1) is created, you are given the option of creating set headers and footers. When this option is selected, the first occurrence of the input header section on the first page of the set will be used as the source for the set header. Similarly, the last occurrence of the input footer section in the set will be used as the source for the set footer.

7.3.2 Continuation header and footer sections

As well as header and footer sections, you can also create continuation header and continuation footer sections. Continuation headers and footers appear wherever there is a page break within that header/footer unit. For example a set continuation header appears on every continuation page of a set. Continuation header and footer sections can have the same content and design as the header/footer section, or be totally different.

For information on creating and removing continuation header and footer sections, see Section 7.3.3.

7.3.3 Create and remove header and footer sections

This procedure deals with headers and footers for the project, sets and pages. For information on creating table header and footer sections, see Section 8.3.3.

To create header and footer sections for the project, sets or pages:

1. From the Main menu, select the appropriate menu option:
   - Project\Headers and Footers
   - Set\Headers and Footers
   - Page\Headers and Footers

2. From the relevant dialog, select the appropriate checkbox to create:
   - a header.
   - a continuation header, and if so whether the continuation header will be the same as the header or different.
   - a footer.
   - a continuation footer, and if so whether the continuation footer will be the same as the footer or different.
   - For set headers and footers, you have the option of creating an Additional first page footer. This is useful for things like a payment advice where you may want to print a check on the bottom of the first page.
7.3 Header and Footer Sections

3. To copy all text objects from another data section, click Create From. For more information see Section 7.3.4.

4. Click OK to close the Header/Footer dialog.
The sections you have selected will be created and displayed in the Design Window and the Project Tree.
You can now format the newly created sections.

To turn a header or footer section off:

Header and footer sections can be turned off (deleted) or turned on (created) at any time. This is done from the relevant headers and footers dialog.

7.3.4 Create header and footer sections from an input section

The Create From function lets you copy the contents and size of another section into header and footer sections in the project. When you use this function, the following occurs:

- All text objects from the specified section and its corresponding original section in the input file are cut and pasted to the new header/footer section. (If objects appear in the specified section and its original section in the input file, Create!form Designer uses the instance from the specified section.)
- If it is bigger than the original, the size of the specified section is adopted by the new header/footer section;
- The section you copied the variables and properties from is set to never display.

To use the Create From function:

1. From the relevant Header/Footer dialog or Table dialog, click Create From.
The Create From dialog will be displayed.

2. In the Source field, select the data section from which you want to copy the contents and dimensions.

3. In the Occurrence field, specify the occurrence of the data section you want to map from. For headers, this will usually be the first occurrence, and for footers, the last occurrence.

4. Click OK to return to the originating dialog.
7.3.5 The display order of header and footer sections

It is possible to have multiple header and footer sections on the same page. Their display order is governed by rules. To understand these rules, consider the following hypothetical project that contains:

- two pages.
- one set.
- a table that runs over both pages.
- project, set, page and table header/footer sections.

The header/footer rules would dictate the following display order:

<table>
<thead>
<tr>
<th>Page 1</th>
<th>Page 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>project header</td>
<td>project continuation header</td>
</tr>
<tr>
<td>set header</td>
<td>set continuation header</td>
</tr>
<tr>
<td>page header</td>
<td>page header</td>
</tr>
<tr>
<td>table header</td>
<td>table continuation header</td>
</tr>
<tr>
<td>data section</td>
<td>data section</td>
</tr>
<tr>
<td>table continuation footer</td>
<td>table footer</td>
</tr>
<tr>
<td>page footer</td>
<td>page footer</td>
</tr>
<tr>
<td>set continuation footer</td>
<td>set footer</td>
</tr>
<tr>
<td>project continuation footer</td>
<td>project footer</td>
</tr>
</tbody>
</table>

7.4 Alternate Sections

Where you have a data section that has a different layout and content in different occurrences, you can create alternate sections to cater for those different occurrences. When certain user-defined conditions are met, the alternate sections will replace the original data section. All alternate sections share the same source and variables as the sections they are alternates for.

There are many possible and equally valid ways of varying the content and format of a section based on its content. Here are some other methods that you may find useful:

- create multiple detail sections in the input design (Section 5.5.10)
- conditionally display text and drawing objects (Section 9.4.3.1)
- use variable subforms (Section 9.2.2.5)
7.5 Inserted Sections

To create an alternate section:

1. Select the data section you want to create an alternate for.
2. From the Main menu, select Section ▶ Alternates
   The Alternates <section> dialog will be displayed.
3. To create the condition for displaying the alternate, click the Condition button
   For information on creating conditions, see Section 11.2. You can edit this condition at any time. Click OK to return to the Alternates <section> dialog.
4. The Alternate Section Name field contains the default name for the new alternate section. You can type another name in here if you want, but be sure to use one that will clearly associate it with the original data section.
5. If necessary, click the Add Row button ( ) to create another condition and another alternate section.
6. If necessary, use the arrow buttons to change the order that Create!form Designer evaluates the conditions.
7. Click OK.
   Where the conditions are met, the alternate section(s) will replace the data section.
8. Add data variables (Section 9.2.1), and format the alternate section.

To delete an alternate section:

1. From the Alternates <section> dialog, select the row that refers to the alternate section you want to delete.
2. Click delete ( ).
3. Click OK.
   The alternate section will be removed from the project.

7.5 Inserted Sections

You can insert one or more sections before and after a detail section. Inserted sections have the following properties:

• always keeps with the parent detail section
• user variables are defined independently from the parent section
• the primary source is the parent section
• no alternates are permitted

Inserted sections will otherwise act like a normal detail section, and can be edited in the same way.

To insert a section before or after another section:

1. In the Design Window, select the section you want to insert the new section before or after.
2. From the Main menu, select Section ➤ Inserted Sections
   The Insert Sections dialog will be displayed.
3. Use the tools provided to add, remove and rearrange inserted sections in the Inserted before <section> and Inserted after <section> list boxes.
4. Click OK.

The Design Window will refresh with the new inserted section(s) displayed.

7.6 Setting Section Properties

What do you want to know about?
• Section positioning options
• Adjusting section size
• Fixed and auto-expanding sections
• About parent/child relationships
• Keeping sections together
• Setting repagination control
• About sources
• Adding sources

To display the Section Properties dialog:
• In the Design Window in Graphics View (Section 4.1.1), double-click in the active section where there are no objects, or
• Double-click on the section tab at the bottom of the Design Window, or
• In the Project Tree, right-click on the section name and select Properties.

7.6.1 Section positioning options

Depending on the type of section, you may be able to:
• fix the section in a location on the page.
7.6 Setting Section Properties

- **float** the section immediately after the previous section, regardless of where the previous section is.
- **sink** the section so that it, along with any sections that follow it, are forced to the bottom of the page.

None of these options will change the order of the sections, only their positioning.

The following rules apply to the different section types:

- Data sections, alternate sections and inserted sections can either float or be fixed.
- Table header sections can either float or be fixed. Other header sections can only float.
- All footer sections can either sink or float.

To specify section positioning, click the appropriate button on the **General** tab of the **Section Properties** dialog.

**To position a fixed section:**

- In the Design Window, select the section and use the top handle of the section to move it up or down the page.

### 7.6.2 Adjusting section size

To adjust the size of a section do one of the following:

- in the Design Window, drag the bottom handle of the selected section
- specify the size on the **General** tab of the **Section Properties** dialog

### 7.6.3 Fixed and auto-expanding sections

By default, every occurrence of each section type will be the same size. Alternatively, you can set the following sections to auto-expand:

- data
- alternate
- project headers
- set headers
- table headers
- page headers
When you set a section to auto-expand, you specify that a minimum distance between the bottom of the lowest text object and the bottom of the section will be maintained, regardless of the size of variable objects in the section. If the number of lines in a variable text object increases or decreases, the size of the section will change accordingly.

**To auto-expand a section:**

1. On the **General** tab of the **Section Properties** dialog, select the **Auto-expand** checkbox.

2. In the **Gap between lowest object and section bottom** field, type the minimum distance between the lowest *text* object and the bottom of the section.

3. In the **Size** field, specify a minimum overall section size.

4. Click **OK**.

### 7.6.4 About parent/child relationships

You can define sections as having a parent/child relationship with other sections. A parent section can have multiple child sections. There are several reasons for doing this:

- When you create parent/child relationships, it enables the “keep with” function, so that you can keep sections together, where they might otherwise be forced onto another page.
- To enable the creation of a table involving more than one section type. When the parent is used to define a table, the child section is automatically included in the table. This is the only way a sub-detail section can be included with its parent-detail section in a table.
- From the parent section, you can use section specific variables from the first occurrence of each associated child section that follows the parent.

**Parent/child relationship requirements**

- The parent/child relationship will only be preserved if the child follows immediately after the parent or another child of the parent.
- All child sections must be set to float.
Alternate sections and parent/child relationships

Alternate sections inherit the parent/child relationships of the section they are alternates for. However the Keep With setting of a parent/child relationship is not inherited.

7.6.5 Keeping sections together

Sections can be associated by a parent/child relationship so they keep together. For more information, refer to Section 7.6.4.

Auto-expanding sections and the Keep With option

When you set an auto-expanding section to break over the end of a page, the Keep With setting will be ignored.

To define a parent/child relationship:

1. Open the Section Properties dialog of the section that will be parent.
2. Click the Repagination tab.
3. For each section you want to be a child:
   • Select the checkbox in the Child Section column.
   • Type an alias or leave blank.
     The alias is used within expressions and conditions to resolve problems with duplicate names in child or source sections.
4. Select the Keep With checkbox for any children you want to keep with the parent.
5. Click OK.

7.6.6 Setting repagination control

To start a new page when a specific section occurs:

1. Double-click the section tab to display the Section Properties dialog.
2. Select the Repagination tab.
3. Select the Start new page checkbox.
4. Click OK.
To control repagination of auto-expanding sections:

If the section size is auto-expanded (Section 7.6.3), you can specify various pagination controls for the section.

1. Double-click the section tab to display the Section Properties dialog.
2. Select the Repagination tab.
3. To allow the section to break over the page, select the Break auto-expanding section and overflow to next page checkbox.
4. To specify the minimum number of lines kept together at the start or end of a broken text object, select the Orphan control checkbox and type the value in the Minimum lines field.
5. To keep the paragraphs together without splitting, select the Don’t split paragraphs checkbox.
6. Click OK.

7.6.7 About sources

All data variables, user variables and derived variables used within a section are derived from the section source(s). The section source(s) are other sections. To view a section source(s), open the Source tab of the Section Properties dialog.

The primary source of each section, is the section itself. If you change the name of the section, the change is reflected in the primary source also, but the content remains the same and is a reflection on what type of section it is. The following table explains the content of the primary source for each section type.

<table>
<thead>
<tr>
<th>Section type</th>
<th>Content of primary source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>The corresponding detail section in the input file.</td>
</tr>
<tr>
<td>Alternate</td>
<td>The same primary source as the section it is an alternate for. (The alternate section also inherits any secondary sources that exist in the original section.)</td>
</tr>
<tr>
<td>Inserted</td>
<td>The same primary source as the section it is inserted from. (The inserted section also inherits any secondary sources that exist in the original section.)</td>
</tr>
<tr>
<td>Header/footer</td>
<td>The primary source of a header or footer section is the section itself. You can add source(s) when you are creating the header/footer section using the “create from” function, or from the Source tab.</td>
</tr>
</tbody>
</table>
7.6 Setting Section Properties

7.6.8 Adding sources

At any time you can add data sections from within the project as additional sources. Each additional source can have an alias name used to qualify any data variable used in conditions or expressions in that section. The alias is used within expressions and conditions to resolve problems with duplicate names in child or source sections.

When defining an additional source you must also specify the occurrence. For example, the first occurrence means the source is the first occurrence of that section in the set.

Typically when a section has been defined as a source for another section, to prevent duplication, the source section should be set not to display.

To add a section as a source:

1. Open the Section Properties dialog (Section 7.6).
2. Select the Source tab.
3. Click Add Source.
   A new row of source will be added.
4. From the Section drop-down, select the section you want to add as source.
5. Type an alias or leave blank.
   The alias is used within expressions and conditions to resolve problems with duplicate names in child or source sections.
6. In the Occurrence column, select which occurrence of the selected section within the set, that you want as source.
7. Click OK.
Tables

Tables allow you to group repeated data sections so they can be labelled, outlined, sorted and summarized.

The following topics are covered:

- About Tables
- Working with Tables
- Table Properties


8.1 About Tables

You can create a table around related repeated sections. Tables provide a robust structure for designing projects. They allow you to:

- sort data into different groups
- create header and footer sections for the different groups of data within the table
- draw table objects such as lines and boxes, that dynamically fit around the table
- calculate sub-totals and totals for the different groups of data you have created (Section 11.5)

Tables are the best way of handling repeated sections. For example in an invoice project, it is recommended that you define a table for the data sections.

What do you want to know about?
- Working with Tables (Section 8.2)
- Table Properties (Section 8.3)

8.2 Working with Tables

What do you want to do?
- Create tables with the wizard (Section 8.2.1)
- Create a table (Section 8.2.2)
- Change table properties (Section 8.2.3)
- Delete a table (Section 8.2.4)

8.2.1 Create tables with the wizard

When a standard project (Section 3.3.5.1) is created, you are given the option of creating tables. When this option is selected, a table is created for the data sections in each set. The project initially only has one data section defined. The table header and footer positions will change as the set definition is changed.

8.2.2 Create a table

To create a table:

1. Select the section you want to create the table for.
   If the table will include multiple sections, this will be the parent.
2. From the Main menu, select **Table**►**New**.
The **New Table** <section> dialog will be displayed.

3. In the **Table name** field, type the table name.
The table name will form the basis for:
   - any table header or footer section names.
   - For example where the table is called “InvoiceData”, the header and footer sections will be called “InvoiceData Header”, “InvoiceData Footer”.
   - the table levels that follow.
   - For example where the table is called “InvoiceData”, the levels will be called “InvoiceData L1”, “InvoiceData L2” etc.

4. To sort the contents of the table or level, click in the grid or click the Add Row button ( ).
   A new level will be created, see **Section 8.3.1**.
   In the new row, specify:
   - The data variable you want to sort or group by.
   - The sort order. If the order is already correct, select **None**.
   - The sort mode. For information on sort modes, see **Section 8.3.2**.

5. To create headers and footers for the table or a level, see **Section 8.3.3**.

6. Select the required grouping options, see **Section 8.3.4**.

7. To specify pagination controls for each table level, see **Section 8.3.6**.

8. To add further levels, repeat steps 4. to 7.

9. When you have finished adding levels and setting table properties, click **OK**.
   In the Design Window, new tabs will be created for each new:
   - header section
   - footer section
   - continuation header section if it is not set to be **Same as first header**
   - continuation footer section if it is not set to be **Same as last footer**

**8.2.3 Change table properties**

**To change the properties of a table:**

1. Click inside any table section.

2. From the Main menu, select **Table**►**Properties**.
The **Table Properties** <section> dialog will be displayed.
3. In the **Table Name** field, type the table name. You can change the name of the table at any time.

4. To change how the table is sorted, click an existing table level or, to add a new level, click the Add Row button ( ). For information on table levels, see **Section 8.3.1**.

   In the level row, specify:
   - The data variable you want to sort or group by.
   - The sort order. If the order is already correct, select **None**.
   - The sort mode. For information on sort modes, see **Section 8.3.2**.

5. To create or remove headers and footers for the table or a level, see **Section 8.3.3**.

6. Select the required grouping options, see **Section 8.3.4**.

7. To specify pagination controls for each table level, see **Section 8.3.6**.

8. When you have finished modifying the table properties, click **OK**.

### 8.2.4 Delete a table

When you delete a table, only the definition of the table and its corresponding header and footer sections is deleted. The data sections the table is built around are preserved in their original state.

**To delete a table:**

1. Select any section contained in the table.
2. From the Main menu, select **Table ➤ Delete**.
   A confirmation dialog will be displayed.
3. Click **OK** to delete the selected table.

### 8.3 Table Properties

What do you want to know about?
- Table levels (**Section 8.3.1**)
- Sorting modes (**Section 8.3.2**)
- Header and footer sections (**Section 8.3.3**)
- Grouping options (**Section 8.3.4**)
- About parent/child relationships (**Section 8.3.5**)
8.3 Table Properties

- Setting table pagination controls (Section 8.3.6)

8.3.1 Table levels

Within a table you can create multiple levels. Levels are the mechanism that allows the data to be sorted and grouped.

To create a table level, refer to Section 8.2.3.

Examples of how you can use table levels can be found in the following topics:
- Header and footer sections (Section 8.3.3)
- Grouping options (Section 8.3.4)
- Setting table pagination controls (Section 8.3.6)

8.3.2 Sorting modes

The following table demonstrates the ASCII, regional and numeric sort modes.

<table>
<thead>
<tr>
<th>Unsorted sample</th>
<th>Sort mode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASCII</td>
</tr>
<tr>
<td>(space)</td>
<td>(space)</td>
</tr>
<tr>
<td>02</td>
<td>02</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>c</td>
<td>c</td>
</tr>
</tbody>
</table>

The sort order when using regional mode can vary with the Windows locale on which the project’s default format style is based (Section 6.1.3).
8.3.3 Header and footer sections

You can create or remove header and footer sections for the table as a whole and for specific levels within the table.

To create or remove header and footer sections:

1. In the Table Properties dialog (Section 8.2.3), select the level you want to create or remove headers and footers for.
2. Select or clear the appropriate checkbox for:
   - **Header** before every new grouping in that level.
   - **Continuation header** for a continuation of a group in that level, and if so, whether it will be the same section as the header or different.
   - **Footer** after every grouping in that level.
   - **Continuation footer** for a continuation of a group in that level, and if so, whether it will be the same section as the footer or different.
3. To create a header or footer section using the content and size of an existing data section, click the Create From button. For more information see Section 7.3.4.

8.3.4 Grouping options

You can control how sections within a table or table level are grouped and sorted from the following options on the Table Properties dialog (Section 8.2.3):

Selecting the Use only first checkbox and typing a value in the characters for sorting/grouping field can have two functions:

- If you are sorting the data, it will qualify the sorting process to only use the number of characters specified. If you set it to use only the first three characters, it will only sort based on those first three characters of every value.
- If you are using headers and footers for the level, it will group the values, based on the number of characters specified.

*Create a derived variable in the input design if you want to sort or group the data using other criteria.*
8.3 Table Properties

The example below shows the various sections within a table when you use the group data sections using only the first character and specify level 1 headers and footers.

<table>
<thead>
<tr>
<th>Page 1</th>
<th>Page 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Header</td>
<td>Table Cont. Header</td>
</tr>
<tr>
<td>Level 1 Header</td>
<td>Level 1 Cont. Header</td>
</tr>
<tr>
<td>A........</td>
<td>B........</td>
</tr>
<tr>
<td>Level 1 Footer</td>
<td>Level 1 Footer</td>
</tr>
<tr>
<td>Level 1 Header</td>
<td>C........</td>
</tr>
<tr>
<td>B........</td>
<td>C........</td>
</tr>
<tr>
<td>B........</td>
<td>C........</td>
</tr>
<tr>
<td>B........</td>
<td></td>
</tr>
<tr>
<td>Level 1 Cont. Footer</td>
<td>Level 1 Footer</td>
</tr>
<tr>
<td>Table Cont. Footer</td>
<td>Table Footer</td>
</tr>
</tbody>
</table>

*Figure 8-1: An example of grouping in tables.*

8.3.5 About parent/child relationships

Tables can contain a parent data section and associated children sections. You must create parent/child relationships if:

- the table will contain multiple section types
- you want to keep sections together

For requirements of parent/child relationships, refer to

---

To create a table containing multiple section types:

1. Create a parent/child relationship between the different sections and select the “Keep With” function (Section 7.6.5) to keep the child section with the parent section.
2. Create a table around the parent (Section 8.2.2).
8.3.6 Setting table pagination controls

You can specify pagination controls for the table as a whole or individual levels within the table.

To set table pagination controls:

1. From the Table Properties dialog, select the level you want to specify pagination controls for.

2. In the Pagination Control section, specify your pagination controls:
   - To start a new page when a new group of the level appears, select the New page checkbox.
   - To specify a minimum number of occurrences of sections in that level that can appear on the top of the page, select the Top orphan control checkbox, and type the minimum number.
   - To prevent a group from splitting over a page, select the Keep together checkbox. If the group is too large to fit on a page, it will start on the next page.
   - To specify a minimum number of sections in the group that can appear at the end of a page, select the Bottom orphan control checkbox and specify the minimum number.
Objects

You can enhance your project by adding text to pages.

The following topics are covered:

- About Objects
- Creating Objects
- Editing and Formatting Objects
- Setting Object Properties
- Selecting and Editing Objects in Graphics View
Chapter 9: Objects

9.1 About Objects

The contents of objects can be static, or you can use variables to control the content of certain objects.

Create!form Transform allows you to create, format and edit objects in many ways; the method you choose will depend on a number of different factors including the type of object you are creating, where the data used to define the object is being sourced from, whether the object is visible, and your own preferences for how you want to work in the Design Window.

Before you begin to create objects, you should be familiar with the following topics:

- About objects and sections (Section 9.1.1)
- Selecting objects in different views (Section 9.1.2)
- Using variables to control objects (Section 9.1.3)
- Window display preferences (Section 9.1.4)

What do you want to know about?

- Creating Objects (Section 9.2)
- Editing and Formatting Objects (Section 9.3)
- Setting Object Properties (Section 9.4)
- Selecting and Editing Objects in Graphics View (Section 9.5)

9.1.1 About objects and sections

When you create an object it will belong to the currently selected section, regardless of where the object is located on the page. You can only select and work with objects belonging to the currently selected section. The object will be displayed in every occurrence of that section. Editing the object in any occurrence of a section changes the object in all occurrences of the same section.
9.1 About Objects

9.1.2 Selecting objects in different views

You can select and edit objects that are visible in the Graphics View directly with the cursor.

When objects are not displayed, or are obscured or hidden

Alternatively, if the objects are not displayed, or are obscured or hidden, you can select objects in either the Listing View or the Project Tree; both of these views offer a non-graphical list of all objects in a section. Follow the procedures detailed in Section 9.4 to display and edit the properties of any object from any view, including the position, size, content and format of objects.

When sections are not displayed

If a section is not displayed, the objects it contains can still be edited from the Project Tree. Follow the procedures detailed in Section 9.5.3 to display and edit objects in hidden sections.

9.1.3 Using variables to control objects

You can use variables in a variety of ways to control the content and display of objects on the page, including:

- to provide the content of text objects (Section 9.2.1)
- to control when objects are visible (Section 9.4.3.1)

The types of variables available, and when to use them, are summarized below:

<table>
<thead>
<tr>
<th>Use</th>
<th>To display</th>
</tr>
</thead>
<tbody>
<tr>
<td>data variables</td>
<td>text, numbers and dates copied directly from input sections (Chapter 5)</td>
</tr>
<tr>
<td>user variables</td>
<td>text, numbers and dates that are manipulated or calculated with functions and user defined expressions (Section 10.3)</td>
</tr>
<tr>
<td>global user variables</td>
<td>information derived from elsewhere in the project (Section 10.4)</td>
</tr>
<tr>
<td>system variables</td>
<td>project page numbers, date, time, file and user attributes and other system values (Section 10.5)</td>
</tr>
<tr>
<td>lookup variables</td>
<td>information derived from external databases (Section 10.6)</td>
</tr>
<tr>
<td>environment variables</td>
<td>information derived from the operating environment (Section 10.7)</td>
</tr>
</tbody>
</table>
9.1.4 Window display preferences

You can customize the way objects are edited and displayed by selecting the Tools\Preferences command. With the options displayed in the panes of the Preferences dialog, you can:

- always display the object Properties dialog whenever an object is created
- turn on and off rulers
- turn on and off outlines
- highlight text variables in red
- show variable names instead of values
- control how objects are displayed in selected and unselected sections

9.2 Creating Objects

To create an object:

1. From the Object toolbar, select the tool for the type of object you want to create.
2. Click-and-drag to create the initial position and size of the object.
   An object will be created using the default attributes (Section 9.4.4).

Object tools

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Insert static text Section 9.2.1.1</td>
</tr>
<tr>
<td></td>
<td>Insert static text containing variables Section 9.2.1.2</td>
</tr>
<tr>
<td>Variable Text</td>
<td>Insert a single variable as text Section 9.2.1.3</td>
</tr>
<tr>
<td>Subform</td>
<td>Insert an existing subform Section 9.2.2.2</td>
</tr>
<tr>
<td></td>
<td>Create subforms Section 9.2.2.4</td>
</tr>
<tr>
<td>Variable Subform</td>
<td>Use a variable to display different subforms Section 9.2.2.5</td>
</tr>
</tbody>
</table>
9.2 Creating Objects

9.2.1 Adding text to the page

There are two types of text objects that you can insert:

- text objects that can contain both static text and variables
- variable text objects that can contain only a single variable

Static text is the text that you type directly into the project from the keyboard, whereas variables can contain text, numbers and dates from the input file and other sources. For information on the types of variables you can use, see Section 9.1.3.

You can add text and variable text objects to the page in several ways:

By using the Object toolbar
- Insert static text (Section 9.2.1.1)
- Insert static text containing variables (Section 9.2.1.2)
- Insert a single variable as text (Section 9.2.1.3)

By copying all data variables from an input section
- Copy data variables from an input section (Section 9.2.1.4)

By drag-and-drop from the Input Window
- Drag-and-drop variables from the input window (Section 9.2.1.5)

9.2.1.1 Insert static text

1. In the Design Window, click on the section where you want to insert the text object.
2. Click the Text button ( ) in the Object toolbar.
   The cursor will change to the text symbol.
3. In the Design Window, click-and-drag the cursor to create a text box.
   A text box will be displayed in text edit mode.
4. Type in the text you want.
5. Click outside the text object to exit text edit mode.

For information on formatting and editing text objects, see Section 9.3.1 and 9.3.2.
9.2.1.2  **Insert static text containing variables**

1. In the Design Window, click on the section you want to place the text object in.
2. Click the Text button (,) in the Object toolbar. The cursor will change to the text symbol.
3. In the Design Window, click-and-drag the cursor to create a text box. A text box will be displayed in text edit mode.
4. Type the required text, and as required, click the **Insert Variable** button (,) on the Text toolbar. The **Insert Variable** dialog will be displayed.
5. From the **Variable** drop-down, select the required variable from the available sources. The **Variable** drop-down displays all currently available variables for that section. For information on making variables available from another source, see **Section 7.6.7**.
6. Click **OK**.
7. Continue typing or add further variables.
8. Click outside the object to exit text edit mode.

For information on formatting and editing text objects containing variables, see **Section 9.3.1** and **Section 9.3.2**.

9.2.1.3  **Insert a single variable as text**

1. In the Design Window, click on the section you want to place the variable text object in.
2. Click the Variable Text button (,) in the Object toolbar. The cursor will change to the variable text symbol.
3. In the Design Window, click-and-drag the cursor to create a text box. A variable text object is created and the **Variable Text Properties** dialog will be displayed.
4. From the **Variable** pane, click the **Variable** drop-down button and select the required variable from the available sources. The **Variable** drop-down displays all currently available variables for that section. For information on making variables available from another source, see **Section 7.6.7**.
5. Edit the object properties on the General (Section 9.4.3.1), Variable (Section 9.4.3.2) and Text (Section 9.4.3.3) panes as required.

6. Click OK.

To format a variable text object, see Section 9.3.3.

9.2.1.4 Copy data variables from an input section

Instead of inserting one variable at a time, you can copy all variables that have been defined in the corresponding section of the input file, or for the entire project. A variable text object will be created for each data variable that is not already copied to the project.

To copy variables from a section:

1. In the Design Window, click on the section corresponding to the input section where the variables are located.

2. Do one of the following:
   - From the Main menu, select Section ▶ Copy Data Variables.
   - Right-click on the section to display the shortcut menu and select Copy Data Variables.

To copy all variables from the project:

- From the Main menu, select Project ▶ Copy Data Variables.

Once the variables have been copied, you can reposition, resize and reformat as required on the page. To format a variable text object, see Section 9.3.3.

---

You can synchronize the display in the Input Window with selections made in the Design Window. For more information, see Section 12.1.7.

---

9.2.1.5 Drag-and-drop variables from the input window

You can create variable text objects by using the cursor to drag-and-drop data variables from the Input Window to the Design Window. To copy more than one data variable at a time, select the data variables while holding down the SHIFT key, or drag a box around a group of data variables, and then drag-and-drop the group into the Design Window.
For more information about selecting groups of objects, see "Section 9.5.1" and "9.5.2".

If the section you are dragging from is not a source for the section you are dragging to, you will be prompted to add the section as a source. When you confirm the operation, the **Section Properties** dialog will open at the **Source** tab, with the section added as a source. Click **OK** to add the source.

To format a variable text object, see "Section 9.3.3".

### 9.2.2 Adding subforms

If you intend to use the same text in several locations in the same project, or in several projects, you only need to create it once as a subform.

- About subforms ("Section 9.2.2.1")

**What do you want to do?**

- Insert an existing subform ("Section 9.2.2.2")
- Convert text files into subforms ("Section 9.2.2.3")
- Create subforms ("Section 9.2.2.4")
- Use a variable to display different subforms ("Section 9.2.2.5")
9.2 Creating Objects

9.2.2.1 About subforms
Subforms are commonly used for items that are repeated in the project or appear in multiple projects. When you modify that subform, all the occurrences where it is used will be automatically updated.

Subform Storage
By default, subforms are stored in the common project directory, however they can also be stored in the current project directory. For information on changing the common project directory and the project directory, see Section 12.1.1.

9.2.2.2 Insert an existing subform
1. Click the Subform tool ( ), located in the Object toolbar.
2. Position the cursor on the Design Window and click-and-drag to define the initial subform position. The Subform Properties dialog will be displayed.
3. In the Subform pane, click Select.
4. Select the subform file (.cff6) from the common project directory or project directory, and click Open.
5. Set other properties as required (Section 9.4).
6. Click OK. The subform will be displayed in the project.
9.2.2.3  Convert text files into subforms
You can convert one or more text files into subforms and place them on either the
design layer or the background layer.

To insert a text file as a subform:
1. Click the Subform tool ( ), located in the Object toolbar.
2. Position the cursor on the Design Window and click-and-drag to define the initial
   subform position.
   The Subform Properties dialog will be displayed.
3. In the Subform pane, click Import.
4. Navigate to and select the text file you want to use.
5. Click Open.
   The subform will be saved to the common project directory. Clear the Save to
   common form project directory checkbox to save to the project directory.
6. Set other properties as required (Section 9.4).
7. Click OK.
   The subform will be displayed in the project.

To convert multiple text files into subforms:
1. From the Main menu, select Tools>Create Subform>Import.
   The Import dialog will be displayed.
2. Either select or clear the Save to common form project directory checkbox. If
   cleared, the subforms will be saved to the project directory.
3. To add a prefix to each file as it is saved, select the Add prefix checkbox and
   type the prefix you want; the prefix will help you identify and manage the files in
   the project folders.
4. Navigate to the folder where the files are stored and select the images you want
   to convert.
   You can select multiple files in the same folder by holding down the CTRL key.
5. Click Open.
   Each of the selected files will be converted in turn and saved to the chosen
   directory.

The subforms can then be placed on the page individually (Section 9.2.2.2), or
displayed using a variable subform object (Section 9.2.2.5).
9.2 Creating Objects

9.2.2.4 Create subforms
Subforms are made of other Create!form Transform objects.

To create a subform:
1. Click the Subform tool ( ), located in the Object toolbar.
2. Position the cursor on the Design Window and click-and-drag to define the initial subform position.
   The Subform pane of the Subform Properties dialog will be displayed.
3. In the Subform pane, click Design New.
   The New Subform dialog will be displayed.
4. In the Name field, type the subform name and click OK.
5. Use the Create!form Designer tools to create the image or text you want.
   The subform file (.cff6) will be saved to the common project directory.
Close the subform window and return to your project.
The subform will be displayed in the project.

9.2.2.5 Use a variable to display different subforms
Use the Variable Subform tool ( ) from the Object toolbar when you want to place different subforms on the page depending on the value of a variable.

For information on creating subforms, see Section 9.2.2.

For example, a data variable, which stores the part number for each line item of a product catalog, can be used to descriptions for each part. If the part number values are, say, AT12P9, AY345W and BRG200F, the variable subform will look for the files AT12P9.cff6, AY345W.cff6 and BRG200F.cff6 in the common project directory, and display the subform corresponding to the part number in each occurrence of the data section.

To create a variable subform object:
1. Click the Variable Subform tool ( ), located in the Object toolbar.
2. Position the cursor on the Design Window and click-and-drag to define the initial height, width and location for the subform.
   The Variable Subform Properties dialog will be displayed.
3. From the Variable field, select the variable that will provide the subform name values.

4. Click OK.

The variable subform object will be created. If the value in the variable corresponds with the name of a subform, the subform will be displayed.

### 9.2.3 Copying objects from other resources

You can copy objects already created from existing projects and subforms.

**What do you want to do?**
- Copy objects from other projects (Section 9.2.3.1)

#### 9.2.3.1 Copy objects from other projects

To copy an object from another project, open the project and copy and paste the objects you want from one project document to the other.

---

**Note:** You can also apply the output design from one project to another, which will copy all the objects, sections, tables, variables and page styles defined in that project (Section 3.3.3).

### 9.3 Editing and Formatting Objects

When working interactively in the Design Window, a selection of frequently used formatting functions are available from the toolbars. The full range of formatting functions can be accessed from the object Properties dialog. The functions available, and how you find them, are described below.

**What do you want to do?**
- Change the display properties of any object (Section 9.3.1)
- Edit and format the contents of a text object (Section 9.3.2)
- Format a variable text object (Section 9.3.3)
- Change number, date and currency formats (Section 9.3.4)
- Edit the contents of a subform (Section 9.3.5)
9.3.1 Change the display properties of any object

The following display options can be accessed from the General pane of the object Properties dialog:

- **Origin** — displays the position of the object in the section
- **Size** — displays the horizontal and vertical dimensions of the object
- **Description** — used to identify the object in other views
- **Display condition** — applies conditions to when the object is visible

The properties specific to each type of object will be displayed on the Text, Subform, Variable, and Subform Variable panes of the object Properties dialog, depending on the type of object, or objects, selected. For information on how to display and edit these options, see Section 9.4.

You can also move, resize, copy and align objects interactively; for more information, see Section 9.5.

9.3.2 Edit and format the contents of a text object

If the text is static, you can edit and format individual characters and words; however for variable text, you can only apply formatting to the entire variable.

**To edit and format the contents of a text object:**

1. Select the text object.
2. Click again inside the text object to start text edit mode.
3. Select the characters or words you want to edit, replace or format.
4. Type the new text and use the formatting tools from the Text toolbar to format as required.
5. To format a variable in a text object, select the variable and use the formatting tools from the Text toolbar to format as required.
6. To edit other properties of a variable in a text object, double-click the variable. The Edit Variable dialog will be displayed from which you can change the variable name and edit the properties of the variable, which may include:
   - **Format as:** — display as text, number, date or currency (Section 9.3.4)
   - **Ignore blank lines** — removes blank lines from the text
   - **Strip leading spaces** — trims spaces from start of text string
   - **Strip trailing spaces** — trims spaces from end of text string
Chapter 9: Objects

- **Wrap (ignore end of line)** — wraps text over multiple lines
  For information on how to use these options, see Section 9.4.3.2.

7. Click OK to close the Edit Variable dialog and apply the changes.
8. Click outside the object to exit text edit mode.

**To change other text properties:**

Double-click the object to display the Text Properties dialog. The following options are available from the Text pane, and any changes will be applied to the object as a whole:

- **Alignment** — text alignment tools
- **Line spacing** — changes the spacing between lines of text
- **Kerning** — changes the spacing between characters
- **Format Style** — changes how currencies, dates and numbers are displayed

For information on how to use these options, see Section 9.4.3.3.

**9.3.3 Format a variable text object**

Any changes to formatting will be applied to the variable as a whole.

**To format a variable text object:**

1. Select the object.
2. Use the formatting tools from the Text toolbar to format as required.
3. To access the full range of formatting options, double-click the object to display the Variable Text Properties dialog.
4. Select the Text pane to display and edit these properties:
   - **Alignment** — text alignment tools
   - **Line spacing** — changes the spacing between lines of text
   - **Kerning** — changes the spacing between characters
   - **Format Style** — changes how currencies, dates and numbers are displayed
     For information on how to use the above options, see Section 9.4.3.3.
5. Select the Variable pane to display and edit these properties:
   - **Format as:** — display as text, number, date or currency (Section 9.3.4)
   - **Ignore blank lines** — removes blank lines from the text
   - **Strip leading spaces** — trims spaces from start of text string
9.3 Editing and Formatting Objects

- **Strip trailing spaces** — trims spaces from end of text string
- **Wrap (ignore end of line)** — wraps text over multiple lines
- **Keep paragraphs** — preserves paragraph structure of text string

For information on how to use the above options, see Section 9.4.3.2.

9.3.4 Change number, date and currency formats

You can apply customized format styles that control the way numbers, dates and currencies are displayed. For example whether currencies are displayed with a $ or £ symbol, or dates are displayed in month-day-year or day-month-year order. Different format styles can be applied to objects in the same project, allowing you to create output for offices or customers in multiple regions with one design project. For information on creating and managing format styles, see Section 6.1.3.

The following procedures explain how you can format variables as numbers, dates or currencies using a particular format style.

**To select the format style:**

1. Double-click the object to display the object **Properties** dialog.
2. Select the **Text** properties pane.
3. Select a style from the **Format Style** drop-down.
4. Click **OK**.

For more information on applying format styles, see Section 9.4.3.3.

**To format a variable text object as a number, date or currency:**

1. Double-click the object to display the object **Variable Text Properties** dialog.
2. Select the **Variable** properties pane.
3. Select a format from the **Format as:** drop-down.
   Selecting the Text format will leave the input data in its original format.
4. Click **OK**.

**To format a variable in a text object as a number, date or currency:**

1. Select the text object.
2. Click again inside the text object to start text edit mode.
3. Double-click the variable you want to edit. The **Edit Variable** dialog will be displayed.

4. Select a format from the **Format as:** drop-down. Selecting the Text format will leave the input data in its original format.

5. Click **OK**.

6. Click outside the object to exit text edit mode.

---

### 9.3.5 Edit the contents of a subform

**To edit a subform:**

1. Do one of the following:
   - Use the Selection tool to double-click on the subform to display the **Subform Properties** dialog. In the **Subform pane**, click **Edit**. The subform will be displayed.
   - Select **File** > **Open**. From the **Open** dialog, in the **Files of type:** drop-down, select **Subforms (*.cff6)**. Select the required file(s) and click **Open**. The subform will be displayed.

2. Use the Object tools to edit the subform.

3. Save and close the subform. The changes will be reflected in all projects that use the subform.

---

### 9.4 Setting Object Properties

**What do you want to do?**

- Display and edit object properties (**Section 9.4.1**)
- Edit the properties of multiple objects (**Section 9.4.2**)
- Set default properties (**Section 9.4.4**)

**What do you want to know about?**

- About property classes (**Section 9.4.3**)
9.4 Setting Object Properties

9.4.1 Display and edit object properties

The position, size, rotation, content, format and display properties of objects can be displayed and edited from the <object type> Properties dialog. When you edit and format objects interactively, the content of the object Properties dialog will be updated when the object properties are changed.

You can display the object Properties dialog from the Design Window in either the Graphics View or the Listing View, or from the Project Tree.

To display the object Properties dialog:

From any view, do one of the following:

- Double-click on the object.
- Right-click on the object and select Properties from the shortcut menu.
- Click on the object and select Edit Properties from the Main menu.
- Click on the object and select the Properties button ( ) from the Main toolbar.

Choose the method that best suits the way you work.

![Variable Text Properties Dialog](image)

**Figure 9-3:** The General pane of the object Properties dialog.

9.4.2 Edit the properties of multiple objects

You can edit the properties of multiple selected objects, including different object types, at the same time.
To edit the properties of multiple objects:

1. Select the objects you want to edit.
2. Click the Properties button ( ) from the Main toolbar. The object Properties dialog displays the properties of all selected objects.
3. Edit the properties from the dialog panes displayed.
4. Click OK to apply the changes.
   Properties that are common to more than one object will be applied to all objects that share those properties, and properties that are unique to one object will be applied only to that object.

9.4.3 About property classes

Objects can belong to several property classes, which can be common to other objects. Each property class will be displayed as a separate pane in the object Properties dialog.

The object Properties dialog displays the property classes for all of the selected objects. For example, if you simultaneously edit a text object and a variable subform, their properties will be displayed on the General, Text and Subform Variable property panes:

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Property Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General</td>
</tr>
<tr>
<td>Text</td>
<td>✔</td>
</tr>
<tr>
<td>Variable Text</td>
<td>✔ ✔</td>
</tr>
<tr>
<td>Subform</td>
<td>✔</td>
</tr>
<tr>
<td>Variable Subform</td>
<td>✔ ✔</td>
</tr>
</tbody>
</table>
9.4 Setting Object Properties

9.4.3.1 General properties
The General pane of the object Properties dialog displays the following options that are common to all objects:

Origin and Size
You can position objects relative to a section, or to the page. The position and size of the object will be displayed in units columns (characters) and rows (lines). Object position and size are controlled by the following properties:

• **Origin**
The object position relative to the section or page origin (1,1) will be displayed in the X and Y fields, measured from the top left corner of the section, or the page, to the top left corner of the object.

**Fix to page**
By default, the object is positioned relative to the section. Select this option to fix the position of the object on the page. Take care not to fix any object that may have more than one occurrence on a page, as it will be redrawn in the same position for every occurrence.

• **Size**
The object **Width** and **Height** will be displayed.

Scaling
For subforms, you have the option of scaling the size of the object along both the horizontal and the vertical axis. The value of the scale is relative to the original object size when first created. When the object is created, the scaling is set to 1. When you change the scaling value, the size of the object will be factored by the scaling value along each axis respectively. For example, if a subform has an initial size of 2" high by 3" wide, when the scale is changed to 2 on the vertical axis, and 0.5 on the horizontal axis, the size of the subform will change to 4" high by 1.5" wide.

Description
You can type a label in the **Description** field that will help identify the object in the Listing view mode of the Design Window and the Project Tree.

Display Condition
Objects can be set to display when specified conditions are met. For example, you can create a condition so that a text object will only display when a data variable has a particular value. For information on building conditions, see Section 11.2.1.
9.4.3.2 Variable properties
The variable used to control the content of variable text, variable subform and variable barcode objects will be displayed on the Variable pane of the object Properties dialog.

The following additional options are only displayed for variable text objects, and are also available from the Edit Variable dialog (Section 9.3.2):

Format as:
Select whether to treat the variable data as Number, Date or Currency. If you select Text, no format style will be applied.
For information on setting the format style, see Section 9.4.3.3.

Ignore blank lines
When selected, blank lines will be removed from the text.

Strip leading spaces
When selected, trims spaces from the start of the text string.

Strip trailing spaces
When selected, trims spaces from the end of the text string.

Wrap (ignore end of line)
When selected, wraps the text over multiple lines according to the object width. If a variable is inserted in a text object and the Wrap (ignore end of line) checkbox is clear, any text that follows the variable will start on a new line.

Keep paragraphs
When selected, preserves new line feeds within the text string.

9.4.3.3 Text properties
A selection of text formatting tools are also available from the Text toolbar (Section 4.1).

The Text pane of the object Properties dialog displays the following tools and options:
Alignment tools

For information on aligning objects on the page, see Section 9.5.8.

Line Spacing

When the Fixed checkbox is selected, the value in the Line spacing field is the spacing, in rows, between the bottom of one line and the bottom of the next line. A line spacing of “0” makes the lines overprint. A line spacing of “1” makes the lines print at one line per row.

When the Fixed checkbox is cleared, the value in the Line spacing field is the gap, in rows, between the bottom of one line and the top of the next line. The line spacing interval is measured in rows. A line spacing of “0” makes the lines print at one line per row. A line spacing of “1” adds one row between lines.

Kerning

Kerning is a typesetting term that defines the distance between characters, measured in columns. Kerning allows you to space text across a text box. A kerning of “1” inserts a character space between each letter. A kerning of “-1” makes the text overprint.

Format Style

You can apply different format styles to variables in text and variable text objects. Format styles determine how currencies, dates and numbers are displayed. When you apply a format style to an object, it applies to all variables contained in the object.

By default, all new text objects will be created with the current default format style. For information on setting the default format style for the project, see Section 6.1.3.

Select the Manage Formats option from the Format style drop list to create, edit, rename and delete format styles.
9.4.3.4 Subform properties
For information on creating and using subform objects, see Section 9.2.2. The following command buttons are displayed on the Subform pane of the Properties dialog:

Select
Displays the subform files available to be inserted in the project.

Import
Allows you to import one or more text files, and saves the selected files as subforms in the common project directory or the project directory.

Edit
Opens a subform for editing in Create!form Transform. Any changes made will be reflected in all projects that use the subform.

Design New
Creates a new subform for editing in Create!form Transform.

9.4.3.5 Subform variable properties

Import
Allows you to import one or more text files, and saves the selected files as subforms in the common project directory or the project directory.

9.4.4 Set default properties
The following command buttons are displayed on various panes of the object Properties dialog.

Set as Default
Select this button to use the current properties as the default for the current property class. The default properties will be applied when creating new objects, in all projects.

Apply Defaults
Select this button to apply the current default settings for this property class to the selected objects.
9.5 Selecting and Editing Objects in Graphics View

What do you want to do?
- Select objects (Section 9.5.1)
- Select hidden objects (Section 9.5.2)
- Edit objects on sections that are not displayed (Section 9.5.3)
- Move objects (Section 9.5.4)
- Resize objects (Section 9.5.5)
- Copy objects (Section 9.5.6)
- Align objects (Section 9.5.7)
- Advanced object selection (Section 9.5.8)

9.5.1 Select objects

Use the Selection tool ( ) to select one or more objects. (To unselect an object and select the Selection tool, press the SPACEBAR.)

To select multiple objects:
1. Select the first object you want to include in the selection.
2. Hold down the SHIFT key and click on the next object.
3. Continue to select additional objects in the same way.

To deselect an object:
1. Hold down the SHIFT key, and click on the selected object.
   The object will be unselected.

9.5.2 Select hidden objects

Objects can be hidden behind each other or difficult to identify because of their content. For example, variable objects may have no content. The following are methods for selecting and identifying these types of object. They all require that the section the object belongs to is selected:

Ctrl+A

Press the CTRL and A keys together to select all objects in the active section.
Chapter 9: Objects

Click-and-drag
If you know the approximate location of the object you are trying to locate, you can use the Selection tool to click-and-drag an area. All objects fully within the area will be selected.

Select Objects Shortcut
Use this method when selecting overlapping objects. Right-click the overlapping objects and choose Select Objects from the shortcut menu, then choose the appropriate object.

Using the Listing View or Project Tree
The Listing View (Section 4.1.1) and the Project Tree (Section 4.1.3) both offer a non-graphical list of all objects in a section. By selecting an object in either of these views, the same object is selected in the Graphics View.

9.5.3 Edit objects on sections that are not displayed
You can still view and edit the objects on sections that are not displayed. There can be several reasons why sections may not be displayed in the design:

- the input file does not contain an instance of a particular section, or
- the conditions for the display of an alternate section are not met, or
- continuation headers and footers do not occur

To edit the contents of a section that is not displayed:
1. In the Project Tree, right-click the section you want to edit, and select Edit Form from the shortcut menu.
   The section, and the objects it contains, will be displayed in a separate view pane.
2. Add, edit and delete objects as required.
3. When you have completed your edits, close the section edit pane by clicking the close button (×) at the top right corner of the view pane.

All sections, including all possible continuation headers and footers and alternate sections, are listed in the Project Tree.
9.5.4 Move objects

There are several methods of moving objects.

Use the cursor:

Use the Selection tool to select and drag an object to the desired location.

- You can constrain movement to increments of 45 degrees from the original location, by holding down the SHIFT key while dragging.
- Movements will snap to the nearest column and row position.

Use the arrow keys:

You can move any selected object(s) using the arrow keys (Appendix B).

Use the Move tool:

This method is useful where a precise movement is required.

1. Select the object.
2. From the Main toolbar, click the Move Objects button ( ) or select Edit\Move from the Main menu.
   The Move dialog will be displayed.
3. Specify the amount of horizontal and vertical movement that you want. Negative numbers move the object(s) left and up.

You can also:

- set the location of an object from the General pane of the Properties dialog (Section 9.4.3.1).

9.5.5 Resize objects

To resize one or more objects:

1. Select the object(s) to resize.
2. Position the cursor over one of the object’s sizing handles.
3. Click-and-drag outwards to increase the size or inwards to decrease the size.
4. Release the mouse button when the object(s) are the size required.
You can also:

- set the exact size of an object from the **General** pane of the **Properties** dialog (Section 9.4.3.1)
- use the arrow keys while holding down the SHIFT or CTRL keys to adjust the size of an object (Appendix B)
- make the width and height the same as a control object (Section 9.5.7).

---

**To resize the object(s) proportionally, hold down the SHIFT key while dragging a corner handle.**

---

### 9.5.6 Copy objects

**To copy one or more objects:**

1. Select the object(s) to copy.
2. From the Main menu, select **Edit** > **Copy**, or press CTRL+C.
3. From the Main menu, select **Edit** > **Paste**, or press CTRL+V.

   The object(s) will be pasted into the currently selected section.

   The new object will be pasted in front of, but fractionally offset from the original.

---

**To create a copy of an object and move it at the same time, hold down the CTRL button and “drag” a copy of the object away from the original.**

---

### 9.5.7 Align objects

The alignment tools enable you to accurately line up objects. The Align toolbar is positioned on the lower left side of the screen by default. In order to use the alignment tools, you must understand the concept of control objects.

**The control object**

The control object is the object that other objects move around, when you are performing any sort of alignment.

- If you’ve selected the objects using the shift and click method, the control object is the last one selected.
If you’ve selected the objects using the click-and-drag or select all method, the control object is the last object created.

Handles around the control object have a thicker black border than the handles on the other selected objects.

Alignment tools

The following table explains the alignment tools when used with fully selected objects. These tools will perform different functions when used with the Handle Selection tool (Section 9.5.8).

<table>
<thead>
<tr>
<th>Tool</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Left]</td>
<td>Left</td>
<td>Aligns the left sides of all objects with the left side of the control object.</td>
</tr>
<tr>
<td>![Horizontal Center]</td>
<td>Horizontal Center</td>
<td>Aligns the horizontal centers of all objects with the horizontal center of the control object.</td>
</tr>
<tr>
<td>![Right]</td>
<td>Right</td>
<td>Aligns the right sides of all objects with the right side of the control object.</td>
</tr>
<tr>
<td>![Top]</td>
<td>Top</td>
<td>Aligns the tops of all objects with the top of the control object.</td>
</tr>
<tr>
<td>![Vertical Center]</td>
<td>Vertical Center</td>
<td>Aligns the vertical centers of all objects with the vertical center of the control object.</td>
</tr>
<tr>
<td>![Bottom]</td>
<td>Bottom</td>
<td>Aligns the bottoms of all objects with the bottom of the control object.</td>
</tr>
<tr>
<td>![Left to Right]</td>
<td>Left to Right</td>
<td>Aligns the left sides of all objects with the right side of the control object.</td>
</tr>
<tr>
<td>![Right to Left]</td>
<td>Right to Left</td>
<td>Aligns the right sides of all objects with the left side of the control object.</td>
</tr>
<tr>
<td>![Top to Bottom]</td>
<td>Top to Bottom</td>
<td>Aligns the tops of all objects with the bottom of the control object.</td>
</tr>
<tr>
<td>![Bottom to Top]</td>
<td>Bottom to Top</td>
<td>Aligns the bottom of all objects with the top of the control object.</td>
</tr>
<tr>
<td>![Width]</td>
<td>Width</td>
<td>Resizes all objects to the width of the control object.</td>
</tr>
<tr>
<td>![Height]</td>
<td>Height</td>
<td>Resizes all objects to the height of the control object.</td>
</tr>
</tbody>
</table>

To align objects:

1. Select the objects you want to align.
2. Select the control object.
3. Click the appropriate tool in the Align toolbar. The selected objects will be aligned with the control object.

### 9.5.8 Advanced object selection

The Handle Selection tool is an advanced feature that enables you to simultaneously align and re-size objects. By allowing you to select some object handles and not others, the Handle Selection tool lets you dictate which parts of an object are static and which parts can be moved or re-sized.

When used in conjunction with multiple objects, the Handle Selection tool allows you to fix the spatial relationship between non-selected handles on different objects.

**Anchor points**

When a handle is selected, the handle diagonally opposite the selected handle becomes the anchor point. Anchor points are fixed in position.

![Anchor point](image)

*Figure 9-4: A box object with one handle selected.*

When two handles are selected, the opposite two handles become anchor points.

![Anchor points](image)

*Figure 9-5: A box object with two selected handles.*
When a line object is selected, two handles are positioned on the perimeter of the object. When you select one handle, the other handle becomes the anchor point.

![Diagram showing anchor point and selected point](image)

*Figure 9-6: A diagonal line object with one handle selected.*

**To select handles:**

1. Activate the Handle Selection tool ( ).
2. Position the cursor on an object’s handle and click.
3. To select other handles (on the same or different objects), hold down the SHIFT key and click on the required handles. To select multiple handles at once, you can click-and-drag an area.

The handles of multiple objects become selected as shown below.

![Selection containing two handles from one object and one handle from another object](image)

*Figure 9-7: A selection containing two handles from one object and one handle from another object.*

**To deselect handles:**

1. Using the Handle Selection tool ( ), hold down the SHIFT key and click the selected handle you want to deselect.

**To re-size one object while moving another:**

1. Activate the Handle Selection tool ( ).
2. Partially select handles of the object to be resized.
3. With the SHIFT key held down, click in the middle of the second object to select the entire object.
   For example:

   ![Selected handles and an entire object.](image)

   Figure 9-8: Selected handles and an entire object.

   Both objects can now be manipulated simultaneously.

4. Click-and-drag downwards to increase the size of one object while moving the second object.

**To align handles of objects:**

When individual handles are selected with entire objects or handles on other objects, the Alignment toolbar may be used to align selected handles. Objects can be automatically resized and aligned with the edge of a control object.

1. Select the handles of the objects you wish to align.
2. Select the control object last ([Section 9.5.7](#)).
3. Click one of the tools on the alignment toolbar.
   For example:

   ![Aligning object handles using the Top tool on the Alignment toolbar.](image)

   Figure 9-9: Aligning object handles using the Top tool on the Alignment toolbar.
Variables

You can use variables to control the content of the project, either as objects on a page, or as arguments in expressions and conditions that you can use to control the display of objects, sections and pages.

The following topics are covered:

- About Variables
- Data Variables and Derived Variables
- User Variables
- Global User Variables
- System Variables
- Lookup Variables
- Environment Variables
- Job Ticket Variables
10.1 About Variables

Variables can be used to create text objects, or as arguments in conditions and expressions. There are several different types of variables that you can use in a project; the type of variable you use will depend on where you want to source your data and how it will be applied in the project:

- Data Variables and Derived Variables (Section 10.2)
- User Variables (Section 10.3)
- Global User Variables (Section 10.4)
- System Variables (Section 10.5)
- Lookup Variables (Section 10.6)
- Environment Variables (Section 10.7)
- Job Ticket Variables (Section 10.8)

10.2 Data Variables and Derived Variables

Data variables and derived variables are created in the input design (Chapter 5). To view the data variables defined in a section, select Section►Data Variables from the Main menu.

10.3 User Variables

User Variables are defined by users and derive their value from functions and expressions using other variables as arguments. User variables are created in and belong to a section. User variables can only derive values from data in the source of the section. They can be used for a wide variety of tasks.
Example

You might create a user variable to display a post-tax total. The user variable can take the pre-tax total (a value obtained by a data variable), and multiply it by the sales tax amount.

<table>
<thead>
<tr>
<th></th>
<th>Net Value:</th>
<th>Tax:</th>
<th>Total:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data variable</td>
<td>£5,000.00</td>
<td>10%</td>
<td>£5,500.00</td>
</tr>
<tr>
<td></td>
<td>£26,002.00</td>
<td>10%</td>
<td>£28,602.20</td>
</tr>
</tbody>
</table>

User variable that combines the value of the data variable, with the constant tax rate of 10%.

Figure 10-1: An example of how user variables can be used.

User variables can be used in other sections, provided the section it belongs to is a source for the section where it will be used (Section 7.6.7).

To create a user variable:

1. In the Design Window, select the section you want the user variable to be available in/belong to.
2. From the Main menu, select Section User Variables. The User Variables dialog will be displayed.
3. If required, click the Add Row button ( ). A new variable is created.
4. To create a condition, click the Condition button ( ) in the Condition column and see Section 11.2 for information on creating conditions.
5. In the Variable Name column, type a name for the user variable.
6. Click the Expression button ( ) to create the expression that will generate the variable value. The Expression Builder will be displayed. For information on building expressions, see Section 11.1.2.
7. Click OK. Once you have created a user variable, you can use it in text objects, variable text objects or to build other expressions or conditions.
10.4 Global User Variables

Global user variables enable you to create a variety of dynamic fields and use them in any section in the project. Global user variables have an initial value and are passed sequentially from section to section. In each section they can have their value changed.

Global user variables are typically used in conjunction with user variables. Global user variables can provide values for user variables, or derive values from user variables.

When to use global variables

Global user variables can be used in the following types of task:

- To perform calculations that involves data from unrelated sections in the project.
- For conditional based decision making that involves data from an unrelated section in the project.

To create a global user variable:

1. From the Main menu select Project ► Global User Variables. The Global User Variables dialog will be displayed.
2. Click the Add Row button.
3. In the Variable Name column, type a name.
4. In the Initial Value column, type an initial value.
5. In the Reset column, select whether you want the global user variable reset every page, set or document.
6. Click OK.

Global variables can be used in text objects, variable text objects or in conditions and expressions.
10.5 System Variables

Create!form Transform provides pre-defined variables that you can use to determine information about the project and the operating environment. The following table describes the system variables available in the input and output of the design:

<table>
<thead>
<tr>
<th>System Variable</th>
<th>Description</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sys.PreviousSection</td>
<td>Name of previous section.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Sys.SetPageNum</td>
<td>Current page number in the current set.</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Sys.PageNum</td>
<td>Collated page number sequence in Next Copy projects.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Sys.SetNumPages</td>
<td>Total number of pages in the current set.</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Sys.Today</td>
<td>Current date value supplied by the system.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Sys.Time</td>
<td>Time of merge supplied by the system.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Sys.Hostname</td>
<td>Name of host server where queue resides.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Sys.OSVersion</td>
<td>Version of operating system running on host.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Sys.SpoolID</td>
<td>Spool file ID on the merge queue.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Sys.DocName</td>
<td>Name of spool file.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Sys.QueueName</td>
<td>Name of print queue where merge occurs.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Sys.DeviceName</td>
<td>Name of destination port, queue or device.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Sys.ProjectName</td>
<td>Name of current project.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Sys.ProductVersion</td>
<td>Details about Create!form Server installed on host.</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

System variables can be displayed as variable text objects or used as arguments in expressions and conditions.

Examples:
- This expression tests for odd or even numbered pages in a document:
  \[ \text{round}(\text{Sys.DocPageNum}/2,0)*2-\text{Sys.DocPageNum} \]
- This expression tests if the date stored in \text{InvDate} is in the current month:
  \[ \text{month(InvDate)}==\text{month(Sys.Today)} \]
10.6 Lookup Variables

Lookup variables retrieve values from external databases. For example you can use lookup variables to retrieve a customer telephone number from a database. Lookup variables can be inserted directly into the project or used as part of an expression or condition.

Lookup variables are created in and belong to a section. They can be used in other sections, provided the section it belongs to is a source for the section where it will be used.

Requirements

To use lookup variables you require a DSN connection to an ODBC database. To enable production printing, you must also create a permanent connection between the e-forms server and the database(s). For more information see the Create!form Server User Guide.

To create a lookup variable:

1. From the Create!form Transform main menu, select Section►Lookup Variables.
   The Lookup Variables dialog will be displayed.

![Screenshot of the Lookup Variables dialog](image.png)

*Figure 10-2: The Lookup Variables dialog.*
2. From the drop-down list under the **Database (DSN)** field, select the DSN for the database you want to retrieve data from, or click **New** to create a new DSN. Setup procedures for databases will vary.

3. If multiple tables/views exist in the database, select the appropriate one from the **Table/view** drop-down.

4. In the **Select records where** field, define a query by selecting:
   - the **Database Field** from the database table that will be searched *in* and;
   - the **Variable** from the project containing the value that will be searched *with*. Ensure the query will only match with one record. If you need to add further qualifiers, clicking the Add Row button (?) and select further fields and variables.

5. In the **Assign to variables** field, define the response to the above query by:
   - typing the **Variable Name** in the project to assign *to* and;
   - selecting the **Database Field** from the database table to assign *from*. It is possible to return more than one response from that particular record in the database by clicking the Add Row button and defining further Variable Names and Fields.

6. By clicking the **Refresh** button at the bottom right of the dialog the returned value(s) from the lookup query will be displayed in the **Content** column.

---

**10.7 Environment Variables**

Environment variables are common to all projects in the production environment. Environment variables may be the same for all output destinations, or can be changed for individual print queues. Typical uses of environment variables include setting values that are used in all projects, like a company name, or using different page styles in the same project depending on whether the destination is a printer, fax, email or archive server port.

**Storage**

Environment variables are stored in the `EnvironmentVars.data` file in the `SystemResource` folder, which will be sent to the e-forms server when the project is published.
To create an environment variable:

1. From the Main menu, select **Tools > Environment Variables**. The **Environment Variables** dialog will be displayed.
2. To create a new environment variable, click the Add Row button.
3. Type the name of the environment variable in the **Name** field.
4. Type the value of the environment variable in the **Value** field.
5. Click **OK**.

### 10.8 Job Ticket Variables

All the job ticket values passed through with the spool file, or added by the e-forms server are available wherever system variables are available in the design. The job ticket variables available to a project must be defined in the following configuration files found in the Win32User folder:

- Text, CSV and XML input files: HeaderParams.cfg
- JDE PDF input files: JDEHeaderParams.cfg

You can edit these files, and add variables in a text editor, e.g. Notepad. Any changes you make to configuration files on the design workstation will be transferred to the e-forms server with the project (see [Section 3.9](#)).

---

*If you edit the configuration file, you must restart the program.*
Expressions and Conditions

You can create expressions and conditions to combine and manipulate text, calculate values, summarize data in tables and test when objects, sections and pages should be displayed.

The following topics are covered:

- Expressions
- Conditions
- Building Blocks
- Syntax and Order of Precedence Rules
- Summarizing Data in Headers and Footers
11.1 Expressions

What do you want to know about?
- Building Expressions (Section 11.1.1)
- Using the Expression Builder (Section 11.1.2)
- Navigating and Editing Expressions (Section 11.1.3)

11.1.1 Building Expressions

Expressions are built using the Expression Builder dialog (referred to as the Expression Builder), which is opened whenever you click the \( \sum \) button from either the Derived Variables dialog (input) or the User Variables dialog (output). The Expression Builder provides a range of operators and functions used to build expressions that evaluate values, called arguments, in a specific order, or syntax.

Expressions are evaluated from left to right subject to the syntax and order of precedence rules detailed in Section 11.4.

11.1.2 Using the Expression Builder

You must build an expression in a particular order. Generally you will follow the list boxes from left to right. The operators, functions and variables available to you are context sensitive. For example, when you select a variable with a numeric value, only functions that can process numbers will be available to you.

The following procedure explains how to build a simple expression to add the text “Inc.” to a user variable called “Company_Name”.

1. From the Main menu, select User Variables.
2. From the User Variables dialog, in the Variable Name field, type the name of the variable you want to create.
3. Click \( \sum \) to open the Expression Builder.
4. From the Expression Builder, click concat in the Function section.
5. In the Variable section, expand and select the user variable.
6. Click on the Edit Value button ( ) and type the text you want to add to the user variable.
   The Expression Builder will display: \( \text{concat(Company\_Name, " Inc." )} \)
7. Click OK to close the Expression Builder.
11.2 Conditions

When you close the **Expression Builder**, you will be returned to the **User Variables** dialog where the expression will be displayed in the **Value** field, and the value the expression returns in the current section will be displayed in the **Content** field. The same procedure applies when building expressions from the **Derived Variables** dialog.

To enable direct editing of the expression in the dialog that calls the Expression Builder, refer to [Section 12.1.1](#).

### 11.1.3 Navigating and Editing Expressions

- To move around the expression, use the arrow buttons, keyboard arrow keys and the cursor.
- To expand or reduce the selection, use the **Level Up** button (↑) or the **Level Down** button (↓).
- To add text or number values, use the **Edit Value** button (A).
- To delete parts of an expression, select the part of the expression to be deleted and click the **Delete** button (X).
- To enclose parts of an expression in parentheses, select the part of the expression to be enclosed and click the **Group** button ({}).

You can also enable direct editing of the expression in the dialog that calls the Expression Builder ([Section 12.1.1](#)).

### 11.2 Conditions

A condition is a special type of expression which produces a **TRUE** or **FALSE** value. Conditions allow you to vary the output of the project, depending on data within the input file. For more information about the ways you can use conditions, see the usage examples below.

#### Usage examples

Conditions can be used to determine:

- when input sections are created ([Section 5.5.10.2](#) and [5.5.11.2](#))
- when sets are created ([Section 5.5.5](#))
- when sections are displayed ([Section 7.2.1](#))
- which section will be displayed, when using alternates ([Section 7.4](#))
- when objects are displayed ([Section 9.4.3.1](#))
- when defining derived values
- when calculating values or manipulating data in user variables
Chapter 11: Expressions and Conditions

What do you want to know about?
- Building Conditions (Section 11.2.1)
- Using Expressions as Conditions (Section 11.2.2)

11.2.1 Building Conditions

The **Condition Builder** is used to build and edit expressions in much the same fashion as the **Expression Builder** (Section 11.1). The Condition Builder can be opened from numerous dialogs, either by clicking the Edit button next to a condition field, or by clicking the button.

The following procedure explains how to build a simple condition to test whether the user variable called “Item_type” contains the word “new”.

1. From the **Condition Builder**, click **contains** in the **Function** section.
2. In the **Variable** section, expand and select the user variable.
3. Click on the Edit Value button ( ) and type “new”. The **Condition Builder** will display: `contains(Item_type,"new")`
4. Click **OK** to close the **Condition Builder**.

When you close the **Condition Builder**, you will be returned to the originating dialog where the expression will be displayed. To enable direct editing of the expression in the dialog that calls the Condition Builder, refer to Section 12.1.1.

11.2.2 Using Expressions as Conditions

Any expression that evaluates to a non-zero value when true can be used as a condition. For example, the expression \( A - B \) is equivalent to the condition \( A! = B \) as both expressions return a zero value when \( A \) equals \( B \) and a non-zero value when \( A \) is not equal to \( B \).

11.3 Building Blocks

The **Expression Builder** and **Condition Builder** dialogs contain list boxes containing the building blocks from which you can build an expression. The content of these dialogs is context sensitive; the building blocks displayed or available will depend on the task you are performing:

- Boolean Operators (Section 11.3.1)
- Operators (Section 11.3.2)
- Functions (Section 11.3.3)
11.3 Building Blocks

- Variables (Section 11.3.4)
- System Conditions (Section 11.3.5)

11.3.1 Boolean Operators

Boolean operators are only available in the Condition Builder. The syntax of an expression involving boolean operators begins with the first argument, followed by the operator, and the second argument, separated by spaces:

```plaintext
condition1 OPERATOR condition2
```

Arguments can be other user defined conditions or system conditions. The following table describes the boolean operators available.

<table>
<thead>
<tr>
<th>Boolean operator</th>
<th>Meaning</th>
<th>Syntax and usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>logical AND (intersection)</td>
<td>condition1 AND condition2 Returns TRUE if both condition1 and condition2 are true. Example: contains(Price, &quot;.&quot;) AND contains(Price, &quot;0&quot;) returns TRUE if the data variable Price contains both &quot;.&quot; and &quot;0&quot;.</td>
</tr>
<tr>
<td>OR</td>
<td>logical OR (union)</td>
<td>condition1 OR condition2 Returns TRUE if either condition1 or condition2 are true. Example: Sys.FirstPageOfDoc OR Sys.FirstPageOfSet returns TRUE on the first page of the document and the first page of each set.</td>
</tr>
</tbody>
</table>

Other boolean operations can be performed by combining AND and OR operators in expressions with the negation operator NOT (Section 11.3.2); for example, a boolean difference operation is performed by the expression:

```plaintext
NOT condition1 AND condition2
```

11.3.2 Operators

The usual syntax of an expression involving operators begins with the first argument, followed by the operator, and the second argument, with or without separating spaces:

```plaintext
argument1 OPERATOR argument2
```
The negation operators are the only exceptions to this rule, as they precede a single argument. Arguments can be numbers, text strings, variables or other functions or expressions. The following table describes the operators available.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Syntax and usage</th>
</tr>
</thead>
</table>
| ==       | equal to           | A==B  
Compares the values of two numbers or two strings and returns TRUE if equal. Case sensitive when comparing strings. |
| !=       | not equal to       | A!=B  
Compares the values of two numbers or two strings and returns TRUE if not equal. Case sensitive when comparing strings. |
| >        | greater than       | A>B  
Compares the values of two numbers and returns TRUE if A is greater than B. |
| <        | less than          | A<B  
Compares the values of two numbers and returns TRUE if A is less than B. |
| >=       | greater than or equal to | A>=B  
Compares the values of two numbers and returns TRUE if A is greater than or equal to B. |
| <=       | less than or equal to | A<=B  
Compares the values of two numbers and returns TRUE if A is less than or equal to B. |
| *        | multiply           | A*B  
Multiplies two numbers. |
| /        | divide             | A/B  
Divides A by B. |
| +        | add                | A+B  
Adds two numbers, or concatenates two strings. |
| -        | subtract           | A-B  
Subtracts B from A. |
| NOT      | logical negation   | NOT condition  
Reverses the logical value of an expression. |
| - (neg)  | arithmetic negation | -B  
Reverses the sign of a numeric value, i.e. -B=-1*B |

For the syntax rules governing the evaluation of expressions using parentheses, multiple operators and mixed data types, see Section 11.4.
11.3 Building Blocks

11.3.3 Functions

The syntax of a function begins with the function name, followed by an opening parenthesis, the arguments for the function separated by commas, and a closing parenthesis.

function(argument1,argument2,...)

Arguments can be numbers, text strings, variables, functions and expressions. The following tables describe the functions available.

Logical functions

Logical functions can be used as conditions returning a TRUE or FALSE value. When used as arguments in expressions, logical functions return a value of 1 if true, and 0 if false.

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax and usage</th>
</tr>
</thead>
</table>
| contains  | contains(string,find_text)  
                    Returns TRUE if string contains find_text.  
                    Example: contains("Heath;Adele;Mrs;;;;OK;;;;OK") = TRUE |
| exists    | exists(name)      
                    Returns TRUE if the variable name exists. |
| isnumeric | isnumeric(name)   
                    Returns TRUE if the variable name contains at least one numeral and any of the characters , . ( ) + - $ % or space. The test will fail if the variable contains any other character.  
                    Examples:  
                    isnumeric("$ 1,234.56") = TRUE  
                    isnumeric("$US1,234.56") = FALSE |

Text and numeric functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax and usage</th>
</tr>
</thead>
</table>
| aswords   | aswords(number,option)  
                    Converts a number into words. Specifying "a" in the second argument will generate a number phrase with “and” e.g. “Five hundred and twenty seven”  
                    Examples:  
                    aswords("123","") = "One hundred twenty three"  
                    aswords("123","a") = "One hundred and twenty three" |
<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax and usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>at</strong></td>
<td><code>at(row, column, length)</code>&lt;br&gt;Returns a string of a specified length, starting at the specified row and column position in the current input page.&lt;br&gt;Example: if row 5 of the input page contains the letters of the alphabet in columns 1 to 26, then<code>at(5,12,3) = &quot;LMN&quot;</code></td>
</tr>
<tr>
<td><strong>concat</strong></td>
<td><code>concat(string1,string2)</code>&lt;br&gt;Joins two text strings.&lt;br&gt;Example: <code>concat(&quot;Adele &quot;, &quot;Heath&quot;) = &quot;Adele Heath&quot;</code></td>
</tr>
<tr>
<td><strong>find</strong></td>
<td><code>find(string,find_text,start_char)</code>&lt;br&gt;Finds one text string within another text string, starting from character position <code>start_char</code>, and returns the starting position. The first character in <code>string</code> is position 1. Returns 0 if <code>find_text</code> is not found in <code>string</code>. &lt;br&gt;Example: <code>find(&quot;Mrs Adele Heath&quot;,&quot; &quot;,5) = 10</code></td>
</tr>
<tr>
<td><strong>length</strong></td>
<td><code>length(string)</code>&lt;br&gt;Returns the length of a text string, including spaces.&lt;br&gt;Example: <code>length(&quot;Mrs Adele Heath&quot;) = 15</code></td>
</tr>
<tr>
<td><strong>lower</strong></td>
<td><code>lower(string)</code>&lt;br&gt;Converts a text string to lower case.</td>
</tr>
<tr>
<td><strong>ltrim</strong></td>
<td><code>ltrim(string)</code>&lt;br&gt;Trims leading spaces from a text string.&lt;br&gt;Example: <code>ltrim(&quot; left trim &quot;) = &quot;left trim &quot;</code></td>
</tr>
<tr>
<td><strong>mod</strong></td>
<td><code>mod(number,base)</code>&lt;br&gt;Returns the remainder after <code>number</code> is divided by <code>base</code>. &lt;br&gt;Example: <code>mod(51,7) = 2</code></td>
</tr>
<tr>
<td><strong>num</strong></td>
<td><code>num(string)</code>&lt;br&gt;Converts a text string into a number. &lt;br&gt;Examples:&lt;br&gt;num(&quot;12ab3&quot;) = 123&lt;br&gt;num(&quot;18+9/3&quot;) = 1893&lt;br&gt;num(&quot;nine&quot;) = 0</td>
</tr>
<tr>
<td><strong>pad</strong></td>
<td><code>pad(string,length)</code>&lt;br&gt;If <code>string</code> is shorter than <code>length</code>, extends <code>string</code> to <code>length</code> with spaces.&lt;br&gt;If <code>string</code> is longer than <code>length</code>, truncates <code>string</code> to <code>length</code>. &lt;br&gt;Examples:&lt;br&gt;<code>pad(&quot;abc&quot;,6) = &quot;abc   &quot;</code>&lt;br&gt;<code>pad(&quot;abc&quot;,1) = &quot;a&quot;</code>&lt;br&gt;<code>pad(&quot;abc&quot;,-2) = &quot;abc&quot;</code>&lt;br&gt;<code>pad(&quot;abc&quot;,&quot;def&quot;) = &quot;&quot;</code></td>
</tr>
<tr>
<td><strong>proper</strong></td>
<td><code>proper(string)</code>&lt;br&gt;Capitalizes the first letter of each word in a text string.</td>
</tr>
<tr>
<td>Function</td>
<td>Syntax and usage</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>previous</td>
<td>previous(var_name) Returns the value of variable var_name in the previous occurrence of the current section.</td>
</tr>
<tr>
<td>previousat</td>
<td>previousat(row,column,length) Returns a string of a specified length, starting at the specified row and column position in the previous input page.</td>
</tr>
<tr>
<td>retrieve</td>
<td>retrieve(string,ordinal_position,delimiter_char) Extracts text from delimited fields in a text string. When the space character is used as a delimiter, leading spaces are ignored and sequential spaces are treated as a single delimiter. Examples: retrieve(&quot;Abc;De;F;;Ghij;;&quot;,6,&quot;;&quot;) = &quot;Ghij&quot; retrieve(&quot; Abc De F&quot;,2,&quot; &quot;) = &quot;De&quot; Use the following arguments when specifying these delimiters: ArgumentDelimiter \n new line \t tab &quot; double quotation mark (&quot;) \ \backslash () Example: retrieve(var,1,&quot;\t&quot;) returns the value before the first tab in the text variable var.</td>
</tr>
<tr>
<td>rfind</td>
<td>rfind(string,find_text,start_char) Finds one text string within another text string by searching from the right to the left, starting from character position start_char, and returns the starting position. The first character in string is position 1. Position 0 specifies the last character in string. Returns 0 if find_text is not found in string. Example: rfind(&quot;Heath,Adele,Mrs&quot;,&quot;,&quot;0) = 12</td>
</tr>
<tr>
<td>round</td>
<td>round(number,num_digits) Rounds a number to a specified number of decimal places. Examples: round(1234.567,2) = 1234.57 round(1234.56,0) = 1234 round(1234.56,-2) = 1200</td>
</tr>
<tr>
<td>rtrim</td>
<td>rtrim(string) Trims trailing spaces from a text string. Example: rtrim(&quot; right trim &quot;) = &quot; right trim&quot;</td>
</tr>
<tr>
<td>sentence</td>
<td>sentence(string) Converts a text string to sentence case.</td>
</tr>
</tbody>
</table>
# Chapter 11: Expressions and Conditions

## Date functions

The result of a date function depends on the current date order setting ([Section 5.5.6.1](#)).

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax and usage</th>
</tr>
</thead>
</table>
| **date** | `date(year,month,day)`  
Returns a date value from arguments representing the year, month and day.  
Values of day and month outside the normal range are treated on the basis that month=0 is the last month of the previous year, and day=0 is the last day of the previous month.  
The following examples assume a date order setting of d-m-y:  
`date("2003","4","19")` = 19-4-2003  
`date(2000,0,0)` = 30-11-1999  
`date(2004,-1,32)` = 2-12-2003 |
| **day** | `day(date)`  
Returns the day of the month (a number between 1 and 31) from a date value.  
Example: `day("19/4/03")` = 19 when date order is d-m-y |
| **month** | `month(date)`  
Returns the month of the year (a number between 1 and 12) from a date value.  
Example: `month("19-4-2003")` = 4 when date order is d-m-y |
| **year** | `year(date)`  
Returns the year (a number between 1970 and 2069) from a date value.  
Example: `year("19 04 03")` = 2003 when date order is d-m-y |
11.3 Building Blocks

Any non-numeric parsing character is permitted in a string representing a date value, for example, the strings "19 4 03" and "19.04.2003" will return the same value when used as arguments in a date function.

For information on using the addition and subtraction operators with date values, see Section 11.4.

Statistical functions

Table functions are defined in a header or footer, and perform calculations on variables that occur within the part of the document associated with the header or footer. For information on how to use statistical functions to summarize data in tables, pages and sets, see Section 11.5.

<table>
<thead>
<tr>
<th>Function</th>
<th>Syntax and usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>count(name)</td>
</tr>
<tr>
<td>max</td>
<td>max(name)</td>
</tr>
<tr>
<td>min</td>
<td>min(name)</td>
</tr>
<tr>
<td>sum</td>
<td>sum(name)</td>
</tr>
</tbody>
</table>

Counts the number of times the variable occurs within part of a document.
Finds the maximum value of a variable within part of a document.
Finds the minimum value of a variable within part of a document.
Sums a variable within part of a document.

11.3.4 Variables

The types of variables that can be used in conditions as arguments of operators and functions are summarized below:

<table>
<thead>
<tr>
<th>Variable Type</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>data variable</td>
<td>input data (Section 5.5.3)</td>
</tr>
<tr>
<td>user variable</td>
<td>user defined expressions relating to section data (Section 10.3)</td>
</tr>
<tr>
<td>global user variable</td>
<td>user defined expressions relating to other sections or the whole project (Section 10.4)</td>
</tr>
<tr>
<td>system variable</td>
<td>system supplied data (Section 10.5)</td>
</tr>
<tr>
<td>lookup variable</td>
<td>data from an external database (Section 10.6)</td>
</tr>
<tr>
<td>environment variable</td>
<td>data that applies to all projects (Section 10.7)</td>
</tr>
<tr>
<td>job ticket variable</td>
<td>job file attributes (Section 10.8)</td>
</tr>
</tbody>
</table>
The variables that are available to you at any time, will depend on the type of expression you are building, and where you are building it from. User variables and data variables are available only in the section they are defined in.

### 11.3.5 System Conditions

Create!form Transform provides pre-defined conditions that you can use to:

- conditionally apply an input section (Text and JDE PDF)
- conditionally display text objects or variable text objects
- conditionally define derived variables (Text and JDE PDF)

System conditions will only be displayed in the **Condition Builder** when you are performing one of the above tasks. The following table describes the system conditions available.

<table>
<thead>
<tr>
<th>Pre-defined conditions</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sys.FirstPageOfDoc</td>
<td>Returns TRUE when the current section is on the first page of the project.</td>
</tr>
<tr>
<td>Sys.ContPageOfDoc</td>
<td>Returns TRUE when the current section is not on the first page or the last page of the project.</td>
</tr>
<tr>
<td>Sys.LastPageOfDoc</td>
<td>Returns TRUE when the current section is on the last page of the project.</td>
</tr>
<tr>
<td>Sys.FirstPageOfSet</td>
<td>Returns TRUE when the current section is on the first page of the set for the project.</td>
</tr>
<tr>
<td>Sys.ContPageOfSet</td>
<td>Returns TRUE when the current section is not on the first page or the last page of the set for the project.</td>
</tr>
<tr>
<td>Sys.LastPageOfSet</td>
<td>Returns TRUE when the current section is on the last page of the set for the project.</td>
</tr>
</tbody>
</table>

### 11.4 Syntax and Order of Precedence Rules

Expressions are evaluated using a specific set of rules, or syntax, that depends on the operators and functions you use in the expression. The **Condition Builder** and **Expression Builder** dialogs will assist you in building expressions with a valid syntax; however you must follow a few basic rules to ensure the expression will perform the way you expect:

- Mixed Data Types ([Section 11.4.1](#))
- Addition Operator ([Section 11.4.2](#))
- Subtraction Operator ([Section 11.4.3](#))
- Comparing Text Strings ([Section 11.4.4](#))
11.4 Syntax and Order of Precedence Rules

- Order of Precedence (Section 11.4.5)
- Nested Functions (Section 11.4.6)
- Parsing (Section 11.4.7)
- Names of Variables (Section 11.4.8)

11.4.1 Mixed Data Types

Avoid mixing text, numeric and date values in the same expression as this can produce unexpected results. All variable data is stored as a string, but will be interpreted as a number in an arithmetic operation. For example:

"-2"*"3" evaluates to the number -6
2*"3A4" evaluates to the number 68
2*"3-4" evaluates to the number 68
2*"3A4-" evaluates to the number -68
"two"*"three" evaluates to the number 0

11.4.2 Addition Operator

The addition operator (+) is a special case. The rules governing the addition of numeric, text and date arguments are:

- if both arguments of an addition are numeric, an addition is performed. For example: 1+1 evaluates to the number 2.
- if either argument of an addition is a string, the arguments are concatenated. For example: 1+"1" evaluates to the text string "11".
- if one argument is a date and the other is a number, a date addition is performed. For example: if Today is a date variable (m-d-y) with a value of 9-6-2004, the expression Today+1 evaluates to the date value 9-7-2004, whereas the expression "9/6/2004"+1 evaluates to the string "9/6/20041".

11.4.3 Subtraction Operator

The subtraction operator (-) is a special case. The rules governing the subtraction of numeric, text and date arguments are:

- if both arguments of a subtraction are either numeric or text, a subtraction is performed. For example: "a5"-2 evaluates to the number 3.
• if the first argument is a date and the second is a number, a date subtraction is performed. For example: if Today is a date variable (m-d-y) with a value of 9-6-2004, the expression today-1 evaluates to the date value 9-5-2004, whereas the expression "9/6/2004"-1 evaluates to the number 962003.
• if both arguments are dates, a date difference is performed. For example: Sys.Today-InvDate evaluates to the number of days difference between today’s date and the date variable InvDate, whereas the expression "9/6/2004"-"9/6/2003" evaluates to the number 1.

11.4.4 Comparing Text Strings

Avoid using the comparison operators <=, >=, <, > to compare text strings as this can produce unexpected results. The rules governing comparisons of numeric, text and date arguments are:
• if either argument of a comparison is a number, a numeric comparison is performed. For example: "10">2 returns TRUE.
• if both arguments are strings, a string comparison is performed based on the ASCII collating sequence, and is case sensitive. For example: "10">"2" returns FALSE.
• if both arguments of a comparison are dates, a date comparison is performed.

11.4.5 Order of Precedence

Expressions are evaluated in the following order of precedence:
• operations inside parentheses ( ) are evaluated first, followed by
• - (negation)
• *, /
• +, -
• ==, !=, <=, >=, <, >
• NOT
• AND, OR

Operators with the same order of precedence e.g. + and -, are evaluated from left to right. For example, the expression 3*2+6/-2 evaluates to 3, whereas 3*(2+6/-2) evaluates to -3.

11.4.6 Nested Functions

Functions can be used as arguments for other functions, and can be nested. For example, if the variable A has a value of "Andersch, Jorge", then the expression:
11.5 Summarizing Data in Headers and Footers

You can use the statistical functions count, max, min and sum (Section 11.3.3) for calculations in any header or footer section. The function is performed over the part of the document associated with the header or footer. For example, when defined as a variable in a set header, the expression \( \text{sum(Price)} \) will sum all occurrences of the data variable Price within the following set.
Rules for calculating statistical functions:

- In page header and footer sections, the functions will calculate based on the content of the page.
- In set header and footer sections, the functions will calculate based on the content of the set.
- In set continuation header and footer sections, the functions will calculate based on the content of the set, up to that point.
- In table header and footer sections, the functions will calculate based on the content of the table.
- In table continuation header and footer sections, the functions will calculate based on the content of the table, up to that point.
- In table level header and footer sections, the functions will calculate based on the content of the level.
- In table level continuation header and footer sections, the functions will calculate based on the content of the level, up to that point.

To use a statistical function in a header or footer section:

1. In the header or footer section, create the user variable (Section 10.3) and use the Expression Builder to build an expression using the count, max, min and sum functions. For example, create a user variable called AvgPrice using the expression $\text{sum(Price)}/\text{count(Price)}$, where Price is a data variable which occurs in data sections that are repeated within a table.

2. Insert the user variable as a text object or variable text object (Section 9.2), in a header or footer section.

   The header or footer section can be of a table, a table level, a page, or a set.

---

*By definition, the use of calculations within the project will create new data values in your output. Common uses of the calculations feature include page sub-totaling and calculation of line item tax rates. Please note that it is always recommended that critical page data should only be modified from within your originating financial or ERP system. Create!form is not responsible for the misuse of this feature and suggests that any calculated data conditions are tested to ensure proper results.*
Example

The following table provides examples of what values will be created in each header and footer section using the count and sum functions.

<table>
<thead>
<tr>
<th>Page 1</th>
<th>Page 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(count,sum)</td>
<td>(count,sum)</td>
</tr>
<tr>
<td>Set Header(8,20)</td>
<td>Set Cont. Header(4,9)</td>
</tr>
<tr>
<td>Page Header(4,9)</td>
<td>Page Header(4,11)</td>
</tr>
<tr>
<td>Table Header (8,20)</td>
<td>Table Cont. Header(4,9)</td>
</tr>
<tr>
<td>Table Level Header(3,7)</td>
<td>Table Level Cont. Header(1,2)</td>
</tr>
<tr>
<td>Data sectionvalue: 3</td>
<td>Data sectionvalue: 2</td>
</tr>
<tr>
<td>Data sectionvalue: 3</td>
<td>Data sectionvalue: 1</td>
</tr>
<tr>
<td>Data sectionvalue: 1</td>
<td>Data sectionvalue: 0</td>
</tr>
<tr>
<td>Table Level Footer(3,7)</td>
<td>Table Level Footer(3,5)</td>
</tr>
<tr>
<td>Table Level Header(3,5)</td>
<td>Table Level Header(2,8)</td>
</tr>
<tr>
<td>Data sectionvalue: 2</td>
<td>Data sectionvalue: 5</td>
</tr>
<tr>
<td>Data sectionvalue: 2</td>
<td>Data sectionvalue: 3</td>
</tr>
<tr>
<td>Table Level Cont. Footer(4,2)</td>
<td>Table Level Footer(2,8)</td>
</tr>
<tr>
<td>Table Cont. Footer(4,9)</td>
<td>Table Footer(8,20)</td>
</tr>
<tr>
<td>Page Footer(4,9)</td>
<td>Page Footer(4,11)</td>
</tr>
<tr>
<td>Set Cont. Footer(4,9)</td>
<td>Set Footer(8,20)</td>
</tr>
</tbody>
</table>
Customizing

You can customize the look and feel of the Create!form Transform interface, set language options and other user preferences.

The following topics are covered:

- **User Preferences**
- **Spelling and Language**
- **Colors**
12.1 User Preferences

You can customize the look, feel and behavior of Create!form Transform to suit the way you work. User preference settings are specific to the local copy of Create!form Transform; changes made to user preferences are not saved with the project.

All user preferences are set from the Preferences dialog, which will be displayed by selecting Tools>Preferences from the Main menu. The categories of settings available are displayed in the tree view on the left of the dialog. Select the category in the tree view to display the corresponding controls on the right.

- General Preferences (Section 12.1.1)
- Section Borders Preferences (Section 12.1.2)
- Input Source Preferences (Section 12.1.3)
- Design Window Preferences (Section 12.1.4)
- Design Window Sections Preferences (Section 12.1.5)
- Thumbnails Preferences (Section 12.1.6)
- Input (CSV) Window Preferences (Section 12.1.7)
- Input (JDE PDF) Window Preferences (Section 12.1.8)
- Input (JDE PDF) Window Sections Preferences (Section 12.1.9)
12.1 User Preferences

- Input (Text) Window Preferences (Section 12.1.10)
- Input (Text) Window Sections Preferences (Section 12.1.11)
- Input (XML) Window Preferences (Section 12.1.12)

12.1.1 General Preferences

Select ToolsPreferences from the Main menu to display the Preferences dialog and select General in the tree view.

The following general preferences settings are available:

**Default project directory**
Click the ( ) button to change the project directory (Section 3.2.1).

**Common project directory**
Click the ( ) button to change the common project directory (Section 3.2.2).

**Spell check language**
To select a different language for checking spelling, choose a language from the drop-down list in the Spell check language field. For information on using the spell checker and adding other languages and dictionaries, see Section 12.2.

**Events to be recorded**
Select the level required for logging. To use the level set in Create!form Server, select Default to printer settings.

**Allow direct editing of expressions and conditions**
Select the checkbox to enable direct editing of expressions and conditions in dialogs without opening the Expression Builder or the Condition Builder.

**Display dialog on creating objects**
Select the checkbox to display the Properties dialog whenever a new text, line, grid, box and ellipse object is created from the Object toolbar (Section 9.2). You can also display the Properties dialog of an object when you:

- double-click the object
- right-click the object and select Properties from the shortcut menu
- select EditProperties from the Main menu
Make backup on save

Select the checkbox to make a backup copy of the project and DataMap files when the project is saved (Section 3.5).

To use a backup version of a project file and a DataMap file, you must manually change the extension of the backup files.

Restore suppressed messages

Select this option to restore messages that have previously been suppressed. A message is suppressed when the “Do not show this message again” option is selected on a message dialog.

Generate log file

Select this option to record events in a log file.

12.1.2 Section Borders Preferences

Select Tools ▶ Preferences from the Main menu to display the Preferences dialog and select General ▶ Section Borders in the tree view. The following preferences settings are available:

• the color of a section when it is selected
• the color of the corresponding section in an inactive window
• the thickness of the border

12.1.3 Input Source Preferences

Select Tools ▶ Preferences from the Main menu to display the Preferences dialog and select General ▶ Input Source in the tree view. The following preferences settings are available:

Display Font

• Select the input file source view display font in the drop-down.
• Select the font Size.
• Select the Monospace only checkbox to restrict the fonts available in the font drop-down.

Colors

Select the colors for the text and the background.
12.1 User Preferences

Scroll bars
Display and hide the horizontal and vertical scroll bars.

12.1.4 Design Window Preferences

Select Tools >> Preferences from the Main menu to display the Preferences dialog and select Design in the tree view. The following preferences settings are available:

View
To set the default page view magnification, choose Fit to Width, Fit in Window or Actual Size from the View drop-down list.

Rulers
Horizontal and vertical rulers are available to assist in object layout and placement.

Unit of measure
The page is always measured in columns (characters) and rows (lines).

Nudge offset
Type the offset distance in the current units of measure applied when nudging objects with the arrow keys (Section 9.5.4).

Synchronize with input window
Select the checkbox to synchronize the page and section display in the Design Window with selections made in the Input Window. You can also separately synchronize the display in the Input Window with selections made in the Design Window (Section 12.1.7).

Show variables in red
Select the checkbox to highlight variable text objects in red in the Design Window.

Show text object outline
Select the checkbox to display the frame of all text and variable text objects.

Show variable names
Select the checkbox to display the actual variable names on screen in place of the values.
Enable tooltips
Select the checkbox to display tooltips.

12.1.5 Design Window Sections Preferences
Select Tools>Preferences from the Main menu to display the Preferences dialog and select Design>Sections in the tree view. The following preferences settings are available:

Section borders
- Select the Show checkbox to display section borders in the Design Window.
- Select the section border line style from the Style drop-down.

Selected section
- Select the Highlight background checkbox to display the selected section in the background color shown. To change the background color, click the background color field to display the Color Picker dialog (Section 12.3).
- Select the Show repeats checkbox in the Objects section to display all objects in all occurrences of the selected section.

Unselected sections
- Select the Highlight background checkbox to display unselected sections in the background color shown. To change the background color, click the background color field to display the Color Picker dialog (Section 12.3).
- Select the Show checkbox in the Objects section to display all objects in unselected sections.
- Select the Gray checkbox in the Objects section to gray out objects in unselected sections.

12.1.6 Thumbnails Preferences
Select Tools>Preferences from the Main menu to display the Preferences dialog and select Design>Thumbnails in the tree view.

To change the background color, click the color drop-down to display the Color Picker dialog.
12.1 User Preferences

12.1.7 Input (CSV) Window Preferences

Select Tools Preferences from the Main menu to display the Preferences dialog and select Input (CSV) in the tree view. The following preferences settings are available:

**Input file size**

To limit the size of an input data file to be used as a sample, select the Limit to checkbox and enter the maximum number of lines you want to use. To most accurately represent the sample data, it is recommended that you don’t limit the number of lines unless the number of lines affects the processing speed.

Whenever you open a project where the input file size exceeds the limit, you will be prompted to truncate the input file.

**Header**

- Select the Show checkbox to display the CSV input file header in the Input Window.
- To change the background color, click the Background color field to display the Color Picker dialog (Section 12.3).

**Detail**

- Select the Show row numbers checkbox to display the row number in the Input Window.
- To alternate the background color between consecutive sets, select different colors for odd and even sets.

**Synchronize with design window**

Select the checkbox to synchronize the display in the Input Window with selections made in the Design Window. You can also separately synchronize the display in the Design Window with selections made in the Input Window (Section 12.1.4).
12.1.8 Input (JDE PDF) Window Preferences

Select Tools>Preferences from the Main menu to display the Preferences dialog and select Input (JDE PDF) in the tree view. The following preferences settings are available:

View

To set the default page view magnification, choose Fit to Width, Fit in Window or Actual Size from the View drop-down list.

Input file size

To limit the size of an input data file to be used as a sample, select the Limit to checkbox and enter the maximum number of lines you want to use. To most accurately represent the sample data, it is recommended that you don’t limit the number of lines unless the number of lines affects the processing speed.

Whenever you open a project where the input file size exceeds the limit, you will be prompted to truncate the input file.

Show rulers

Horizontal and vertical rulers are available to measure column (character) and row (line) positions on the input page.

Enable tooltips

Select the checkbox to display names as the cursor moves over data variables in the Input Window.

Synchronize with design window

Select the checkbox to synchronize the page and section display in the Input Window with selections made in the Design Window. You can also separately synchronize the display in the Design Window with selections made in the Input Window (Section 12.1.4).
12.1.9 Input (JDE PDF) Window Sections Preferences

Select Tools ▶ Preferences from the Main menu to display the Preferences dialog and select Input (JDE PDF) ▶ Sections in the tree view. The following preferences settings are available:

**Section borders**
- Select the Show checkbox to display section borders in the Input Window.
- Select the section border line style from the Style drop-down.

**Selected section**
- Select the Highlight background checkbox to display the selected section in the background color shown. To change the background color, click the background color field to display the Color Picker dialog (Section 12.3).
- To change the variable background color, click the Variable background color field to display the Color Picker dialog.
- To change the selected variable frame color, click the Selected variable frame color field to display the Color Picker dialog.

**Unselected sections**
- Select the Highlight background checkbox to display unselected sections in the background color shown. To change the background color, click the background color field to display the Color Picker dialog (Section 12.3).

12.1.10 Input (Text) Window Preferences

Select Tools ▶ Preferences from the Main menu to display the Preferences dialog and select Input (Text) in the tree view. The following preferences settings are available:

**View**
To set the default page view magnification, choose Fit to Width, Fit in Window or Actual Size from the View drop-down list.

**Input file size**
To limit the size of an input data file to be used as a sample, select the Limit to checkbox and enter the maximum number of lines you want to use. To most accurately represent the sample data, it is recommended that you don’t limit the number of lines unless the number of lines affects the processing speed.
Whenever you open a project where the input file size exceeds the limit, you will be prompted to truncate the input file.

**Grid**

A grid can be displayed in the Input Window to assist in mapping sections and data variables. The grid lines correspond to the column and row numbering displayed on the rulers.

Use the **Grid** options to:

- Hide and show a grid.
- Show grid as points.
- Change the grid line style.

**Rulers**

Horizontal and vertical rulers are available to measure column (character) and row (line) positions on the input page. Use the **Rulers** options to:

- Hide or show ruler.
- Change the cursor indicator color.

**Display font**

- Select the **Monospace only** checkbox to restrict the fonts available in the font drop-down.
- Select the input file display font in the drop-down.

**Enable tooltips**

Select the checkbox to display names as the cursor moves over data variables in the Input Window.

**Synchronize with design window**

Select the checkbox to synchronize the page and section display in the Input Window with selections made in the Design Window. You can also separately synchronize the display in the Design Window with selections made in the Input Window (Section 12.1.4).
12.11 Input (Text) Window Sections Preferences

Select Tools\Preferences from the Main menu to display the Preferences dialog and select Input (Text)\Sections in the tree view. The following preferences settings are available:

**Section borders**
- Select the Show checkbox to display section borders in the Input Window.
- Select the section border line style from the Style drop-down.

**Selected section**
- Select the Highlight background checkbox to display the selected section in the background color shown. To change the background color, click the background color field to display the Color Picker dialog (Section 12.3).
- To change the variable background color, click the Variable background color field to display the Color Picker dialog.
- To change the selected variable frame color, click the Selected variable frame color field to display the Color Picker dialog.

**Unselected sections**
- Select the Highlight background checkbox to display unselected sections in the background color shown. To change the background color, click the background color field to display the Color Picker dialog (Section 12.3).
- To change the variable background color, click the Variable background color field to display the Color Picker dialog.

12.12 Input (XML) Window Preferences

Select Tools\Preferences from the Main menu to display the Preferences dialog and select Input (XML) in the tree view. The following preferences settings are available:

**Tooltips**
You can choose to display tooltips when the cursor hovers over an XML element, and optionally, the tooltip will display values, variable and section names, and the type of node.

**Initial document view**
When the document is opened, choose if the whole tree is expanded, or only the first level.
Chapter 12: Customizing

Colors
Select a component of the tree to display the current color setting. To customize colors, clear the **Use system color** checkbox and click the color field to display the **Color Picker** dialog.

Synchronize with design window
Select the checkbox to synchronize the page and section display in the Input Window with selections made in the Design Window. You can also separately synchronize the display in the Design Window with selections made in the Input Window (Section 12.1.4).

12.2 Spelling and Language
Create!form Transform is able to check each section for spelling errors occurring in text objects. You cannot check the spelling of variable data that is mapped from the input data file.

- Checking Spelling (Section 12.2.1)
- Languages (Section 12.2.2)
- User Dictionaries (Section 12.2.3)

12.2.1 Checking Spelling
1. Select the section you want to check spelling for.
2. From the Main menu, select **Tools** ➤ **Spelling**.
   To only check specific text objects in a section, select those text objects to spell check. If no text objects are selected, Create!form Transform will check the entire section.

Check Spelling dialog options:

<table>
<thead>
<tr>
<th>Dialog item</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not in Dictionary</td>
<td>Indicates that a misspelled word was found. The word is considered misspelled because it was not located in any open dictionaries, or was found in an exclude dictionary.</td>
</tr>
<tr>
<td>Ignore</td>
<td>Causes this occurrence of a misspelled word to be skipped.</td>
</tr>
<tr>
<td>Ignore All</td>
<td>Causes this and all further occurrences of a misspelled word to be skipped.</td>
</tr>
<tr>
<td>Change To</td>
<td>Contains a word which will replace a misspelled word when you select the <strong>Change</strong> or <strong>Change All</strong> buttons.</td>
</tr>
</tbody>
</table>
12.2 Spelling and Language

12.2.2 Languages

Create!form Transform can check spelling in a project using any installed language. For a ‘Typical’ installation, English-US and English-UK are installed:

To install another language from the installation CD:

1. Insert the installation CD and navigate to the \Utilities\Language directory. This directory contains a separate sub-directory for each of the available languages.
2. Locate the sub-directory for the language required.
3. Copy the language directory from the installation CD into the <install dir>\Win32User\MainLexicon directory.
4. The language(s) will be available the next time you start Create!form Transform. To use the new language(s), make the settings in the Preferences dialog.

<table>
<thead>
<tr>
<th>Dialog item</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suggestions</td>
<td>Contains a list of suggested replacements for the word reported as misspelled. A word selected in this list will automatically be copied to the Change To field, where it can be substituted for the misspelled word by clicking the Change button.</td>
</tr>
<tr>
<td>Change</td>
<td>The misspelled word will be replaced with the word in the Change To field.</td>
</tr>
<tr>
<td>Change All</td>
<td>This and all following occurrences of the misspelled word will be replaced with the word in the Change To field.</td>
</tr>
<tr>
<td>Suggest</td>
<td>Causes a group of suggested replacements for misspelled words to be displayed in the Suggestions list.</td>
</tr>
<tr>
<td>Add</td>
<td>Causes the misspelled word to be added to the dictionary selected in the Add words to list.</td>
</tr>
<tr>
<td>Add Words To</td>
<td>Indicates which user dictionary words will be added to when you click the Add button.</td>
</tr>
<tr>
<td>Dictionaries</td>
<td>Display the Dictionaries dialog. Use this feature to open or close user dictionaries and to edit the contents of user dictionaries. For more information on the dictionary, click Help in the Dictionaries dialog.</td>
</tr>
<tr>
<td>Options</td>
<td>Display the Options dialog to customize the way the dictionary works.</td>
</tr>
<tr>
<td>Lock Pos.</td>
<td>Locks the position of the Check Spelling dialog in place.</td>
</tr>
</tbody>
</table>
12.2.3 User Dictionaries

There are four types of dictionary. When you create a new dictionary, you must specify the type. The type indicates what happens when a word is found in that dictionary during a spelling check as explained below:

<table>
<thead>
<tr>
<th>Dictionary type</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Change</td>
<td>Words found in an Auto Change dictionary are automatically replaced with other words. Typically, Auto Change dictionaries hold frequently misspelled words and their correct replacements. Each entry in an Auto Change dictionary contains a word and the replacement, separated by a colon (e.g., “recieve:receive”).</td>
</tr>
<tr>
<td>Conditional Change</td>
<td>Words found in a Conditional Change dictionary are presented to you as candidates for replacement and are replaced with other words if you request.</td>
</tr>
<tr>
<td>Exclude</td>
<td>Words found in an Exclude dictionary are always considered misspelled, even if they are defined in other dictionaries.</td>
</tr>
<tr>
<td>Ignore</td>
<td>Words found in an Ignore dictionary are considered correctly spelled.</td>
</tr>
</tbody>
</table>

12.3 Colors

The default color palette provides a standard range of colors to choose from. The current color selection on a dialog option will be shown on the color button.

To display the Color Picker dialog, click the color drop-down. The default color palette is shown on the Basic tab. To customize the color palette, select the Custom tab on the Color Picker dialog and use the controls to mix the colors you want.
Figure 12-2: The Color Picker dialogs.
Glossary

This appendix provides a list of commonly used Create!form terms.
## A.1 Glossary

The following table explains many of the terms specific to Create!form.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>alias</td>
<td>An alternative name for a source or child section used to resolve problems with duplicate variable names in child and source sections.</td>
</tr>
<tr>
<td>alternate section</td>
<td>A section that alternates with a data section when user-defined conditions are met. The characteristics and content of an alternate section are independent of the section they alternate for.</td>
</tr>
<tr>
<td>argument</td>
<td>The component of a function or expression that defines a value.</td>
</tr>
<tr>
<td>common project directory</td>
<td>The directory used to store shared project resources.</td>
</tr>
<tr>
<td>condition</td>
<td>A user-defined logical test that results in either a true or false value. Conditions enable the project to behave differently according to the content of the input file or the production environment.</td>
</tr>
<tr>
<td>data section</td>
<td>A section that is brought into the project from the input file.</td>
</tr>
<tr>
<td>DataMap</td>
<td>A DataMap records how the different parts of an input file are identified and labelled.</td>
</tr>
<tr>
<td>data variable</td>
<td>A variable defined in the input file.</td>
</tr>
<tr>
<td>data copying</td>
<td>The creating of variable text objects in the project, using data variables from the input file.</td>
</tr>
<tr>
<td>derived variable</td>
<td>A user-defined expression, defined in the input.</td>
</tr>
<tr>
<td>detail section</td>
<td>A section in the input file containing repeating line items.</td>
</tr>
<tr>
<td>e-forms server</td>
<td>The system that hosts Create!form Server, which merges the project with incoming spool files.</td>
</tr>
<tr>
<td>enterprise server</td>
<td>ERP system or other application that is the originator of the input files.</td>
</tr>
<tr>
<td>expression</td>
<td>A user-defined statement that performs calculations and derives values from the input data.</td>
</tr>
<tr>
<td>form project</td>
<td>See project.</td>
</tr>
<tr>
<td>function</td>
<td>A pre-defined formula that performs a calculation from user defined arguments.</td>
</tr>
<tr>
<td>global user variable</td>
<td>A user-defined variable, global to the project.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>input file</td>
<td>A sample file from the enterprise server, used to build the DataMap.</td>
</tr>
<tr>
<td>inserted section</td>
<td>An inserted section is created by the user and attached to a data section.</td>
</tr>
<tr>
<td>lookup variable</td>
<td>A variable that derives its value from an external database.</td>
</tr>
<tr>
<td>standard project</td>
<td>A project that utilizes all the design features of Create!form Transform.</td>
</tr>
<tr>
<td>object</td>
<td>Text, graphic or barcode object inserted in the project.</td>
</tr>
<tr>
<td>operator</td>
<td>A component of an expression or condition that defines an arithmetic, logical or text operation on adjacent arguments.</td>
</tr>
<tr>
<td>overlay project</td>
<td>A project where the data from the input file will be displayed in the project without change.</td>
</tr>
<tr>
<td>project</td>
<td>A project is a record of the way you want to redesign the input file. Also referred to as a form project.</td>
</tr>
<tr>
<td>project directory</td>
<td>The directory where the project resources are stored.</td>
</tr>
<tr>
<td>repagination</td>
<td>The process of expanding or condensing pages in the project depending on the space required.</td>
</tr>
<tr>
<td>repeated section</td>
<td>A data section that appears more than once in a set.</td>
</tr>
<tr>
<td>set</td>
<td>A user-defined construct that divides the input file into its basic document units, that the project is designing.</td>
</tr>
<tr>
<td>simple project</td>
<td>A project where only limited reformatting of the input file is required.</td>
</tr>
<tr>
<td>source</td>
<td>Where a section derives its data from. The source of a section can be a section from the input file, or another section in the project.</td>
</tr>
<tr>
<td>subform</td>
<td>A subform is a user defined graphical object.</td>
</tr>
<tr>
<td>system variable</td>
<td>System variables are predefined and relate to the project. They include such items as the number of pages in the set or document.</td>
</tr>
<tr>
<td>table</td>
<td>A table is a user created structure that enables you to better organize and display a series of consecutive data sections.</td>
</tr>
<tr>
<td>template</td>
<td>A project, used to define other projects.</td>
</tr>
<tr>
<td>user variable</td>
<td>User variables are user-defined expressions that relate only to the section they were created in.</td>
</tr>
</tbody>
</table>
Quick Guide

A quick reference to menu commands and how to select commands, dialog options and perform other tasks from the keyboard.

The following topics are covered:

➤ Quick Guide to Menu Commands
➤ Quick Guide to Keyboard Shortcuts
➤ Quick Guide to Tutorials
## B.1 Quick Guide to Menu Commands

### File Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Start the New Project Wizard.</td>
<td>3.3</td>
</tr>
<tr>
<td>Open</td>
<td>Open an existing project.</td>
<td>3.6</td>
</tr>
<tr>
<td>Close</td>
<td>Close the active project.</td>
<td></td>
</tr>
<tr>
<td>Save</td>
<td>Save the active project.</td>
<td>3.5</td>
</tr>
<tr>
<td>Save As</td>
<td>Save the active project to another name or</td>
<td>3.8</td>
</tr>
<tr>
<td></td>
<td>as a packed project.</td>
<td></td>
</tr>
<tr>
<td>Page Setup</td>
<td>Change the page layout settings.</td>
<td>6.2</td>
</tr>
<tr>
<td>Print</td>
<td>Print the active project.</td>
<td>3.7</td>
</tr>
<tr>
<td>Print Preview</td>
<td>Preview the active project in Notepad.</td>
<td>3.7</td>
</tr>
<tr>
<td>Properties</td>
<td>Display information about the active</td>
<td></td>
</tr>
<tr>
<td></td>
<td>document.</td>
<td></td>
</tr>
<tr>
<td>1 2 3 ...</td>
<td>Open a recent file.</td>
<td></td>
</tr>
<tr>
<td>Exit</td>
<td>Close all open documents and exit.</td>
<td></td>
</tr>
</tbody>
</table>

### Edit Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undo</td>
<td>Undo last edit.</td>
<td></td>
</tr>
<tr>
<td>Redo</td>
<td>Redo previously undone edit.</td>
<td></td>
</tr>
<tr>
<td>Cut</td>
<td>Copy selected items to clipboard and delete.</td>
<td></td>
</tr>
<tr>
<td>Copy</td>
<td>Copy selected items to clipboard.</td>
<td>9.5.6</td>
</tr>
<tr>
<td>Paste</td>
<td>Paste contents of clipboard to project.</td>
<td>9.5.6</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete selected object or item.</td>
<td></td>
</tr>
<tr>
<td>Select All</td>
<td>Select all objects in current section.</td>
<td>9.5.2</td>
</tr>
</tbody>
</table>
### B.1 Quick Guide to Menu Commands

#### View Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move</td>
<td>Move object by horizontal or vertical increments.</td>
<td>9.5.4</td>
</tr>
<tr>
<td>Properties</td>
<td>Display and edit object properties.</td>
<td>9.4</td>
</tr>
</tbody>
</table>

**Command**

- **Graphic**: Display objects graphically. (4.1.1)
- **Listing**: Display objects in a list. (4.1.1)
- **Input Window**: Hide and show Input Window. (4.1.2)
- **Input Tree**: Hide and show Input Tree view. (4.1.4)
- **Project Tree**: Hide and show Project Tree view. (4.1.3)
- **Thumbnails**: Hide and show page thumbnail view. (4.1.5)
- **Actual Size**: Display active window true size.
- **Fit in Window**: Fit page inside active window.
- **Fit to Width**: Fit page to width of active window.
- **Zoom Window**: Zoom into user defined window.
- **Zoom In**: Increase magnification.
- **Zoom Out**: Decrease magnification.
- **Toolbars**: Hide and show toolbars.

#### Project Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Display and edit properties of active project.</td>
<td>6.1</td>
</tr>
<tr>
<td>Headers and Footers</td>
<td>Manage project headers and footers.</td>
<td>7.3.3</td>
</tr>
<tr>
<td>Global User Variables</td>
<td>Create and edit global user variables.</td>
<td>10.4</td>
</tr>
</tbody>
</table>
### Chapter B: Quick Guide

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Input File</td>
<td>Select a new input file.</td>
<td>5.1.2</td>
</tr>
<tr>
<td>Apply Design Template</td>
<td>Select and apply another project as a template.</td>
<td>3.3.3</td>
</tr>
<tr>
<td>Change DataMap</td>
<td>Change DataMap options.</td>
<td>3.3.4</td>
</tr>
</tbody>
</table>

#### Set Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define</td>
<td>Define when sets are created.</td>
<td>5.5.5</td>
</tr>
<tr>
<td>Headers and Footers</td>
<td>Manage set headers and footers.</td>
<td>7.3.3</td>
</tr>
<tr>
<td>Previous</td>
<td>Go to first page of previous set.</td>
<td>4.2</td>
</tr>
<tr>
<td>Next</td>
<td>Go to first page of next set.</td>
<td>4.2</td>
</tr>
<tr>
<td>Go To</td>
<td>Go to specified set.</td>
<td>4.2.1</td>
</tr>
</tbody>
</table>

#### Page Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers and Footers</td>
<td>Manage display of page headers and footers.</td>
<td>7.3.3</td>
</tr>
<tr>
<td>First</td>
<td>Go to first page.</td>
<td>4.2</td>
</tr>
<tr>
<td>Previous</td>
<td>Go to previous page.</td>
<td>4.2</td>
</tr>
<tr>
<td>Next</td>
<td>Go to next page.</td>
<td>4.2</td>
</tr>
<tr>
<td>Last</td>
<td>Go to last page.</td>
<td>4.2</td>
</tr>
<tr>
<td>Go To</td>
<td>Go to specified page.</td>
<td>4.2.1</td>
</tr>
<tr>
<td>Style Properties</td>
<td>Display properties of the current page.</td>
<td>6.2</td>
</tr>
</tbody>
</table>

#### Table Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Define a new table in the selected section.</td>
<td>8.2.2</td>
</tr>
</tbody>
</table>
### Section Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Properties</td>
<td>Display and edit properties of selected section.</td>
<td>7.6</td>
</tr>
<tr>
<td>Alternates</td>
<td>Define alternates for selected section.</td>
<td>7.4</td>
</tr>
<tr>
<td>Insert</td>
<td>Insert a section before or after the selected section.</td>
<td>7.5</td>
</tr>
<tr>
<td>Data Variables</td>
<td>Display data variables defined in selected section.</td>
<td>10.2</td>
</tr>
<tr>
<td>User Variables</td>
<td>Create and modify user variables in selected section.</td>
<td>10.3</td>
</tr>
<tr>
<td>Lookup Variables</td>
<td>Create and modify lookup variables in selected section.</td>
<td>10.6</td>
</tr>
<tr>
<td>Previous</td>
<td>Select previous section.</td>
<td>4.2.2</td>
</tr>
<tr>
<td>Next</td>
<td>Select next section.</td>
<td>4.2.2</td>
</tr>
<tr>
<td>First</td>
<td>Select first section on page.</td>
<td>4.2.2</td>
</tr>
<tr>
<td>Last</td>
<td>Select last section on page.</td>
<td>4.2.2</td>
</tr>
<tr>
<td>Previous Occurrence</td>
<td>Select previous occurrence of the same section.</td>
<td>4.2.2</td>
</tr>
<tr>
<td>Next Occurrence</td>
<td>Select next occurrence of the same section.</td>
<td>4.2.2</td>
</tr>
<tr>
<td>Copy Data Variables</td>
<td>Copy all data variables from input section.</td>
<td>9.2.1.4</td>
</tr>
</tbody>
</table>

### Input Menu (CSV)

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Properties</td>
<td>Define CSV file structure.</td>
<td>5.4.2</td>
</tr>
<tr>
<td>Derived Variables</td>
<td>Create and modify derived variables in selected section.</td>
<td>5.8</td>
</tr>
</tbody>
</table>
## Input Menu (Text)

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Section After</td>
<td>Insert a new section after the selected section.</td>
<td>5.5.10.2</td>
</tr>
<tr>
<td>Input Headers</td>
<td>Manage input headers control list.</td>
<td>5.5.9.3</td>
</tr>
<tr>
<td>Input Footers</td>
<td>Manage input footers control list.</td>
<td>5.5.11.2</td>
</tr>
<tr>
<td>Delete Section</td>
<td>Delete the selected section.</td>
<td>5.5.7.4</td>
</tr>
<tr>
<td>File Properties</td>
<td>Define input page width and length.</td>
<td>5.5.4</td>
</tr>
<tr>
<td>Derived Variables</td>
<td>Create and modify derived variables in selected section.</td>
<td>5.8</td>
</tr>
<tr>
<td>Section Properties</td>
<td>Display and edit properties of selected input section.</td>
<td>5.5.7.1</td>
</tr>
<tr>
<td>Show All Instances</td>
<td>Show how selected section is populated.</td>
<td>5.5.2</td>
</tr>
</tbody>
</table>

## Input Menu (XML)

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag Properties</td>
<td>Change how sections are created and named.</td>
<td>5.6.4</td>
</tr>
<tr>
<td>Variable Properties</td>
<td>Change the properties of data variables.</td>
<td>5.6.5</td>
</tr>
<tr>
<td>File Properties</td>
<td>Define input data formats.</td>
<td>5.6.2</td>
</tr>
<tr>
<td>Derived Variables</td>
<td>Create and modify derived variables in selected section.</td>
<td>5.8</td>
</tr>
</tbody>
</table>

## Input Menu (JDE PDF)

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Properties</td>
<td>Define input data mapping.</td>
<td>5.7.1</td>
</tr>
<tr>
<td>Derived Variables</td>
<td>Create and modify derived variables in selected section.</td>
<td>5.8</td>
</tr>
</tbody>
</table>

B.1 Quick Guide to Menu Commands

Tools Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Subform.Designer New</td>
<td>Create a subform.</td>
<td>9.2.2.4</td>
</tr>
<tr>
<td>Create Subform.Import</td>
<td>Convert text files into subforms.</td>
<td>9.2.2.3</td>
</tr>
<tr>
<td>Preferences</td>
<td>Customize the display of windows and other user preferences.</td>
<td>12.1</td>
</tr>
<tr>
<td>Spelling</td>
<td>Run spelling checker.</td>
<td>12.2.1</td>
</tr>
<tr>
<td>Format Styles</td>
<td>Manage number, date and currency formats.</td>
<td>6.1.3</td>
</tr>
<tr>
<td>Environment Variables</td>
<td>Create and modify environment variables.</td>
<td>10.7</td>
</tr>
<tr>
<td>Transfer</td>
<td>Transfers project and resources to e-forms server for production printing.</td>
<td>3.9</td>
</tr>
</tbody>
</table>

Window Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cascade</td>
<td>Arrange windows so they overlap.</td>
<td></td>
</tr>
<tr>
<td>Tile Horizontally</td>
<td>Arrange windows as non-overlapping horizontal tiles.</td>
<td></td>
</tr>
<tr>
<td>Tile Vertically</td>
<td>Arrange windows as non-overlapping vertical tiles.</td>
<td></td>
</tr>
<tr>
<td>Split</td>
<td>Split the active window into duplicate panes.</td>
<td></td>
</tr>
<tr>
<td>Split Horizontally</td>
<td>Arrange Design Window and Input Window horizontally.</td>
<td></td>
</tr>
<tr>
<td>Split Vertically</td>
<td>Arrange Design Window and Input Window vertically.</td>
<td></td>
</tr>
<tr>
<td>1 2 3 ...</td>
<td>Make the selected window active.</td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td>Manage workspace.</td>
<td></td>
</tr>
</tbody>
</table>

Help Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help Topics</td>
<td>Display help.</td>
<td>1.5</td>
</tr>
</tbody>
</table>
B.2 Quick Guide to Keyboard Shortcuts

B.2.1 Menu Commands

To select any command from a menu with the keyboard:

1. Press the ALT key together with the letter underlined in the menu name that contains the command you want.
2. In the menu that appears, press the underlined letter in the command name that you want.

To select a command directly with the keyboard:

<table>
<thead>
<tr>
<th>Commands</th>
<th>Usage</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create!form on the Web</td>
<td>Visit the Create!form web site.</td>
<td></td>
</tr>
<tr>
<td>About</td>
<td>Display information about this version of</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create!form Transform.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To select this command</th>
<th>Press</th>
</tr>
</thead>
<tbody>
<tr>
<td>File New</td>
<td>CTRL+N</td>
</tr>
<tr>
<td>File Open</td>
<td>CTRL+O</td>
</tr>
<tr>
<td>File Save</td>
<td>CTRL+S</td>
</tr>
<tr>
<td>File Save As</td>
<td>F12</td>
</tr>
<tr>
<td>File Print</td>
<td>CTRL+P</td>
</tr>
<tr>
<td>File Preview</td>
<td>CTRL+SHIFT+P</td>
</tr>
<tr>
<td>Edit Undo</td>
<td>CTRL+Z OR ALT+BACKSPACE</td>
</tr>
<tr>
<td>Edit Redo</td>
<td>CTRL+Y</td>
</tr>
<tr>
<td>Edit Cut</td>
<td>CTRL+X OR SHIFT+DEL</td>
</tr>
<tr>
<td>Edit Copy</td>
<td>CTRL+C OR CTRL+INSERT</td>
</tr>
<tr>
<td>Edit Paste</td>
<td>CTRL+V OR SHIFT+INSERT</td>
</tr>
<tr>
<td>Edit Delete</td>
<td>DEL</td>
</tr>
<tr>
<td>Edit Select All</td>
<td>CTRL+A</td>
</tr>
<tr>
<td>Edit Move</td>
<td>CTRL+M</td>
</tr>
<tr>
<td>Edit Properties</td>
<td>ALT+R</td>
</tr>
<tr>
<td>View Graphic</td>
<td>CTRL+G</td>
</tr>
<tr>
<td>View Listing</td>
<td>CTRL+L</td>
</tr>
</tbody>
</table>
B.2 Quick Guide to Keyboard Shortcuts

B.2.2 Dialog Options

To select any option on an active dialog with the keyboard:

<table>
<thead>
<tr>
<th>To do this on an active dialog</th>
<th>Press</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select next option or option group</td>
<td>TAB</td>
</tr>
<tr>
<td>Select previous option or option group</td>
<td>SHIFT+TAB</td>
</tr>
<tr>
<td>Switch to next dialog tab</td>
<td>CTRL+TAB</td>
</tr>
<tr>
<td>Switch to previous dialog tab</td>
<td>CTRL+SHIFT+TAB</td>
</tr>
<tr>
<td>Select an option with an underlined letter</td>
<td>ALT+LETTER</td>
</tr>
<tr>
<td>Open a selected drop-down list</td>
<td>ALT+DOWN</td>
</tr>
<tr>
<td>Move between options in an open drop-down list</td>
<td>arrow keys</td>
</tr>
<tr>
<td>Select an item in a drop-down list with the corresponding first letter</td>
<td>letter key</td>
</tr>
<tr>
<td>Perform the action assigned to the selected command button (dotted outline)</td>
<td>spacebar</td>
</tr>
<tr>
<td>Perform the action assigned to the default command button (bold outline)</td>
<td>ENTER</td>
</tr>
<tr>
<td>Cancel the command and close the active dialog</td>
<td>ESC</td>
</tr>
</tbody>
</table>
Task shortcuts when editing objects:

<table>
<thead>
<tr>
<th>To do this when an object is selected</th>
<th>Press</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move object to the left</td>
<td>LEFT</td>
</tr>
<tr>
<td>Resize object by moving left border to the left</td>
<td>SHIFT+LEFT</td>
</tr>
<tr>
<td>Resize object by moving right border to the left</td>
<td>CTRL+LEFT</td>
</tr>
<tr>
<td>Move object to the right</td>
<td>RIGHT</td>
</tr>
<tr>
<td>Resize object by moving right border to the right</td>
<td>SHIFT+RIGHT</td>
</tr>
<tr>
<td>Resize object by moving left border to the right</td>
<td>CTRL+RIGHT</td>
</tr>
<tr>
<td>Move object up</td>
<td>UP</td>
</tr>
<tr>
<td>Resize object by moving top border up</td>
<td>SHIFT+UP</td>
</tr>
<tr>
<td>Resize object by moving bottom border up</td>
<td>CTRL+UP</td>
</tr>
<tr>
<td>Move object down</td>
<td>DOWN</td>
</tr>
<tr>
<td>Resize object by moving bottom border down</td>
<td>SHIFT+DOWN</td>
</tr>
<tr>
<td>Resize object by moving top border down</td>
<td>CTRL+DOWN</td>
</tr>
</tbody>
</table>

To change the nudge offset distance when using arrow keys, see Section 12.1.4

<table>
<thead>
<tr>
<th>To do this in the active window</th>
<th>Press</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select next window pane</td>
<td>F6</td>
</tr>
<tr>
<td>Select previous window pane</td>
<td>SHIFT+F6</td>
</tr>
<tr>
<td>Display shortcut menu</td>
<td>SHIFT+F10</td>
</tr>
</tbody>
</table>
### B.3 Quick Guide to Tutorials

<table>
<thead>
<tr>
<th>Topic</th>
<th>Subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the Create!form sample projects</td>
<td>Input Design</td>
</tr>
<tr>
<td>To create the CSV sample project:</td>
<td>Designing for CSV Input Files</td>
</tr>
<tr>
<td>Input file properties</td>
<td>Designing for CSV Input Files</td>
</tr>
<tr>
<td>Copying data variables to the project</td>
<td>Designing for CSV Input Files</td>
</tr>
<tr>
<td>To create the text sample project:</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>Input sections</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>Setting the input page size</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>Defining sets</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>Create data variables</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>Copy data variables to the project</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>Create data variables in a detail section</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>Make an input header section expandable</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>Add detail sections</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>Make a detail section expandable</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>Adjust the input footer section size</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>Add input footer sections</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>Rules for creating multiple input footer sections</td>
<td>Designing for Text Input Files</td>
</tr>
<tr>
<td>To create the XML sample project:</td>
<td>Designing for XML Input Files</td>
</tr>
<tr>
<td>Defining sections</td>
<td>Designing for XML Input Files</td>
</tr>
<tr>
<td>Copying data variables to the project</td>
<td>Designing for XML Input Files</td>
</tr>
</tbody>
</table>
Index

, 185
-, 185
", 185
.cff6, 35, 143, 148
.cmp6, 35
.ctp6, 35
.ctp6bak, 35
.dmp6, 35
dmp6bak, 35
.jmp6, 35
tpp, 34
tpp6, 35
.xmp6, 35

Symbols

^, 185

A

Actual Size, 193, 196, 197
adding arguments, 183
Additional first page footer checkbox, 114
Adobe Reader, 18, 22
alias, 121, 123, 208
aligning
  and resizing, 161
  handles, 162
  objects, 158, 159
  text, 153
alignment
  buttons, 159
tools, 158
alternate sections, 116, 121, 208
deleting, 117
anchor points, 160
arguments, 208
  adding, 183
  mixed, 183
arrow keys, 157
ASCII collating sequence, 184
Assign to variables field, 169
auto-expanding sections, 119

B

background graphics, 110
backup
  on save, 192
blank pages, 67
bold and underline formatting, 68
bold and underline overstrike control characters,
  68
boolean operators, 175
Bottom orphan control checkbox, 132
Break auto-expanding section and overflow to
  next page checkbox, 122

C

Check Spelling dialog, 200
child section, 121
collate
  by set, 106	options, 107
  settings, 106
Color Picker dialog, 202
colors, 202
  mixing, 202
commands
  quick guide, 212
common project directory, 25, 191
CommonProject directory, 25
concatenate, 183
Condition Builder, 174, 182
  syntax, 174, 182
Index

conditions, 208
  boolean, 175
  building, 174
  displaying objects conditionally, 151
  examples, 173
  functions, 177
  operators, 175
  syntax, 182
  system, 182
  using, 173
  using expressions as, 174
  variables, 181
continuation header and footer sections, 114
control object, 158
Copy Data Variables command, 139
copying objects, 158
count function, 185
Create From button, 115, 130
create table with multiple section types, 120
Create!archive, 19
Create!email, 19
Create!fax, 19
Create!form Server, 19, 35
Create!form Transform
  customizing, 190
  installation requirements, 22
  installing, 22
  navigating, 37
  prerequisites, 18
  starting, 22
  viewing, 37
CSV
  columns, 58
  copying data variables, 59
  detail section, 58
  file structure, 54
  input data format, 59
  input header section, 58
  paginated, 54
  properties, 54
  currency format, 152
  currency format see also format styles
data copying, 208
data format, 67
data sections, 112, 208
display properties, 113
  properties, 112
data variable, 48
  expandable, 75
data variables, 48, 164, 181, 208
copying to project, 59, 73, 100
  creating, 70
  CSV, 58
database, 168
  Database (DSN) field, 169
DataMap, 25, 47, 208
  options, 27
date
  addition, 183
difference, 184
  format, 152
  format see also format styles
  order, 59, 70, 100
  subtraction, 184
date format, 55
date order, 70, 102
decimal format, 55
decimal separator, 59, 70, 100
decimal symbol, 70
default project directory, 191
defining sets, 69
derived variable, 49
derived variables, 49, 164, 208
  creating, 104
Description field, 151
design
  input, 54, 63, 102
  overview, 28
  top-down, 63
design concepts, 47
Design New button, 143, 154
Design Window, 38, 39
detail section
  additional, 85, 86
  creating, 85, 86
  expandable, 89
  rules, 90
  working with, 85
dictionaries, 201, 202
display order of header and footers, 116
display preferences, 190
displaying objects conditionally, 151
document header, 67
Don’t split paragraphs checkbox, 122
DSN, 168
duplicates, setting, 107

E

Edit button, 154
e-forms server, 19, 35
e-mailing projects, 34
enterprise server, 208
environment variable, 170
environment variables, 181
ERP system, 186
expandable
  data variable, 72, 75
  detail section, 89
  input header section, 81
Expression Builder, 172
  using, 172
expressions, 164, 208
  building, 172
  direct editing, 191
  functions, 177
  navigating and editing, 173
  operators, 175
  system variables, 167
  variables, 167, 181

F

Fit in Window, 193, 196, 197
Fit to Width, 193, 196, 197
fixed sections, 118, 119
floating section, 119
flowchart
  input design, 64
Followed By list, 77, 81
font, 198
form projects, 208
format, 152
  input data, 59, 70, 100
format style
  default, 109
  text, 153
format styles
  applying to variables, 153
  creating and setting default, 107
  managing, 107
formats used by data variables, 67
Index

function
  aswords, 177
  at, 178
  concat, 178
  contains, 177
  count, 181
  date, 180
  day, 180
  exists, 177
  find, 178
  isnumeric, 177
  length, 178
  lower, 178
  ltrim, 178
  max, 181
  min, 181
  mod, 178
  month, 180
  num, 178
  pad, 178
  previous, 179
  previousat, 179
  proper, 178
  retrieve, 179
  rfind, 179
  round, 179
  rtrim, 179
  sentence, 179
  substitute, 180
  substring, 180
  sum, 181
  trim, 180
  upper, 180
  year, 180
functions, 208
  date, 180
  in conditions, 177
  in expressions, 177
  logical, 177
  nested, 184
  statistical, 181, 185
  text and numeric, 177

G

Gap between lowest object and section bottom field, 120
General Preferences, 191
general properties, 151
Generate Section, 78
global user variables, 166, 181, 208
  creating, 166
go to page, 42
go to set, 42
Graphics View, 39
Gray checkbox, 194
grid
  hiding and showing, 198
  grid settings, 198
  grouping within a table, 130

H

handles
  aligning, 162
  deselecting, 161
  selecting, 160, 161
Header and Footer dialog, 114
header and footer sections, 113
  creating, 114, 115
  display order, 116
  in tables, 130
  page, 114
  project, 114
  set, 114
  summarizing data, 185
  turning off, 115
HeaderParams.cfg, 170
headers and footers sections
  creating, 114
  highlight background, 197, 199
  Highlight background checkbox, 194
  highlight variable text objects, 193
Index

I

Input file header section, 61
Ignore blank lines checkbox, 152
ignore blank pages, 67
Ignore data when hidden checkbox, 113
ignored tags, 96
Import button, 142, 154
initial value, 166
Initial Value column, 166
input
data format, 59, 70, 100
design
flowchart, 64
design principles, 54, 63, 102
page size, 64, 67
sections, 48, 61
input file, 209
changing, 46
properties, 64, 67
size, 195, 196, 197
viewing, 47
input file header section, 67
input file properties, 54, 68
input file source, 47
Input File Source View, 38
input files
text, 60
input footer section
additional, 91
adjusting, 90
rules, 93
working with, 90
input footers control list, 91
input header section
additional, 83
adjusting, 80
creating, 83
deleting, 84
expandable, 81
multiple, 84
rules, 84
working with, 80
input headers control list, 83
input sections
generating, 78
logic, 78
multiple, 78
properties, 76, 78
Input Tree, 38, 40
Input Window, 38, 39
Insert Variable button, 138
inserted sections, 117, 209
installation requirements, 22
installing Create!form, 21, 22

J

J.D. Edwards PDF format, 101
JDE PDF
data variables, 103
date format, 102
defining sets, 103
file properties, 102
non-breakable sections, 102
sections, 103
JDE PDF input files, 101
JDEHeaderParams.cfg, 170
job ticket variable, 170
job ticket variables, 181

K

keep sections together, 120
Keep together checkbox, 132
Keep With checkbox, 121
kerning, 153
key
SHIFT, 161
key concepts, 47
keyboard shortcuts
tasks, 220

L

languages, 191, 200
installing from CD, 201
Index

Limit number of pages to checkbox, 195, 196, 197
line spacing, 153
lines and columns, 67
Listing View, 39, 135, 156
lookup variables, 168, 181, 209
Lookup Variables dialog, 168

---

M

Main menu, 212
Manage Formats, 153
mapping data, 139
max function, 185
media types, 110
menu commands, 212
min function, 185
minus sign, 185
mixed data types, 183
monospace, 192, 198
Move Object(s) dialog, 157
Move tool, 157
moving objects, 157

---

N

navigating in Create!form, 42
nested functions, 184
Never Display setting, viewing section properties when set to, 113
new line, 185
New page checkbox, 132
New Project wizard, 25
New Subform dialog, 143
non-breakable input sections, 102
NOT, 175
nudge offset, 193
number formats, 152
see also format styles

---

O

object, 209
object properties
default, 154
displaying, 149
object property classes, 149
objects
aligning, 158
copying, 158
creating, 136
description, 151
deselecting, 155
display dialog on creating, 191
displaying conditionally, 151
list, 39
moving, 157
object listing, 39
outline, 193
position and size, 151
properties, 148
property classes, 150
resizing, 157
scaling, 151
selecting, 155
subform, 140
type, 150
variable subform, 143
Occurrence, 123
Occurrence field, 115
ODBC, 168
operator
, 176, 176
-, 176
-(negation), 176
!=, 176
*, 176
+, 176
/, 176
==, 176
>, 176
>=, 176
AND, 175
NOT, 176
OR, 175
operators, 175, 209
in conditions, 175
order of precedence, 184
Index

orphan control
  for broken text objects, 121
  for tables, 132
Orphan control checkbox, 122
OS/400 spool files, 68
output options, 27
overlay project, 67
overlay projects, 28, 209
overview
  design, 28


P

packing projects, 34
Page control, 109
page control, 109
page size, 109
  calculating, 66
  input, 64, 67
  recalculating, 67
page style, 109
page styles
  using for background graphics, 110
Pagination Control, 132
pagination control within a table, 132
parent/child relationships, 120
  defining, 121
parentheses, 184
parsing character, 181
paste, 158
PDF, 19
position and size, 151
preferences, 190
  Design Window, 193
  general, 191
  grid settings, 198
  input sections, 197, 199
  input source, 192
  Input Window, 195, 196, 197, 199
  section borders, 192, 194
  thumbnails, 194
Preferences dialog, 190
prerequisites, 18
previewing projects, 33

previous
  occurrence, 42
  page, 42
  set, 42
printing, 33
production printing, 35
project directory, 24, 191, 208, 209
project properties, setting the, 106
Project Tree, 38, 40, 135, 156
projects, 209
  creating, 25
  managing files, 35
  navigating in, 42
  opening, 33
  options, 25
  overlay, 28
  overview, 28
  packing, 34
  printing and previewing, 33
  saving, 32
  sending, 34
  simple, 28
  standard, 27
  transferring, 35
property classes, 150
  default properties, 154
  general, 151
  objects, 149
  subform, 154
  variable, 152

Q

quotation mark, 185

R

Refresh button, 169
repagination, 121, 209
repeated section, 209
resize and move object, 161
resizing a section, 119
resizing objects, 157
Index

rulers, 193, 196, 198

S

ds, 57
alternate, 116, 208
auto-expanding, 120
conditional display, 113
continuation header and footer, 114
CSV, 57
data, 112
deleting alternates, 117
detail, 48, 61, 112
displaying properties, 118
fixed and auto-expanding, 119
header and footer, 113
input, 48, 61
input footer, 48, 61, 112
input header, 48, 61, 80, 112
inserted, 117
mapping data to, 139
parent/child relationships, 120
populated, 63
positioning, 118
rules for positioning, 119
selecting, 43
setting when to display, 112
sinking position, 119
unselected, 194, 197, 199

Select button, 141, 154
Select records where field, 169
selecting, 155
handles, 161
hidden objects, 155
set, 48
sets, 69, 209
CSV, 56
defining, 69
defining in CSV, 57
Show repeats checkbox, 194
show variable names, 193
simple project, 28, 66, 209
sinking section, 119
sort mode, 129
sorting data using tables, 126
sorting modes
ASCII, 129
numeric, 129
regional, 129
source, 164
  adding a, 123
Source field, 115
special characters, 185
spell check, 191
spelling, 200
  language, 191
spelling languages, 201
standard project, 66
standard projects, 27, 209
statistical functions, 181
  rules, 186
  using, 185
Status bar, 38, 41
Strip leading spaces checkbox, 152
Strip trailing spaces checkbox, 152
subform objects, 140
subforms, 140, 209
  CFF6, 143
  creating new, 143
  editing, 148
  insert existing, 141
  properties, 154
  storage, 141
  variable, 143
subtraction of numeric, text and date arguments, 183
sum function, 185
synchronize, 193, 195, 196, 198, 200
  windows, 42
syntax
  boolean operators, 175
  conditions, 182
  functions, 177
  operators, 175
system conditions, 182
system variables, 167, 181, 209
tables, 126, 209
  creating, 126, 127
  drawing objects, 126
  grouping, 130
  header and footer sections in, 130
  levels, 129
  pagination control, 132
  parent/child relationships, 131
  sort modes, 129
Tag Properties, 97
templates, 209
text
  addition, 183
  comparing, 184
  data format, 70
  data variables, 70
  defining sets, 69
  detail section, 62
  format style, 153
  input footer section, 62
  input header section, 61
  input sections, 61
  kerning, 153
  line spacing, 153
text delimiter, 185
text input files, 60
Thumbnail View, 40
toolbars
  Alignment, 38
  Main, 38
  Navigation, 38
  Object, 38
  Text, 38
tooltips, 194, 196, 198
Top orphan control checkbox, 132
Transfer command, 35
transferring projects, 35
uncollate, 106
unit of measure, 193
Use only first checkbox, 130
use variables from associated sections, 120
variable format styles see format styles
variable properties, 152
variable subform objects, 143
creating, 143
Variable Subform Properties dialog, 143
variables
  in expressions and conditions, 181
  naming, 185
  see also lookup variables, data variables,
  user variables, global user
  variables, system variables,
  archive variables and constants.
  show names, 193
viewing and selecting Sections, 43

Windows locale, 129
wizard, 114
wizard for new projects, 25
WorkDir directory, 24
Wrap (ignore end of line) checkbox, 152

XML
  copying data variables, 100
  data format, 95, 96, 100
  data variables, 99
  defining sections, 97
  defining sets, 97
  file properties, 96
  file structure, 95
XML input files, 94