

# Interstage List Creator V10.0.0



## Advanced Form Design

Windows/Windows(64)/Solaris/Linux(64)

B1X1-0195-01ENZ0(00)  
October 2012

---

# Preface

---

This manual explains the following tasks performed with List Creator Designer:

- Form designing
  - Create the form descriptor information required for outputting forms. Also, design the input data placed in the form descriptor information to output.
- Managing and maintaining form descriptor information (Maintaining forms)
  - Manage form descriptor information, and maintain (correct or modify) it as necessary.

The support status of the functions defined in the form descriptor information vary depending on the operating system of the List Creator print server. For details, refer to the Online Manual "Setup & Operation".

Contents of this manual are common for all operating systems.

## Structure of This Manual

This manual contains the following chapters.

### Part 1 Getting Started with Advanced Form Designing

This part explains the basic knowledge for form designing in List Creator.

Since the commonly required information for creating any form is mentioned, be sure to refer to it before designing a form.

#### **Chapter 1 Basic Knowledge of Form Designing**

This chapter explains the following basic knowledge of form designing.

- What is List Creator form designing?
- Information Defined during Form Designing
- Components of form layout data
- What is form format?
- Input data type and format
- Files created in form designing

#### **Chapter 2 Form Designing Flow**

This chapter explains the work flow and tasks in form designing.

#### **Chapter 3 Windows and Operating Environment used for Form Designing**

This chapter explains the windows used for form designing and the operating environment when designing forms.

### Part 2 Form Designing

This part explains the details of form designing.

Refer to this part to understand the functions and the flow of operation while designing a form.

#### **Chapter 4 Preparing for Form Designing**

This chapter explains the preparations for form designing.

#### **Chapter 5 Creating Form Layout Data**

This chapter explains the procedures for using the wizard to create form layout data.

How to create new form layout data when there are no existing resources, and how to utilize existing resources are described.

#### **Chapter 6 Definition of Association Data**

---

This chapter explains the method for defining association data.

---

### **Chapter 7 Confirming the Output Image**

---

This chapter explains how to check the output image of form layout data ("Layout Print", "Preview", and "Test Print").

---

### **Chapter 8 Designing Input Data**

---

This chapter explains the method for designing input data for the form style information created during form designing.

---

### **Chapter 9 Defining the Form Output Methods**

---

This chapter explains the method for defining form output methods regarding the output method of the form, operation during output, and operation after output.

---

### **Chapter 10 Designing the Combination-Form**

---

This chapter explains the Combination-Form descriptor, the input data designing, and some important points etc. required to perform "Combination-Form Output" that combines multiple pieces of form descriptor information in one page.

---

### **Chapter 11 Replacing Properties when Outputting Forms**

---

This chapter explains the overview of property replacement, how to use this function, and details of the property replace file.

## **Part 3 Management and Maintenance**

This part explains the functions for managing and maintaining the form descriptor information created in the Part 2. Refer to it at the time of managing or correcting the created form descriptor information.

---

### **Chapter 12 Managing Form Descriptor Information**

---

This chapter explains the method for managing created form descriptor information.

---

### **Chapter 13 Maintaining Form Descriptor Information**

---

This chapter explains the method for maintaining created form descriptor information.

## **Part 4 Overlay Utility**

This part explains the procedures for creating overlays and forms based on PDF files and other form images and for adding created overlays and forms to List Creator. Refer to it at the time of using List Creator Overlay Utility.

---

### **Chapter 14 The Overlay Utility**

---

This chapter explains the overview of the Overlay Utility.

---

### **Chapter 15 Installation**

---

This chapter explains the installation and environment settings of the Overlay Utility.

---

### **Chapter 16 Overlay/Form Creation**

---

This chapter explains the flow of operation when an overlay or form is created.

---

### **Chapter 17 Points to Note**

---

This chapter explains the cautions when using the Overlay Utility.

## Appendixes

These appendixes explain the supplementary information, cautions, and related functions for designing forms in List Creator.

Refer to the cautions before designing forms.

### Appendix A Graph Output

This appendix explains the graph output function summary, graph output method, and cautions.

### Appendix B Cautions for Designing Forms

This appendix explains the following cautions for designing forms.

- Cautions for designing forms
- Cautions for each output method
- Cautions when designing forms for multi-form output
- Cautions for importing fields in a file
- Cautions regarding environment for designing forms
- Cautions when using Unicode

### Appendix C Cautions regarding Input Data

This appendix explains the actions when changing input data format, and the cautions for each character code when designing forms or input data.

### Appendix D XML Files Used in List Creator

This appendix explains the terms related to XML files used in List Creator, the conditions under which XML files and DTD files can be used in List Creator, and the cautions for outputting XML format data.

### Appendix E Combination-Form Descriptor File

This appendix explains the description format of combination-form descriptor files.

### Appendix F COPY Clause File Creation Function

This appendix explains the COPY clause file creation function.



#### Note

- For details and the cautions for using the following functions, refer to the Japanese Online Manuals.
  - Output Methods
    - e-Record
    - Sending FAX
  - Upper Layer Tool/Database Link Functions
    - Navigator links function
    - XBRL data support function
- In this manual, Windows path separator is shown as "¥" (yen sign). If you operate this software on the English environment, please read it as a backslash.

## Copyright

Screens are reproduced according to Microsoft Corporation guidelines.

Copyright 2012 FUJITSU LIMITED

Copyright PFU LIMITED 2012

## *About the List Creator Manuals*

---

Before reading this manual, please refer to the Online Manual "Documentation Road Map".

In the "Documentation Road Map", the product manuals for List Creator, usage of the manuals, manual conventions, target audience, required knowledge, glossary, and trademarks are described.

# Contents

## Part 1 Getting Started with Advanced Form Designing ..... 14

### Chapter 1 Basic Knowledge of Form Designing ..... 15

|         |  |    |
|---------|--|----|
| 1.1     | What is Form Designing in List Creator? .....                        | 16 |
| 1.2     | Information Defined during Form Designing .....                      | 17 |
| 1.2.1   | Form Descriptor Information.....                                     | 17 |
| 1.2.1.1 | Form Style Information.....  | 17 |
| 1.2.1.2 | Form Output Methods.....   | 18 |
| 1.3     | Components of Form Layout Data.....                                  | 19 |
| 1.3.1   | Data for Areas (Blocks).....   | 19 |
| 1.3.1.1 | Areas for Each Form Format.....                                      | 19 |
| 1.3.1.2 | Partition (Frame partition) and Frame.....                           | 20 |
| 1.3.2   | Data for Components (Objects) .....                                  | 21 |
| 1.3.2.1 | Fields.....  | 21 |
| 1.3.2.2 | Overlays (Overlay Graphic and Overlay Text).....                     | 24 |
| 1.3.2.3 | Partition Graphics (Partition Graphic and Partition Rectangle) ..... | 24 |
| 1.4     | What is Form Format? .....   | 25 |
| 1.4.1   | Free Frame Format.....   | 25 |
| 1.4.2   | Single Page Format.....  | 29 |
| 1.4.3   | Simple Table Format.....   | 30 |
| 1.4.4   | Multipart Table Format.....  | 31 |
| 1.4.5   | Block Repeat Format .....  | 33 |
| 1.4.6   | Sticky Label Format .....  | 34 |
| 1.5     | Input Data Type and Format .....                                     | 36 |
| 1.5.1   | Data File .....  | 36 |
| 1.5.1.1 | Variable Length .....  | 37 |
| 1.5.1.2 | Fixed Length.....  | 37 |
| 1.5.1.3 | CSV Format.....  | 38 |
| 1.5.1.4 | XML Format .....   | 38 |
| 1.5.2   | Linking with a Database .....  | 39 |
| 1.5.2.1 | Navigator Links Function .....                                       | 39 |
| 1.5.2.2 | SSIS Link Function .....   | 39 |
| 1.5.3   | Linking with XBRL Data.....  | 40 |
| 1.5.4   | Linking with ERP .....   | 40 |
| 1.5.4.1 | SAP R/3 RDI Integration .....  | 40 |

|   |               |
|---|---------------|
| 1.5.4.2 Oracle E-Business Suite Integration .....                               | 40            |
| 1.6 Created Files .....   | 41            |
| <b>Chapter 2 Form Designing Flow .....</b>                                      | <b>42</b>     |
| 2.1 Form Designing Flow .....   | 43            |
| 2.1.1 Preparing for Form Designing .....  | 44            |
| 2.1.2 Starting the [Form Descriptor List].....                                  | 44            |
| 2.1.3 Creating Form Style Information .....                                     | 44            |
| 2.1.4 Checking Output Image of Form.....  | 45            |
| 2.1.5 Designing (Preparing) Input Data .....                                    | 45            |
| 2.1.6 Setting Form Output Methods.....  | 45            |
| 2.1.7 Saving Form Descriptor Information.....                                   | 45            |
| <b>Chapter 3 Windows and Operating Environment used for Form Designing ....</b> | <b>46</b>     |
| 3.1 Windows Used for Form Designing.....  | 47            |
| 3.1.1 [Form Descriptor List] Window.....  | 47            |
| 3.1.2 [Form Style Descriptor] Window.....                                       | 48            |
| 3.1.2.1 Menubar .....   | 48            |
| 3.1.2.2 Toolbar.....  | 48            |
| 3.1.2.3 Forms Navigation.....   | 49            |
| 3.1.2.4 List Pane .....   | 49            |
| 3.1.2.5 Property List.....  | 52            |
| 3.1.2.6 Quick Handle .....  | 52            |
| 3.1.2.7 Edit Area .....   | 52            |
| 3.1.2.8 Status Bar .....  | 54            |
| 3.2 Operating Environment for Form Designing.....                               | 55            |
| 3.2.1 Layout Creation and Layout Preview Windows .....                          | 55            |
| 3.2.2 Customizing Operating Environment.....                                    | 56            |
| 3.3 Grid Function .....   | 58            |
| 3.3.1 Fixed-Spacing Grid (Default).....   | 58            |
| 3.3.2 Custom-Spacing Grid .....   | 59            |
| 3.3.3 Snap to Grid.....   | 59            |
| <br><b>Part 2 Form Designing .....</b>  | <br><b>60</b> |
| <b>Chapter 4 Preparing for Form Designing .....</b>                             | <b>61</b>     |
| 4.1 Determining the Form Format .....   | 62            |
| 4.1.1 Criteria for Selecting a Form Format.....                                 | 62            |
| 4.1.2 Information Required for Designing Forms in each Format .....             | 63            |
| 4.1.2.1 Information Required for Free Frame Format .....                        | 63            |

|  |   |            |
|--|---|------------|
| 4.1.2.2  | Information Required for Single Page Format .....   | 63         |
| 4.1.2.3  | Information Required for Simple Table Format.....   | 64         |
| 4.1.2.4  | Information Required for Multipart Table Format .....   | 64         |
| 4.1.2.5  | Information Required for Block Repeat .....   | 64         |
| 4.1.2.6  | Information Required for Sticky Label Format.....   | 64         |
| 4.2  | Determine Whether to Use Unicode characters .....   | 65         |
| 4.3  | Determining the Methods for Creating Form Layout Data.....  | 66         |
| 4.3.1  | Using Layout Data from an Existing Form or Preprinted Paper .....   | 66         |
| 4.3.2  | Using Input Data Field Information.....   | 67         |
| 4.3.3  | Using DTD/XML File .....  | 68         |
| 4.3.4  | Using XBRL Data Support Function/Navigator Links Function.....  | 68         |
| 4.4  | Preparing the Required Files.....   | 69         |
| 4.4.1  | Creating an Image File .....  | 69         |
| 4.4.2  | Creating Field Info Files.....  | 69         |
| 4.4.3  | Using Schema File (schema.ini) .....  | 73         |
| 4.4.4  | Preparing DTD/XML File .....  | 74         |
| <b>Chapter 5 Creating Form Layout Data.....</b>      |   | <b>75</b>  |
| 5.1  | Creating Form Layout Data .....   | 76         |
| 5.1.1  | Using Field Entry List .....  | 77         |
| 5.1.1.1  | Creating Form Layout Data in Free Frame Format .....  | 77         |
| 5.1.1.2  | Creating Form Layout Data for Other Formats .....   | 78         |
| 5.1.2  | Using Layout Data from Existing Form or Preprinted Paper .....  | 82         |
| 5.1.3  | Using Field Information of Input Data .....   | 86         |
| 5.1.4  | Using DTD/XML file .....  | 89         |
| 5.1.4.1  | Using DTD/XML File.....   | 90         |
| 5.1.4.2  | Extracting to Data List.....  | 95         |
| 5.1.4.3  | Data Structure of Form Layout Data and XML Input Data.....  | 97         |
| 5.1.4.4  | Changing Data List Structure (for Free Frame Format Only)..   | 98         |
| <b>Chapter 6 Definition of Association Data.....</b> |   | <b>101</b> |
| 6.1  | Definition of Association Data .....  | 102        |
| 6.2  | Selecting Information to be Defined for Association Data .....  | 104        |
| 6.2.1  | Determining the Type of Input Data.....   | 104        |
| 6.2.2  | Checking Fields and Contents to which Operators are Specified....   | 104        |
| 6.2.2.1  | Types of Operators .....  | 105        |
| 6.2.3  | Selecting the Output Order and Output Conditions of Frames and<br>Partitions, and Timing of Calculations..... | 108        |
| 6.3  | Defining Association Data .....   | 109        |
| <b>Chapter 7 Confirming the Output Image .....</b>   |   | <b>111</b> |
| 7.1  | Layout Print.....   | 112        |
| 7.2  | Preview .....   | 113        |



|                   |  |            |
|-------------------|--|------------|
| 7.2.1             | Previewing when Designing a Form.....  | 114        |
| 7.2.2             | Previewing with Utility (Preview) .....  | 118        |
| 7.3               | Test Print .....   | 120        |
| <b>Chapter 8</b>  | <b>Designing Input Data .....</b>  | <b>121</b> |
| 8.1               | Checking Input Data Reports .....  | 122        |
| 8.2               | Designing the Input Data based on Input Data Reports .....   | 124        |
| 8.2.1             | Understanding the Data Configuration .....   | 124        |
| 8.2.1.1           | How a Record of Input Data is Handled for each<br>Form Format .....  | 125        |
| 8.2.1.2           | How Input Data in Free Frame Format is Handled .....   | 127        |
| 8.2.2             | Understanding the Data Format for each Field .....   | 132        |
| 8.2.3             | Understanding the Data Length of each Field .....  | 140        |
| 8.2.4             | Sorting the Data.....  | 141        |
| 8.3               | Designing Input Data for Multi-form Output.....  | 142        |
| 8.3.1             | When the Input Data Format is "Variable Length" or<br>"CSV Format" .....   | 142        |
| 8.3.2             | When the Input Data Format is "Fixed Length" .....   | 143        |
| 8.3.3             | When the Input Data Format is "XML Format" .....   | 145        |
| 8.3.4             | Points .....   | 146        |
| 8.4               | Designing a Data File with the Field Name Label .....  | 148        |
| 8.4.1             | Definition of a Data File with Field Name Label.....   | 148        |
| 8.4.2             | Description Format of a Data File with the Field Name Label.....   | 148        |
| 8.4.3             | When Outputting a Single Form .....  | 149        |
| 8.4.4             | When Performing Multi-form Output .....  | 149        |
| 8.4.5             | When Using the Single Page Format in which Repeats are<br>Specified for Fields .....                               | 150        |
| 8.4.6             | When Using the Multipart Table Format in which a Report Header<br>(RH) Partition is Defined .....                  | 150        |
| 8.4.7             | Specifying Operation when Excess or Shortage of Sata is Detected<br>in a Data File with the Field Name Label ..... | 151        |
| 8.5               | New Page/New Frame of Form .....   | 153        |
| 8.5.1             | Page Feed Timing.....  | 153        |
| 8.5.2             | New Frame Timing .....   | 154        |
| <b>Chapter 9</b>  | <b>Defining the Form Output Methods.....</b>   | <b>156</b> |
| <b>Chapter 10</b> | <b>Designing the Combination-Form .....</b>  | <b>158</b> |
| 10.1              | Designing the Combination-Form.....  | 159        |
| 10.1.1            | Combination-Form Defining Flow.....  | 159        |
| 10.1.2            | Designing the Combination-Form .....   | 160        |
| 10.1.2.1          | Display the Combination-Form Descriptor list.....  | 160        |
| 10.1.2.2          | Create the Combination-Form Descriptor .....   | 160        |

---

|                   |  |            |
|-------------------|--|------------|
| 10.1.3            | Checking the Output Image of Combination-Form .....                      | 162        |
| 10.1.3.1          | Previewing the Combination-Form.....                                     | 162        |
| 10.1.3.2          | Edit the Form Style Descriptor .....                                     | 163        |
| 10.1.4            | Save the Combination-Form Descriptor .....                               | 163        |
| 10.1.5            | Cautions for Designing a Combination-Form.....                           | 164        |
| 10.2              | Designing Input Data for Combination-Form.....                           | 167        |
| 10.2.1            | When the Input Data Format is "Variable Length" or<br>"CSV Format" ..... | 167        |
| 10.2.2            | When the Input Data Format is "XML Format" .....                         | 168        |
| 10.2.3            | Points .....   | 170        |
| 10.2.4            | Page Feed Timing.....  | 171        |
| <b>Chapter 11</b> | <b>Replacing Properties when Outputting Forms .....</b>                  | <b>172</b> |

---

|        |   |     |
|--------|---|-----|
| 11.1   | What is "Replacing Properties"? .....             | 173 |
| 11.1.1 | Properties that Can Be Replaced.....              | 174 |
| 11.1.2 | Points to Note when Replacing Properties.....     | 175 |
| 11.2   | Procedure for Using "Property Replacement" .....  | 178 |
| 11.2.1 | Creating Format Property Replace File.....        | 179 |
| 11.2.2 | Creating Property Replace File .....              | 182 |
| 11.2.3 | Checking Output Image of Form.....                | 182 |
| 11.3   | Property Replace File.....                        | 183 |
| 11.3.1 | Description Format of Property Replace File ..... | 184 |
| 11.3.2 | Format of Format Property Replace File.....       | 191 |

## **Part 3 Management and Maintenance ..... 195**

### **Chapter 12 Managing Form Descriptor Information ..... 196**

---

|        |   |     |
|--------|---|-----|
| 12.1   | About Form Descriptor Information Management.....                                 | 197 |
| 12.2   | Listing Form Descriptor Information .....   | 198 |
| 12.3   | Specifying the Form Storage Directory .....                                       | 198 |
| 12.4   | Creating, Opening, Copying, Deleting, and Renaming Form Style<br>Information..... | 199 |
| 12.5   | Setting and Viewing Form Output Methods.....                                      | 199 |
| 12.6   | Checking the Output Image of the Form Style Information.....                      | 199 |
| 12.7   | Outputting the List of Form Descriptor Information .....                          | 200 |
| 12.7.1 | Identifying Changes from Output Results .....                                     | 202 |
| 12.8   | Printing Reports of Form Style Information.....                                   | 205 |
| 12.9   | Creating a Format Property Replace File .....                                     | 206 |

---

|                   |   |            |
|-------------------|---|------------|
| <b>Chapter 13</b> | <b>Maintaining Form Descriptor Information .....</b>      | <b>207</b> |
| 13.1              | Changing the Form Style Information .....                 | 208        |
| 13.2              | Changing the Form Output Methods .....                    | 210        |
| <br>              |   |            |
| <b>Part 4</b>     | <b>Overlay Utility .....</b>                              | <b>211</b> |
| <br>              |   |            |
| <b>Chapter 14</b> | <b>The Overlay Utility .....</b>                          | <b>212</b> |
| 14.1              | The Overlay Utility .....                                 | 213        |
| 14.2              | Features of the Overlay Utility.....                      | 214        |
| 14.2.1            | Overlay Creation Efficiency.....                          | 214        |
| 14.2.2            | Form Design using Your Favorite Tools .....               | 214        |
| 14.2.3            | Forms Published on the Internet .....                     | 214        |
| <br>              |   |            |
| <b>Chapter 15</b> | <b>Installation .....</b>                                 | <b>215</b> |
| 15.1              | Installation.....   | 216        |
| 15.1.1            | Installing from Product Media .....                       | 216        |
| 15.2              | Uninstall .....   | 219        |
| 15.3              | Setup.....  | 220        |
| 15.3.1            | Printer Definition in Windows .....                       | 220        |
| 15.3.2            | Setup (Port Definition Window) .....                      | 224        |
| <br>              |   |            |
| <b>Chapter 16</b> | <b>Overlay/Form Creation .....</b>                        | <b>228</b> |
| 16.1              | Form Printing Operation .....                             | 229        |
| 16.2              | Output Settings Window .....                              | 232        |
| 16.3              | Completion of Overlay/Form Creation.....                  | 234        |
| 16.4              | Adding the Overlay to List Creator .....                  | 235        |
| 16.4.1            | Adding Overlay .....                                      | 235        |
| 16.4.2            | Adding Forms .....  | 236        |
| <br>              |   |            |
| <b>Chapter 17</b> | <b>Points to Note.....</b>                                | <b>238</b> |
| 17.1              | Points to Note during Printing Forms.....                 | 239        |
| 17.2              | Points to Note regarding Overlay Conversion.....          | 240        |
| 17.2.1            | Points to Note regarding Line Recognition .....           | 240        |
| 17.2.1.1          | Line that can be Recognized.....                          | 240        |
| 17.2.1.2          | Lines which cannot be Recognized .....                    | 241        |
| 17.2.2            | Points to Note regarding Recognizing the Color Form ..... | 242        |
| 17.2.2.1          | Color that can be Recognized.....                         | 242        |

---

## **Appendixes ..... 244**

### **Appendix A Graph Output ..... 245**

---

### **Appendix B Cautions for Designing Forms ..... 246**

---

- B.1 Cautions for Designing Forms ..... 247**
  - B.1.1 Cautions regarding Specification of Form Name ..... 247**
  - B.1.2 Cautions When Changing the Grid Type..... 247**
  - B.1.3 Cautions regarding Custom-Spacing Grid..... 247**
  - B.1.4 Cautions when defining Overlay Lines and Partition Graphics ..... 248**
  - B.1.5 Cautions regarding the Transparency of Image Fields..... 248**
  - B.1.6 Output Order of Parts (Objects) ..... 249**
- B.2 Cautions for Each Output Method..... 251**
- B.3 Cautions when Designing Forms for Multi-Form Output..... 252**
- B.4 Cautions for Importing Fields in a File ..... 253**
- B.5 Cautions regarding Environment for Designing Forms ..... 254**
  - B.5.1 Cautions regarding Access Privileges ..... 254**
  - B.5.2 Cautions regarding the Location (Source) of Form Resources ..... 254**
- B.6 Cautions when using Unicode..... 255**

### **Appendix C Cautions regarding Input Data ..... 256**

---

- C.1 Operation when Changing the Input Data Format ..... 257**
  - C.1.1 When Changing from Variable Length, Fixed Length, or CSV Format to XML Format ..... 257**
  - C.1.2 When Changing from XML Format to Variable Length, Fixed Length, or CSV Format..... 258**
- C.2 Cautions for Code Type ..... 259**
  - C.2.1 When the Input Data is Shift-JIS ..... 259**
  - C.2.2 When the Input Data is UNICODE (UTF16) ..... 259**
  - C.2.3 When the Input Data is UNICODE (UTF8) ..... 260**
  - C.2.4 When the Input Data is EUC ..... 260**

### **Appendix D XML Files Used in List Creator..... 261**

---

- D.1 Terms regarding XML Format Files..... 262**
  - D.2 Conditions under which XML Files and DTD Files can be Used in List Creator ..... 263**
    - D.2.1 Character codes of XML Files and DTD Files that can be Used in List Creator ..... 263**
    - D.2.2 When using an XML File ..... 263**
    - D.2.3 When using a DTD File..... 265**
-

---

|  |   |            |
|--|---|------------|
| D.2.4  | Range of Input Data for each form format (per Page, per Detail).....                | 266        |
| D.3  | Specifying the Form Format responding to the XML Data Configuration...              | 267        |
| D.3.1  | If there is only Page Worth of Data or If the Data is in<br>Single Page Units ..... | 267        |
| D.3.2  | If there are Multiple Details of Data (an Undetermined Amount of<br>Data) .....     | 269        |
| D.4  | Conditions under which Elements can be Specified as a Data Root.....                | 275        |
| D.5  | Conditions under which Elements can be Specified as<br>Body Group Items .....       | 276        |
| D.6  | Conditions under which Elements can be Specified as<br>Details Group Items .....    | 278        |
| D.7  | Conditions under which Elements can be Specified as<br>Frame Group Items .....      | 281        |
| D.8  | Data Root and Page Feed.....  | 284        |
| D.9  | Points .....  | 285        |
| D.9.1  | When Designing Forms.....   | 285        |
| D.9.2  | When Designing Input Data .....   | 288        |
| <b>Appendix E</b> Combination-Form Descriptor File .....   |   | <b>289</b> |
| <hr/>  |   |            |
| E.1  | Format of Combination-Form Descriptor File .....                                    | 290        |
| <b>Appendix F</b> COPY Clause File Creation Function ..... |   | <b>292</b> |
| <hr/>  |   |            |
| F.1  | COPY Clause File Creation Function .....  | 293        |
| <b>Index</b> .....   |   | <b>294</b> |

---

# ***Part 1 Getting Started with Advanced Form Designing***

---

- Before reading [Part 1...](#)

"[Part 1 Getting Started with Advanced Form Designing](#)" explains the basic knowledge required for form designing in List Creator, the form designing flow as well as the windows and operating environment used for form designing.

# **Chapter 1**

## *Basic Knowledge of Form Designing*

---

This chapter explains the followings as a basic knowledge to design forms in List Creator.

- What is List Creator form designing?
- Information defined during form designing
- Components of form layout data
- What is form format?
- Input data type and format
- Files created in form designing

## 1.1 What is Form Designing in List Creator?

"Form" in List Creator is a combination of data that is required to output forms. The following data should be prepared in order to output forms in List Creator.

- Form descriptor information
- Input data
- User applications
- Form resources required for form output, such as media data and information files

"Form Designing" in List Creator is a process for creating "form descriptor information" from the required information indicated above for form output.

Form descriptor information is the information to define a form layout, an output method of data, etc., and is created with List Creator Designer.

For details about form descriptor information, refer to the following:

=> ["1.2.1 Form Descriptor Information"](#)

In addition, List Creator forms can be classified by type and function. This is called "Form Format". When designing and outputting forms with List Creator, it is necessary to understand the features of each format.

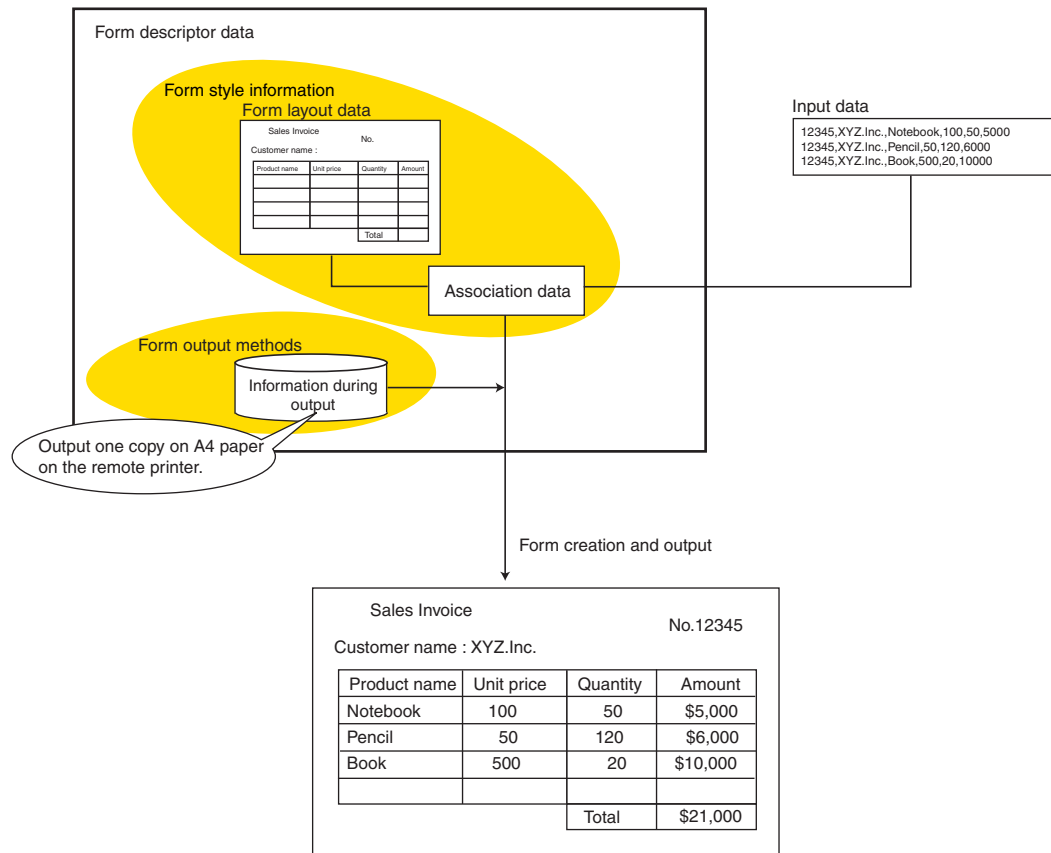
For details about form format, refer to the following:

=> ["1.4 What is Form Format?"](#)



## 1.2 Information Defined during Form Designing

During form designing, form descriptor information required to output forms is created.



**Figure 1.1** Information Defined during Form Designing

For input data details, refer to the following:

=> "[1.5 Input Data Type and Format](#)"

This chapter explains the form descriptor information.

### 1.2.1 Form Descriptor Information

Form descriptor information defines the format of a form that is designed using List Creator Designer, and the operation when the form is output. It consists of the followings.

- Form style information
- Form output methods

The following explains form style information and form output methods.

#### 1.2.1.1 Form Style Information

This information consists of "form layout data" and "association data for form layout data and input data". The definition of form style information is referred to as "form style descriptor".

- **Form layout data**

Form layout information is related to form appearance and layout.

This information is made up of a variety of components.

The components that make up form layout data, refer to the following:

=> "[1.3 Components of Form Layout Data](#)"

It should be noted that the components for form layout information will vary according to form format.

For details on form format, refer to the following:

=> ["1.4 What is Form Format?"](#)

- **Association Data for Form Layout Data and Input Data**  
(also abbreviated as association data)

This information defines how to output input data according to form layout information.

"Data Definitions" and "Extended Definitions" are defined as association data for form layout and input data.

#### Data definitions

In the [Data Definitions], you can specify the association of the fields defined in [Form Style Descriptor] and the input data, calculations such as functions, and so on.

Calculations such as the following can be specified:

- BREAK control
- Media data type of each image field
- String Operation
- Condition Judgment
- Functions (e.g. four arithmetic expressions and sum total)
- Settings for page numbers and total number of pages

Data definitions are specified on the Data Definitions Screen when defining form style information.

#### Extended definitions

Used for specifying the output conditions for partitions or the timing of calculations and defining extensions only when a form in multipart table format or free frame format (with frames defined) is used.

Extended definitions are specified on the [Extended] tab of the Data Definitions Screen when defining form style information.

Record information can also be defined as association data for input data.

Record information definitions include the specification of output conditions for defined fields, such as repeat property definitions for input data and definitions for multiple fields with the same name.

Record definitions are specified in field properties according to form layout data definitions.

### 1.2.1.2 Form Output Methods

Form output methods define the attributes (properties) and operations at output.

It defines the information for form output methods (printing, saving to PDF file, outputting to Excel file, etc.) and operations at output for each output method.

The following is an example of this kind of information.

- Printing three copies of each form at a printer
- Sending a form as a FAX to a branch office
- Saving a form as a PDF file

## 1.3 Components of Form Layout Data

Form layout data includes data for blocks (areas) and data for components (objects) placed in the blocks. The following sections explain the components of the form layout.

### 1.3.1 Data for Areas (Blocks)

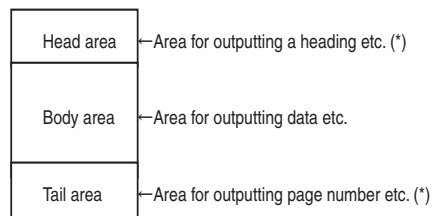
In List Creator, a form consists of various areas. The types, number, and name of the areas are different depending on the form format.

#### 1.3.1.1 Areas for Each Form Format

The following explains the areas (blocks) for each form format:

- **Simple table/block repeat formats**

These formats are made up of three main areas: "head area", "body area", and "tail area". When defined, these areas are always output.



(\*) These areas can be omitted.

- **Single page/sticky label formats**

The single page format and sticky label format are made up of only a single area for a body area.

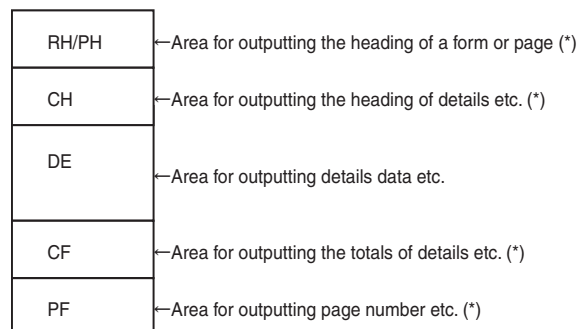
- **Multipart table format**

This format is made up of areas called "partitions".

A partition is an area for which output timing and output conditions can be specified.

There are various types of partitions with specific features and functions.

Partitions



(\*) These areas can be omitted.

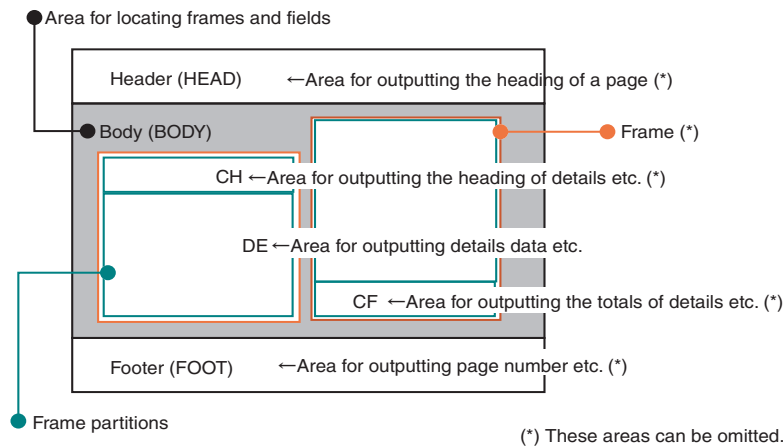
For details on partitions, refer to the following:

=> ["1.3.1.2 Partition \(Frame partition\) and Frame"](#)

- Free Frame format

This format is made up of "header (HEAD)", "body (BODY)", and "footer (FOOT)" partitions.

You can also define an area called "frame" in the body and define a partition called "frame partition" in the frame.



For details on frames and frame partitions, refer to the following:

=> ["1.3.1.2 Partition \(Frame partition\) and Frame"](#)

### 1.3.1.2 Partition (Frame partition) and Frame

A partition is an area that can be defined when the form format is multipart table or free frame format.

This is an area where objects are placed, and you can control outputs on the area by specifying whether or not to output the area and specifying output conditions.

A free frame has "header (HEAD)", "body (BODY)", and "footer (FOOT)" partitions. In the body (BODY) partition for defining fields, you can define an area called frame.

In a defined frame, you can define another partition (frame partition). Therefore, unlike multipart table format, you can define a multiple-column layout by outputting details in multiple frames.

Partitions are again of two types, one which are output in a fixed position usually (fixed partitions) and the other type of partition whose positions change according to the output location or size (floating partitions).

For details on partitions, frames, and frame partitions, refer to the List Creator Designer Help guide or the following:

=> ["1.4 What is Form Format?"](#)



- In this manual, partitions in multipart table format, partitions in free frame format (header, body and footer), and frame partitions are referred to as "partitions" if not specified otherwise.
- By defining "partition graphics" in a partition, you can output defined lines or rectangles only when the partition is output. For partition graphics, refer to the following:  
=> ["1.3.2.3 Partition Graphics \(Partition Graphic and Partition Rectangle\)"](#)

## 1.3.2 Data for Components (Objects)

In form layout, "objects" are data that can be defined in an area (Object).

Objects contain the followings:

- Field

A field is the area defined in form style information for outputting data, fixed string, or image.

- Overlays

An overlay is a graphic (e.g., a table rectangle) or a string (output at a fixed position) defined in form style information.

The defined overlay is always output at the defined position regardless of the settings of partitions.

- Partition graphic

Partition graphics defined in a partition are output only when the partition is output.

There are two types of partition graphics: partition graphic and partition rectangle.

The following explains the details of each object.

### 1.3.2.1 Fields

A field is an area for outputting the data, fixed strings, or images.

The following fields can be defined for form layout data.

#### Numeric field

This field is used for outputting numeric data.

Over 50 types of patterns can be edited and output. Signs, currency symbols, separator characters, and decimal points can be set.

Over 20 different types of 8-digit date formats for western and Japanese calendar can be edited and output.

#### Text field

This field is used for outputting string data in one line.

7-digit numeric strings can be edited and output as post codes.

#### Block text field

This field is used for outputting string data in multiple lines.

#### OCR-B field

This field is used for outputting alphanumeric data using an OCR-B font.



**Note**

- When OCR-B fields are output on a Windows List Creator print server, the "OCR-B FJ 10cpi (True Type)" font, which was registered to the system when installing List Creator, is used. If you want to use a different OCR-B font, define the text field, and then set the font name to the OCR-B font you want to use.

#### Radio button

This field is used for making a single selection from the choices available.

#### Check box

This field is used for toggling an option or for selecting one or more options, by checking/unchecking it.

#### Barcode field

This field is used for outputting a barcode.



**Caution**

- Barcodes may be output larger or smaller than the specified range. Also, since the paper, reader, the resolution of the output device, and other conditions affect the quality of the barcode, test print the barcode and check whether it can be read before actual operation.

The following barcode types can be output.

- JAN Standard
- JAN Reduced

- Code 3 of 9
- Code 3 of 9 (EIAJ Compliant)  
Code 3 of 9 (EIAJ Compliant) is compliant with the following standard:
  - "EIAJ-EDI Standard" (2001)The printable range by Code 3 of 9 (EIAJ Compliant) covers C-3 style labels, D style labels, standard delivery statements, and standard return forms in a single column format.
- Industrial 2 of 5
- Interleaved 2 of 5
- NW-7
- Custom
- CODE-128
- EAN-128
- EAN-128 (Convenience store)  
Compliant with "Guideline for Tax and Payment Collection Agencies via UCC/EAN-128".
- UPC version A (UPC-A)
- UPC version E (UPC-E)
- EAN-13
- U.S. POSTNET (Delivery Point Code)
- U.S. POSTNET (ZIP+4 Code)
- U.S. POSTNET (5-Digit ZIP Code)
- QR Code (Model 1)
- QR Code (Model 2)
- QR Code (Micro QR)
- PDF417
- MaxiCode
- Intelligent Mail Barcode
- GS1 DataBar Omnidirectional
- GS1 DataBar Truncated
- GS1 DataBar Stacked
- GS1 DataBar Stacked Omnidirectional
- GS1 DataBar Limited
- GS1 DataBar Expanded
- GS1 DataBar Expanded Stacked

#### Barcode field (U.S. Postal FIM)

This field is used for outputting barcodes (U.S. Postal FIM).

The following barcode types (U.S. Postal FIM) can be output:

- U.S. Postal FIM (FIM A)
- U.S. Postal FIM (FIM B)
- U.S. Postal FIM (FIM C)

#### Image field

This field is used for outputting media data.

The following media data types can be output:

- Bitmap data
- JPEG data
- TIFF data
- OLE2 object
- PNG data
- Graph data



**Note**

- Transparency of background color can be specified for image fields. For details, refer to the Designer Help guide.
- For details regarding the available media data, refer to the Online Manual "Setup & Operation".
- For details of graph data, refer to the following:  
=> ["Appendix A Graph Output"](#)

#### Static field

This field is used for outputting fixed strings such as headings and labels in one line.



**Note**

- You can set outputIDs to be output when a form is output. [OutputID] is valid only in List Manager.

#### Block static field

This field is used for outputting fixed strings such as headings and labels in multiple lines.

#### Date field

This field automatically obtains and outputs the date when the form is output.  
Over 20 types of date formats for western and Japanese calendar can be edited and output.  
Dates are acquired for each date field output.

#### Time field

This field automatically obtains and outputs the time when the form is output.  
These can be output using a specified format.  
Times are acquired for each time field output.

#### Post code field

This field is used for outputting a post code to a fixed location of a postcard and an envelope.  
The location for printing a post code for each type of paper is shown below. It can be moved within a range of 7.6 mm - 108.0 mm.

- Postcard, reciprocating postcard, envelopes, etc.: 12.0 mm
- Tyoukei 3 gou: 37.0 mm
- Tyoukei 4 gou: 29.0 mm

Only one Post code field can be defined for single page forms, simple table forms, and block repeat forms. Post code fields are not displayed in the edit area.

#### Hidden field (Record)

This work field is not output to the form. When using numerals, set the field format (record format) to "Numeric"; for strings, set it to "Text".

It is used for the following purposes:

- When values of input data are not directly output to the form, but used in calculations for other fields
- For storing intermediate values of calculations
- As a field for accepting input data that is not intended for output

#### Group item

This is a group of multiple fields that are handled as one field.

### 1.3.2.2 Overlays (Overlay Graphic and Overlay Text)

Overlay is a graphic (line, rectangle, circle, or image) or a string output at a defined (fixed) position.

List Creator supports the following overlays:

- Overlay graphic (graphics)

An overlay graphic is a graphic such as a line, a rectangle, or an image that can be defined for form style information.

You can define various types of lines and rectangles.

You can use an overlay image when you want to import a fixed image such as a company logo mark in a form.

- Overlay text

Fixed strings which can be defined for form style information.

The strings are always output at a defined position regardless of the settings of partitions.

### 1.3.2.3 Partition Graphics (Partition Graphic and Partition Rectangle)

Partition graphic is a graphic such as line and rectangle to be output as part of each partition.

This can be defined when the form format is multipart table or free frame format.

There are two types of partition graphics: partition graphic and partition rectangle.

Unlike overlay graphics, partition graphics are not output when no partition is output.

For details on partitions, frames, and frame partitions, refer to the List Creator Designer Help guide or the following:

=> ["1.3.1.2 Partition \(Frame partition\) and Frame"](#)



## 1.4 What is Form Format?

List Creator forms can be classified by type and function.

This is called "form format", and List Creator supports the following form formats.

- Free Frame format
- Single page format
- Simple table format
- Multipart table format
- Block repeat format
- Sticky label format

When designing and outputting forms with List Creator, it is necessary to understand the features of each format, and decide what format will best serve your purpose.

For the criteria of selecting form format, refer to the following:

=> ["4.1.1 Criteria for Selecting a Form Format"](#)

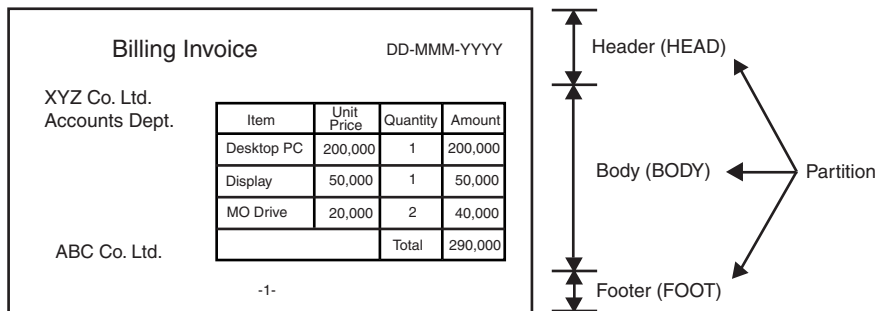
The following describes each form format and its features.

### 1.4.1 Free Frame Format

- What kind of format is this?

On a form in free frame format, you can output multiple details on one page or output details in columns.

The form in free frame format is composed of frames and three partitions: header, body and footer. Fields and overlays can be defined in the header, body and footer.



**Figure 1.2** Free frame format (Structure of partitions)

#### Header

This is the area for outputting data, such as a title. This partition may be omitted.

#### Body

This is the area for defining frames and fields.

#### Footer

This is the area for outputting data, such as a total. This partition may be omitted.

Also, you can define an area called "frame" in any place within the body. Further, you can define partitions in a frame. The partitions defined in a frame are called "frame partitions".



- In this manual, partitions in multipart table format, partitions in free frame format (header, body and footer), and frame partitions are referred to as "partitions" if not specified otherwise.

#### Frame

This is the area defined for outputting frame partitions. Up to 100 frames can be defined in one form. You can link multiple frames to handle them as one frame. This is called "frame links" or "setting frame links" (refer to [Figure 1.6](#)).

Frame partitions can be defined in a frame.

Frame partition

The frame partition has the following types: CH (Control Header), DE (Details), and CF (Control Footer).

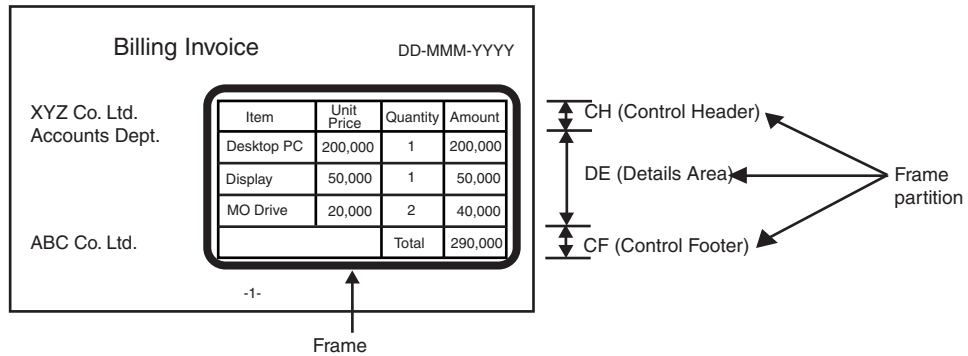


Figure 1.3 Free frame format (Structure of frame partitions)

CH (Control Header)

This is the area for setting fields that are output before details when data for a specified field changes, such as headings of details.

This partition is output to a floating position according to data (floating partition).

A maximum of 16 control header partitions can be defined in one form. This partition may be omitted.

DE (Details)

This is the area for setting detail data.

This partition is output to a floating position according to data (floating partition).

A maximum of 100 details partitions can be defined in one form. At least one details partition must exist in an independent frame.

CF (Control Footer)

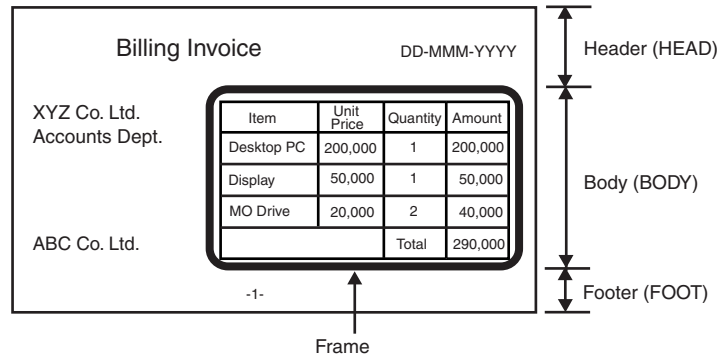
This is the area for setting fields that are output after details when data for a specified field changes, such as a total and subtotal of the details.

This partition can be output to a specified position (fixed partition) or to a floating position according to data (floating partition).

A maximum of 16 control footer partitions can be defined in one form. This partition may be omitted.

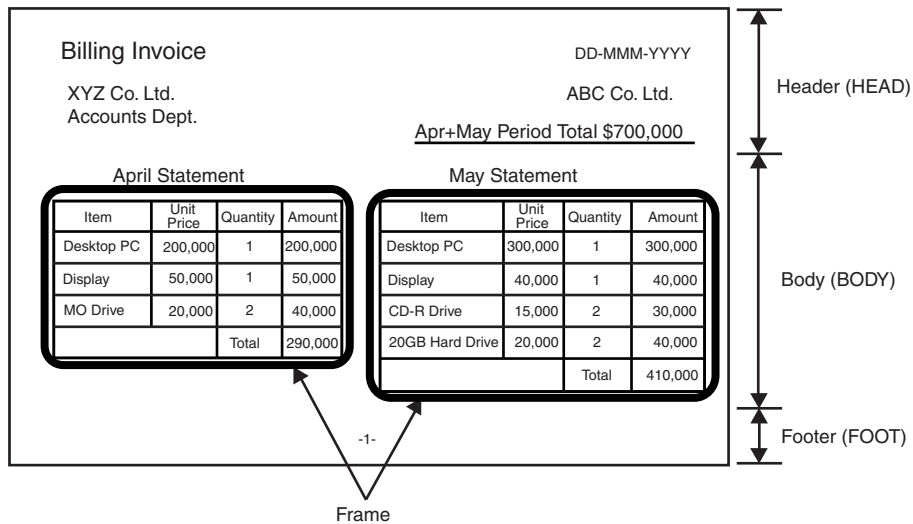
• Features

- Positions of fields to be output are designed in relation to one another in a frame partition. The defined frame partitions are automatically positioned and output according to data.
- You can create a form that has details on the right side of the body and headings on the left side.



**Figure 1.4** Free frame format (Output example with details on the right side of the body and headings on the left)

- You can define multiple frames in order to output multiple sets of details with different contents to a one-page form.



**Figure 1.5** Free frame format (Output example with multiple sets of details with different contents in one page)

- You can set frame links in order to output a set of details across multiple frames.

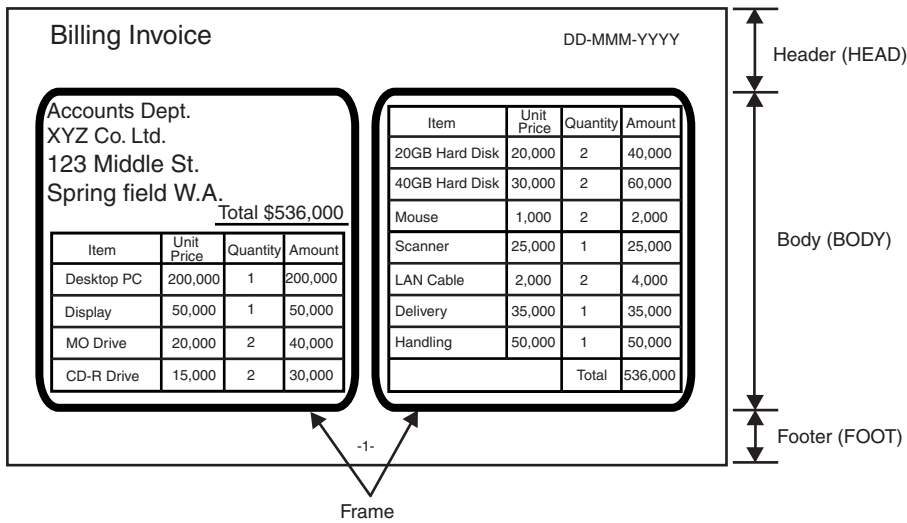


Figure 1.6 Free frame format (Output example with frame links)

- You can output frame partitions horizontally.

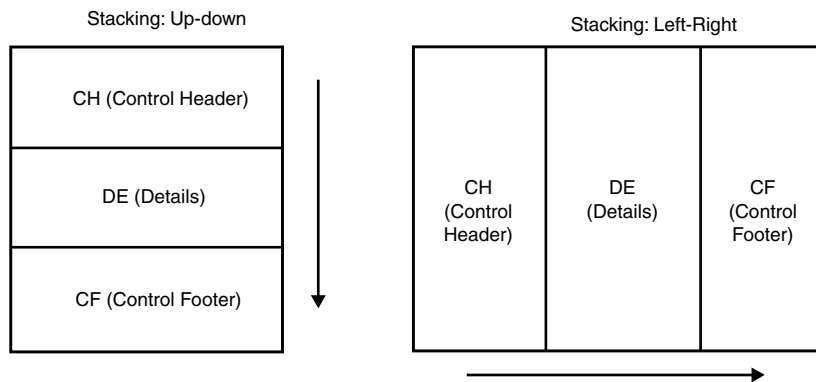


Figure 1.7 Stacking for frame partitions

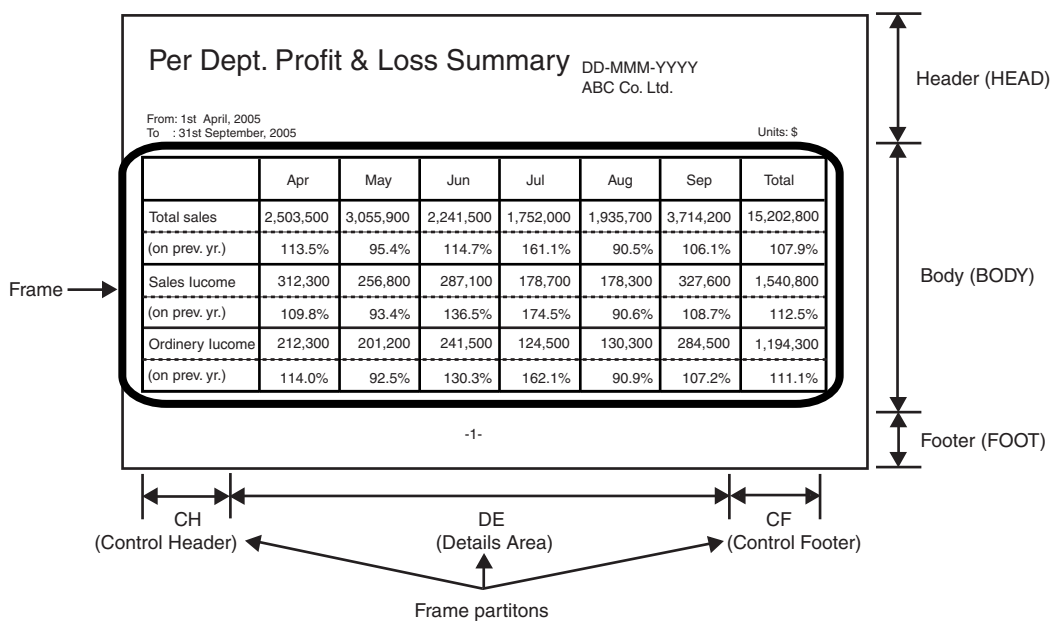


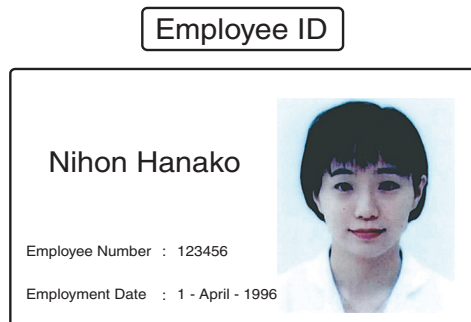
Figure 1.8 Free frame format (Output example of horizontal frame partitions)

- The details partitions to be output can be switched according to field data by specifying conditions on the fields defined in a form.
- Lines/fill that are output to positions relative to each corresponding partition (partition graphic/fill) can be defined. It is also possible to define lines/fill that are output to positions defined at a design phase, independent of the positions of partitions to be output (overlay lines/fill).
- The following input data is specified for each detail:
  - When the input data format is "Variable Length", "Fixed Length", or "CSV Format", one record (line) is specified for each detail.
  - When the input data format is "XML Format", one element of a details group item is specified for each detail.  
For details of the details group item, refer to the following:  
=> ["D.6 Conditions under which Elements can be Specified as Details Group Items"](#)
- Multiple timing settings can be specified for calculations, such as calculating a subtotal when data for a specified field (such as "Month") changes, and calculating a total when data for another field (such as "Year") changes. The timing of calculations is specified by priority (control level) of each task.
- A new page or new frame (when frames are linked) can be created when data for a specified field (such as "Day") changes. The timing of page/frame feed is specified by priority (control level) of each task.

## 1.4.2 Single Page Format

- What kind of format is this?

This format is made up of the following one partition. You can define fields as you like.  
This format is suitable for employee IDs or employee cards that do not contain details.



**Figure 1.9** Single page format

- Features

- A form that is identical to the image defined at a design phase is output.
- The following input data is specified for each page.
  - When the input data format is "Variable Length", "Fixed Length", or "CSV Format", one record (line) is specified for each page.
  - When the input data format is "XML Format", one element of a data root is specified for each page.

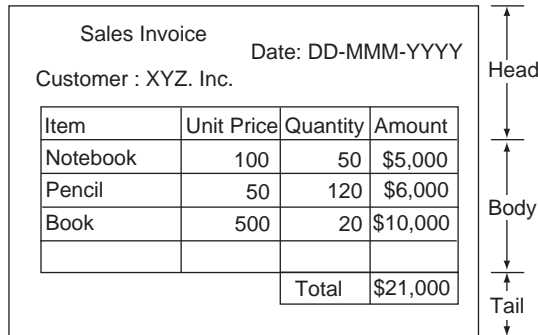
### 1.4.3 Simple Table Format

- What kind of format is this?

This format is made up of the following three partitions.

One type of detail table can be defined.

In this format, a total is automatically calculated and output, which is suitable for sales invoices or bills. When you need different types of totals such as "subtotal" and "total", use free frame format or multipart table format.



**Figure 1.10** Simple table format

#### Header

This is the area for outputting data, such as a title. This partition may be omitted.

#### Body

This is the area for outputting lines, such as sales detail lines, repeatedly downward.

#### Tail

This is the area for outputting data, such as a sales total. This partition may be omitted.

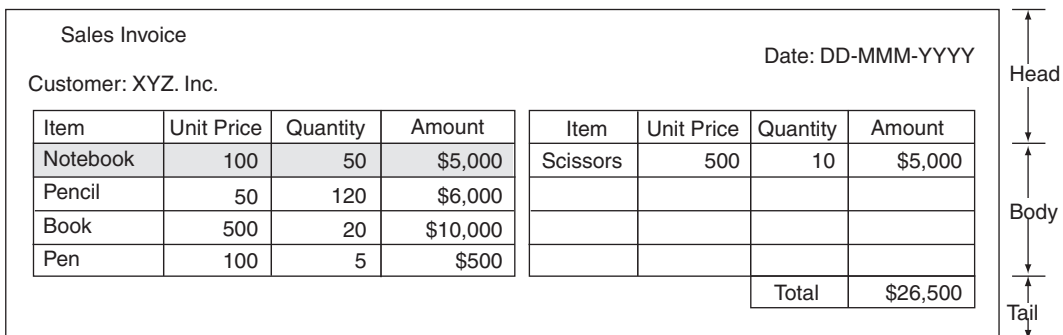
- Features

- A form that is identical to the image defined at a design phase is output.

[Figure 1.11](#) shows that there are lines for eight data items in the body. Even if only five data items are output, the tail (that is, Total) does not move up to fill the space, and the line for sixth item and following lines are output as blank lines.

- The numbers of columns and lines can be specified so that the specified number of lines are arranged.

So in [Figure 1.11](#), the number of columns is set to "2" and that of rows is set to "4".



**Figure 1.11** Simple table format (Settings for the numbers of columns/lines)

- The following input data is specified for each detail in the body.

- When the input data format is "Variable Length", "Fixed Length", or "CSV Format", one record (line) is specified for each detail in the body.
- When the input data format is "XML Format", one element of a body area group item is specified for each detail in the body.

For details of the body area group item, refer to the following:

=> "D.5 Conditions under which Elements can be Specified as Body Group Items"

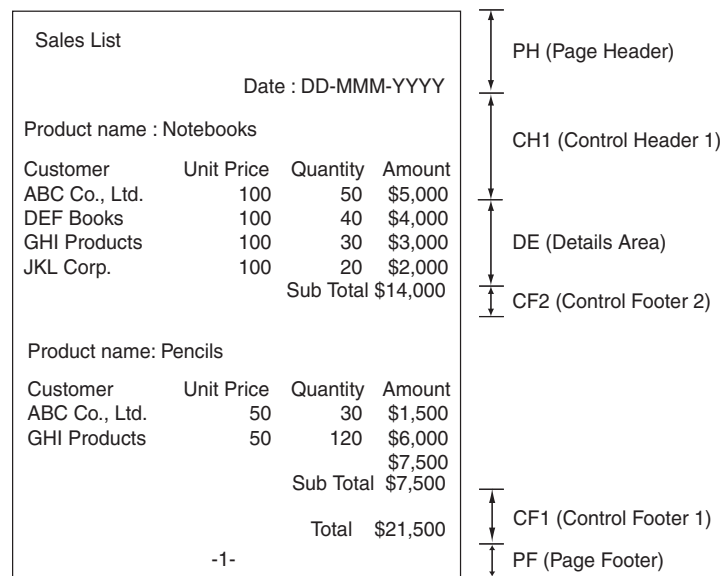
- A new page can be created (issuing one simple table) when data for a specified field (such as "Customer name") changes. Only one field can be specified for this purpose.

**Note** - For example, if you want to perform a calculation, such as a total, for each simple table, and one table spans multiple pages, a cumulative total is output on each page. If you do not want to output a cumulative total on each page, use the multipart table format.

### 1.4.4 Multipart Table Format

- What kind of format is this?

This format is made up of multiple partitions. Depending on data, the number of details to be output may increase or decrease. You can specify timing of page feed and calculations, and output positions in detail. This format is suitable for multipart tables, such as those calculating subtotals monthly and totals annually.



**Figure 1.12** Multipart table format (Structure of partitions)

#### RH (Report Header)

This is the area for setting fields to be output only to a cover or the first page, such as headings.

This partition is output to a specified position (fixed partition).

The report header partition may be omitted. For examples on using report header partitions, refer to the following:

=> "Figure 1.13 Multipart table format (Report Header: heading on the cover)"

=> "Figure 1.14 Multipart table format (Report Header: heading on the first page)"

#### PH (Page Header)

This is the area for setting fields that are always output at the top of each page, such as a title.

This partition is output to a specified position (fixed partition).

The page header partition may be omitted.

#### CH (Control Header)

This is the area for setting fields that are output before details when data for a specified field changes, such as headings of details.

This partition is output to a floating position according to data (floating partition).

A maximum of 16 control header partitions can be defined. This partition may be omitted.

#### DE (Details)

This is the area for setting detail data.

This partition is output to a floating position according to data (floating partition).  
 A maximum of 100 details partitions can be defined. At least one details partition must be defined.

**CF (Control Footer)**

This is the area for setting fields that are output after details when data for a specified field changes, such as a total and subtotal of the details.

This partition can be output to a specified position (fixed partition) or to a floating position according to data (floating partition).

A maximum of 16 control footer partitions can be defined. This partition may be omitted.

**PF (Page Footer)**

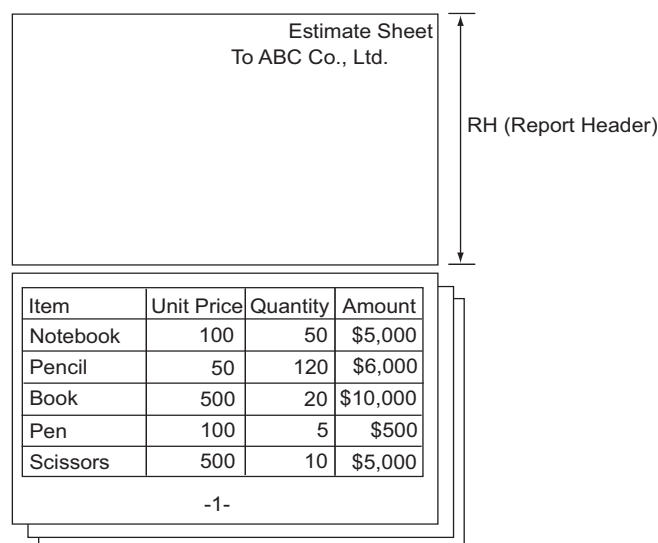
This is the area for setting fields that are always output at the bottom of each page, such as a page number.

This partition is output to a specified position (fixed partition).

The page footer partition may be omitted.

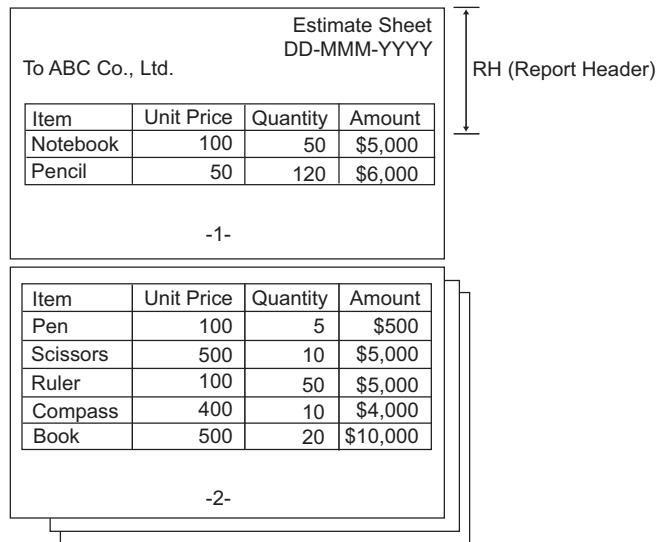
• **Features**

- Positions of fields to be output are designed in relation to one another in a partition. The defined partitions are automatically positioned and output according to data.
- The details partitions to be output can be switched according to field data by specifying conditions on the fields defined in a form.
- Lines/fill that are output to positions relative to each corresponding partition (partition graphic/fill) can be defined. It is useful to use them in floating partitions. It is also possible to define lines/fill that are output to positions defined at a design phase, independent of the positions of partitions to be output (overlay lines/fill).
- The following input data is specified for each detail:
  - When the input data format is "Variable Length", "Fixed Length", or "CSV Format", one record (line) is specified for each detail.
  - When the input data format is "XML Format", one element of a details group item is specified for each detail.
 For details of the details group item, refer to the following:  
 => ["D.6 Conditions under which Elements can be Specified as Details Group Items"](#)
- Multiple timing settings can be specified for calculations and page feed, such as calculating a subtotal when data for a specified field (such as "Month") changes, and calculating a total when data for another field (such as "Year") changes. The timing for calculations and page feed is specified by priority (control level) of each task.
- You can use the RH (Report Header) to output a form that has different headings only on the cover or the first page.



**Figure 1.13** Multipart table format (Report Header: heading on the cover)





**Figure 1.14** Multipart table format (Report Header: heading on the first page)

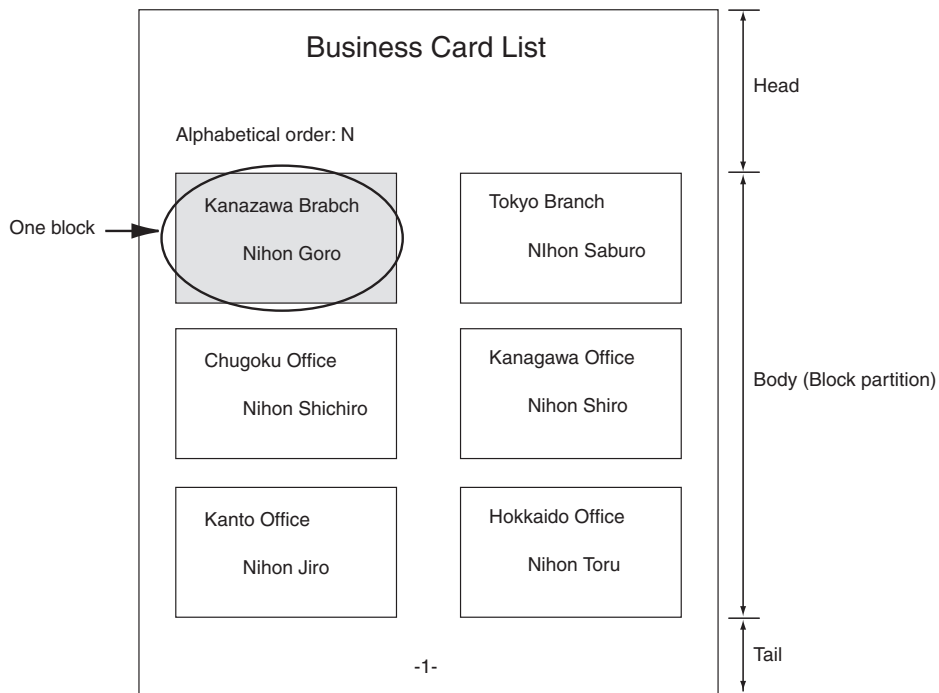


- If the RH (Report Header) is output as the heading on the first page, a partition set as the PH (Page Header) is not output to the first page. Partitions set as the PH (Page Header) are output on the second page and following pages.

### 1.4.5 Block Repeat Format

- What kind of format is this?

This format is made up of the following three partitions. In this format, a block is arranged repeatedly and a heading can be defined for classifying blocks. This is suitable for a list of business cards.



**Figure 1.15** Block repeat format

#### Header

This is the area for outputting data, such as a heading. This partition may be omitted.

Body (Block area)

This is the area for outputting a block repeatedly.

Tail

This is the area for outputting data, such as a page number. This partition may be omitted.

• Features

- The numbers of repeated blocks horizontally and vertically, and the repeat direction can be set.
- A form that is identical to the image defined at a design phase is output.
- The following input data is specified for each block.
  - When the input data format is "Variable Length", "Fixed Length", or "CSV Format", one record (line) is specified for each block.
  - When the input data format is "XML Format", one element of a body area group item is specified for each block.

For details of the body area group item, refer to the following:  
 => ["D.5 Conditions under which Elements can be Specified as Body Group Items"](#)
- A new page can be created when data for a specified field (such as "Department name") changes. Only one field can be specified for this purpose.

### 1.4.6 Sticky Label Format

• What kind of format is this?

This format is made up of the following one partition. This format is suitable for various labels, such as name & address labels, for which a block is arranged repeatedly.

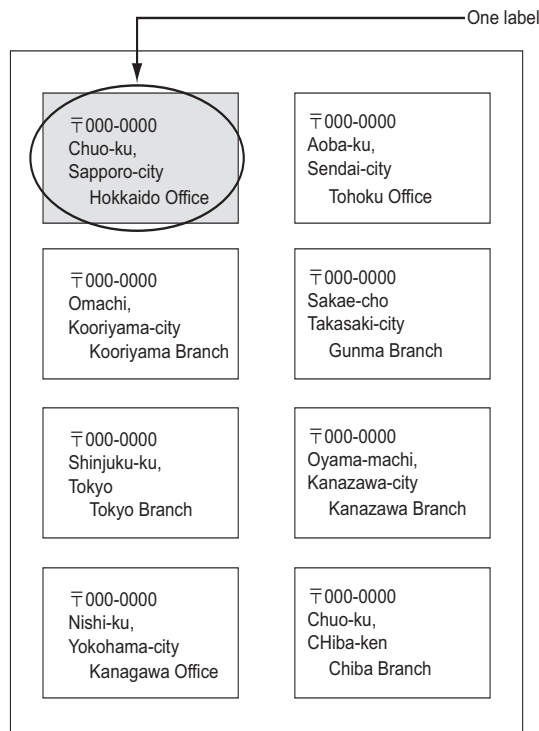


Figure 1.16 Sticky label format

- Features

- A sheet of labels can be designed easily by simply setting the number of labels vertically and horizontally, and margins.
- A form that is identical to the image defined at a design phase is output.
- The following input data is specified for each label.
  - When the input data format is "Variable Length", "Fixed Length", or "CSV Format", one record (line) is specified for each label.
  - When the input data format is "XML Format", one element of a body area group item is specified for each label.

For details of the body area group item, refer to the following:

=> ["D.5 Conditions under which Elements can be Specified as Body Group Items"](#)

## 1.5 Input Data Type and Format

Input data is the data that is overlapped with form style information and output.

List Creator supports the following input data types:

- Data files (text format)
  - Variable length
  - Fixed length
  - CSV format
  - XML format
- Data extracted from a linked database
  - Navigator links function
  - SSIS link function
- XBRL data
- Data linked to ERP



### Note

- For variable length or CSV format, data with the field name label can be used as input data. Since the data can include information of the corresponding field names, it is not necessary to match the order of the fields and that of the input data. For details, refer to the following: => ["8.4 Designing a Data File with the Field Name Label"](#)
- For the input data when performing multi-form output, refer to the following: => ["8.3 Designing Input Data for Multi-form Output"](#)
- For the input data when performing Combination-Form output, refer to the following: => ["10.2 Designing Input Data for Combination-Form"](#)
- Input data can be specified in units of records (lines) or streams on each form output interface. For details, refer to the Online Manual "Application Design".

The following explains the details of each type of input data.

### 1.5.1 Data File

There are the following types of data files. When using data files, you need to determine the description format (input data formats and a separator character used between data fields) for the contents of the data files.

- Variable length
- Fixed length
- CSV format
- XML format

These data files can be extracted or created using the following methods:

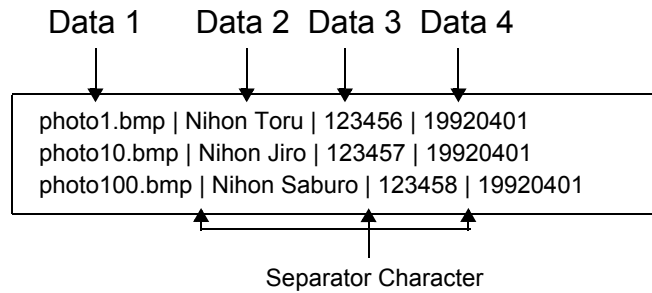
- Extracting from a database
- Extracting from spreadsheet software
- Creating from 4GL

The following describes each data file type.

### 1.5.1.1 Variable Length

The length of each data block is variable. Each data block is separated by a separator character. Specify a character that is not used in any data as the separator character.

The following example shows a data file that uses a vertical bar (|) as the separator character.



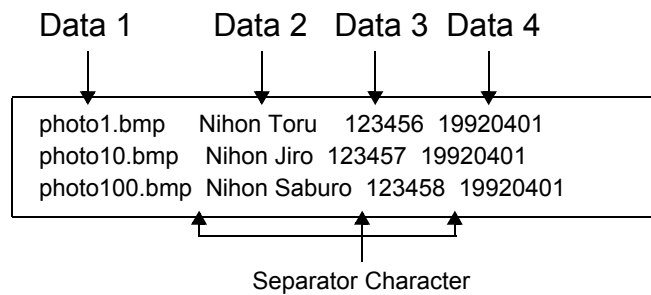
**Note**

- A data file with the field name label can be created by adding a row of field names defined in the form to the top of the input data file. For details on a data file with the field name label, refer to the following:  
=> ["8.4 Designing a Data File with the Field Name Label"](#)

### 1.5.1.2 Fixed Length

The length of each data block is fixed. If the length of actual data is shorter than the specified length, the remaining length is needs to be filled with half-width spaces to create the data. For example, if the data for a 20-byte field has only 15 bytes, fill the missing 5 bytes with half-width spaces.

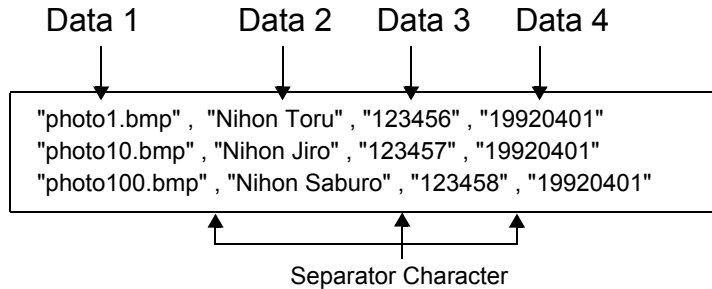
The following example shows a data file that uses spaces as the separator characters.



### 1.5.1.3 CSV Format

CSV format data is used. The length of each data block is variable. Each data block is separated by a separator character. A "comma (,)" or a "tab" is used as the separator character. A character used as the separator character is handled as data when enclosed in double quotation marks (").

The following example shows a data file that uses a "comma (,)" as the separator character.



**Caution**

- If a separator character is included in an input data, each data block needs to be enclosed in double quotation marks.
- If there is a separator character after the "double quotation mark (")", the characters before the double quotation mark (") is considered as one data unit.
- You can specify the operation that should take place when there are two consecutive double quotation marks (") in the data.  
For details, refer to the section about the setup command for List Creator in the Online Manual "Setup & Operation".



**Note**

- A data file with the field name label can be created by adding a row of field names defined in the form to the top of the input data file. For details on a data file with the field name label, refer to the following:  
=> ["8.4 Designing a Data File with the Field Name Label"](#)

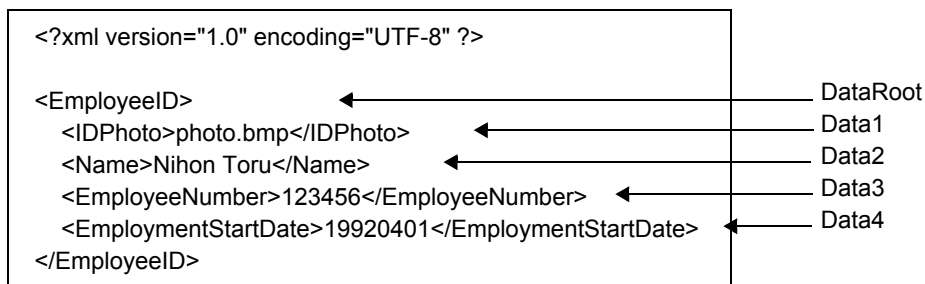
### 1.5.1.4 XML Format

In XML format, each input data block is specified as a child element under a data root.

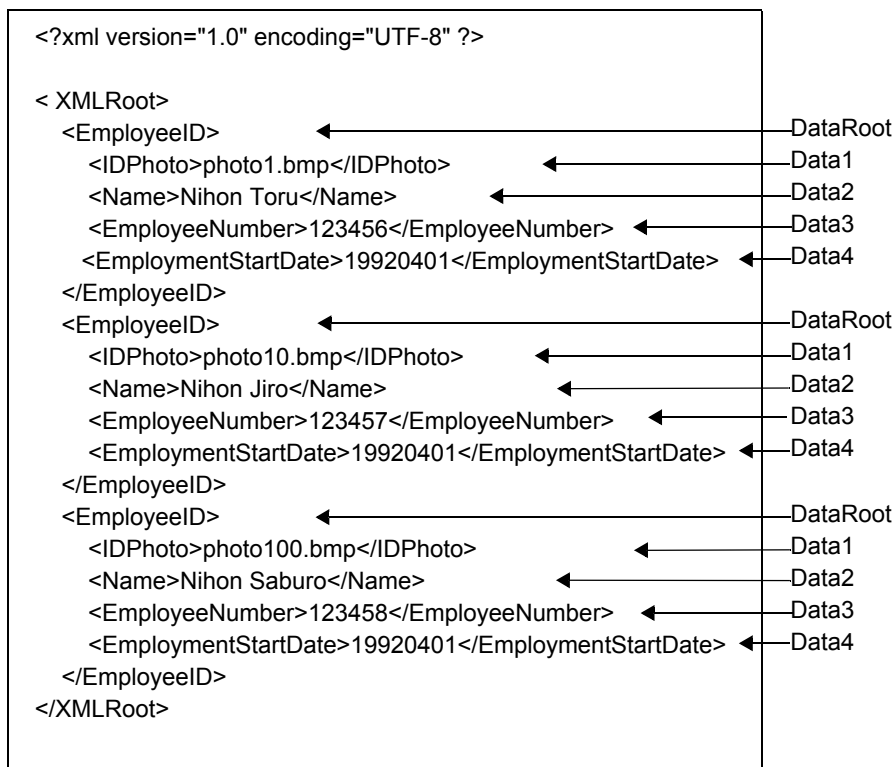
When there is only one set of data in XML format, specify the XML root or the child element under the XML root as the data root.

When there are multiple sets of data in XML format, always specify a child element under the XML root as the data root.

The following shows an XML format file with only one set of input data.



The following shows an XML format file with multiple sets of input data.



## 1.5.2 Linking with a Database

To use data stored in a database as input data, use the following function:

- Navigator Links Function
- SSIS Link Function

The following section explains the overview of Navigator links function and SSIS link function.



**Note**

- When connecting to a database with the Navigator links function, specify the following in the [Data Definitions] window:
  - Input Data Format: "CSV Format"
  - Separator Character: "Comma (,)"
  - Group Item Handling: "Use Configuration Item Name"
- When connecting to a database with the SSIS link function, specify the following settings in the [Data Definitions] window:
  - Database Type: "Data File"
  - Input Data Format: "CSV Format"
  - Separator Character: "Comma (,)"

### 1.5.2.1 Navigator Links Function

Navigator links function can be used to extract the data from the Interstage Navigator server according to the data extraction conditions specified in the form descriptor information in order to output forms.

### 1.5.2.2 SSIS Link Function

SSIS (Microsoft SQL Server 2005 Integration Services) is the platform for building high performance data integration solutions including ETL (extraction, transformation, and load) package for data warehousing.

SSIS link function can be used to extract the data from the database without complex programming in order to output forms in List Creator.

### 1.5.3 Linking with XBRL Data

You can associate the form fields with XBRL data when designing forms, and output the forms by specifying XBRL data.



**Note**

- When linking with XBRL data using XBRL data support function, specify the following settings in the [Data Definitions] window:
  - Input Data Format: "CSV Format"
  - Separator Character: "Comma (,)"
  - Group Item Handling: "Use Configuration Item Name"

### 1.5.4 Linking with ERP

When you output the data created by SAP R/3 or Oracle E-Business Suite to a form by using the List Creator's form descriptor, use the following functions:

- SAP R/3 RDI Integration
- Oracle E-Business Suite integration

The following describes the outline of each function.



**Note**

- When linking a database with the SAP R/3 RDI integration, specify the following settings in the [Data Definitions] window:
  - Database Type: "Data File"
  - Input Data Format: "CSV Format"
  - Separator Character: "Comma (,)"
- When linking a database (Oracle) with the Oracle E-Business Suite integration, specify the following settings in the [Data Definitions] window:
  - Database Type: "Oracle"
  - Input Data Format: "Variable Length" or "CSV Format"

#### 1.5.4.1 SAP R/3 RDI Integration

With the SAP R/3 RDI integration, you can use the RDI data created by a SAP Script and the print program on SAP R/3 RDI. When using the SAP R/3 RDI integration, you need to create definition (RDI) for associating a SAP R/3 form (SAP Script) with List Creator form descriptor information before outputting a form. A form is output in accordance with the created RDI.

#### 1.5.4.2 Oracle E-Business Suite Integration

With the Oracle E-Business Suite integration, you can use data stored in a database (Oracle) as input data. By registering the List Creator form output command in the concurrent program of Oracle E-Business Suite, you can output a form through List Creator from the client window of Oracle E-Business Suite.



## 1.6 Created Files

When you create form descriptor information, the following files are automatically created.

- form\_name.bip  
A file that associates the form style information with input data.
- form\_name.pmd  
The form file.
- form\_name.psf  
A supplementary file that is required for editing the form style information.
- form\_name.ovd  
The overlay file.
- form\_name.dse  
This file is required when XBRL data support function or Navigator links function is used.

These files are stored in a folder that is used as the form storage directory.

When you place the form descriptor information to a production computer, copy all of these files.

If form descriptor information is classified and saved in form storage directories, according to the types and purposes of the forms, you can switch the form storage directories depending on the intended usage during form designing.

For form storage directories, refer to the following:

=> "[Chapter 12 Managing Form Descriptor Information](#)"



- If you place the form descriptor information in a PC running UNIX OS, please convert it to EUC or UNICODE (UTF8) character code using the form code conversion function if needed.

# **Chapter 2**

## *Form Designing Flow*

---

This chapter explains the work flow and tasks in form designing.

## 2.1 Form Designing Flow

To design a form in List Creator, perform the following procedures.

**Table 2.1** Form Designing Flow & References

| Procedure | Tasks  | Reference  |  |  |
|-----------|--|--|--|--|
| 1         | Preparing for Form Designing                                 | (1) Determining the form format  | "4.1 Determining the Form Format"                                    |  |
|           |  | (2) Determine whether to use Unicode   | "4.2 Determine Whether to Use Unicode characters"                    |  |
|           |  | (3) Determining the methods for creating forms                                     | a) Using field entry list  | "4.3 Determining the Methods for Creating Form Layout Data"                                    |
|           |  |  | b) Using layout data from an existing form                           |  |
|           |  |  | c) Using field info files  |  |
|           |  |  | d) Using a schema file   |  |
|           | e) Using a DTD file/XML file                                 |  |  |  |
|           | f) Using XBRL data support function/Navigator links function |  |  |  |
|           | (4) Preparing the resources required for creating a form     | Preparing files and information to be used in the above-mentioned steps (b) to (e) | "4.4 Preparing the Required Files"                                   |  |
| 2         | Starting the [Form Descriptor List]                          | When creating a new form   | Create a new form and specify the form name                          | "12.4 Creating, Opening, Copying, Deleting, and Renaming Form Style Information"               |
|           |  | When updating an existing form   | Open the existing form   |  |
| 3         | Creating Form Style Information                              | Creating form layout data  | Start Form Creation Wizard (only when creating new form)             | "• Display the Form creation Wizard (Select wizard, Specify whether to use Unicode character)" |
|           |  |  | Set the operating environment for defining the form style descriptor | "3.2.2 Customizing Operating Environment"  |
|           |  |  | Define areas, partitions etc.  | Designer Help guide  |
|           |  |  | Define fields, overlays, partition graphics etc.                     | Designer Help guide  |
|           |  | Setting association data   | "Chapter 6 Definition of Association Data"                           |  |
| 4         | Checking the form output image                               | Preview, layout print, test print etc.   | "Chapter 7 Confirming the Output Image"                              |  |
| 5         | Designing input data   |  | "Chapter 8 Designing Input Data"                                     |  |
| 6         | Setting Form Output Methods                                  | Specifying output methods, output attributes etc.                                  | "Chapter 9 Defining the Form Output Methods"                         |  |
| 7         | Saving Form Descriptor Information                           |  | Designer Help guide  |  |

The following sections explain the tasks in form designing.

### 2.1.1 Preparing for Form Designing

The following tasks must be performed before you start form designing:

(1) Determining the form format

Determine the form format that is appropriate for the form you want to create.

Note that when you determine the form format, you must consider what type of form format provides the features appropriate for the format you want to create and decide what format will best serve your purpose. Specify the form format you choose here to create a prototype of form style information by using the Form Creation Wizard in Step 3 of [Table 2.1](#).

(2) Determine whether to use Unicode

Determine whether Unicode characters are to be used in the strings defined in Form definition information.



**Note**

- To design and handle multi-language forms, refer to the descriptions on multi-language form operation in the Online Manual "Setup & Operation".

(3) Determining the methods for creating

Determine the methods for creating the form layout.

Any information such as input data and paper forms (existing resources) that is related to the form you want to create would make it easier to create the form's layout information.

Task (4) must be performed if the method (b), (c), (d) or (e) in [Table 2.1](#) is used.

(4) Preparing the required files

Prepare the files required for the method(s) in (3) of [Table 2.1](#).

Existing resources like paper forms can also be used effectively.

For use of paper forms, refer to the following:

=> ["4.3.1 Using Layout Data from an Existing Form or Preprinted Paper"](#)

### 2.1.2 Starting the [Form Descriptor List]

To design a form, open form descriptor list from List Creator menu.

The Form Descriptor List window displays a list of form descriptor information in the specified folder (form storage directory) to manage and maintain the information. For details, refer to the following:

=> ["3.1.1 \[Form Descriptor List\] Window"](#)

When creating a new form, select [New] from [File] menu and specify the name of the form to be newly created. On specifying the form name, the Form Creation Wizard starts automatically.

When updating a form, select the form descriptor information to be updated in the [Form Descriptor List] window and select [Open] from [File] menu. [Form Style Descriptor] window is opened and you can edit the form descriptor information.

### 2.1.3 Creating Form Style Information

Create form style information. The definition of form style information is referred to as "form style descriptor".

In the form style descriptor, [Form Layout data] and the [Association Data] of form layout data and input data are defined.

- **Creating form layout data**

**Starting the Form Creation Wizard**

When creating a new form, the Form Creation Wizard automatically starts.

By using the Form Creation Wizard, you can create a prototype of form by setting the sequence of information including the form format.

When the Form Creation wizard is complete, the [Form Style Descriptor] window is displayed.

This is the window to define form style information. For details, refer to the following:

=> ["3.1.2 \[Form Style Descriptor\] Window"](#)

**Defining and editing form layout data**

Based on the form layout data of the prototype created by the Form Creation Wizard, add or modify the required definitions and complete the form layout.

Fields and overlays (figures) etc are defined as Form layout data.

- **Defining association data**

Define the association between fields defined by form layout data and input data.

For example, you match the output order of the fields in the form layout data with the order of the input data, or you define an operation to automatically calculate a value that is output in the form.

The association data is also defined in [Form Style Descriptor] window.

## 2.1.4 Checking Output Image of Form

Check the output image for the created form style information by using the preview function and so on. The actual output image can be displayed in print image, PDF file format or Excel file format according to the output method. In this case, you can automatically create a format of the input data and check the output image. Or, if the input data used for form outputting has been determined before you create form descriptor information, you can check the output image with that input data.

In addition to the preview function, there are functions such as layout print or test print which also can be used for checking the output image of the form layout data.

## 2.1.5 Designing (Preparing) Input Data

Design the input data that is overlapped over the created form style information.

If the input data used for form outputting has been determined before you create form descriptor information, you should check that the input data format is appropriate for the configuration of the form style information.

## 2.1.6 Setting Form Output Methods

Define information regarding the output method for a form, operation during output, and operation after output.

## 2.1.7 Saving Form Descriptor Information

After completing form designing, save the created form descriptor information.

The saved form descriptor information can be managed and maintained from the [Form Descriptor List] window.

## ***Chapter 3***

# *Windows and Operating Environment used for Form Designing*

---

This chapter explains the windows used for form designing and the operating environment at the time of designing.

## 3.1 Windows Used for Form Designing

This section explains the following windows used for form designing:

- [Form Descriptor List] Window
- [Form Style Descriptor] Window

Various item names and the main functions of each window are explained here. For details of each window and its operation, refer to the Designer Help guide.

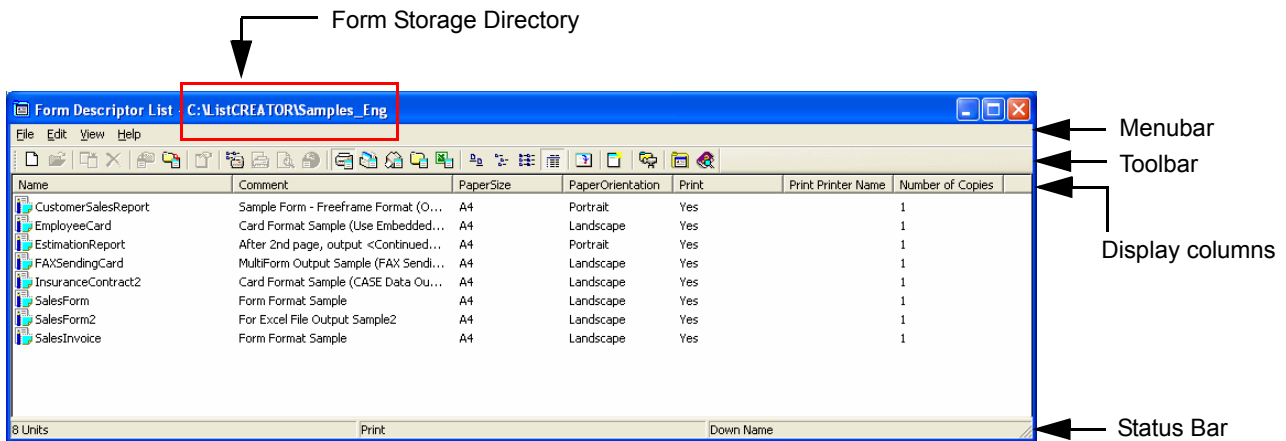
### 3.1.1 [Form Descriptor List] Window

This is the first window that is displayed when form style information is newly created or updated (maintained). Form output methods (operations performed when a form is output) can also be set or changed.

The created form descriptor information is centrally managed in the [Form Descriptor List] window. For details on central management of form descriptor information, refer to the following:

=> "[12.1 About Form Descriptor Information Management](#)"

To display the [Form Descriptor List] window, select [Start] - [Programs] - [List Creator] - [Form Descriptor List].



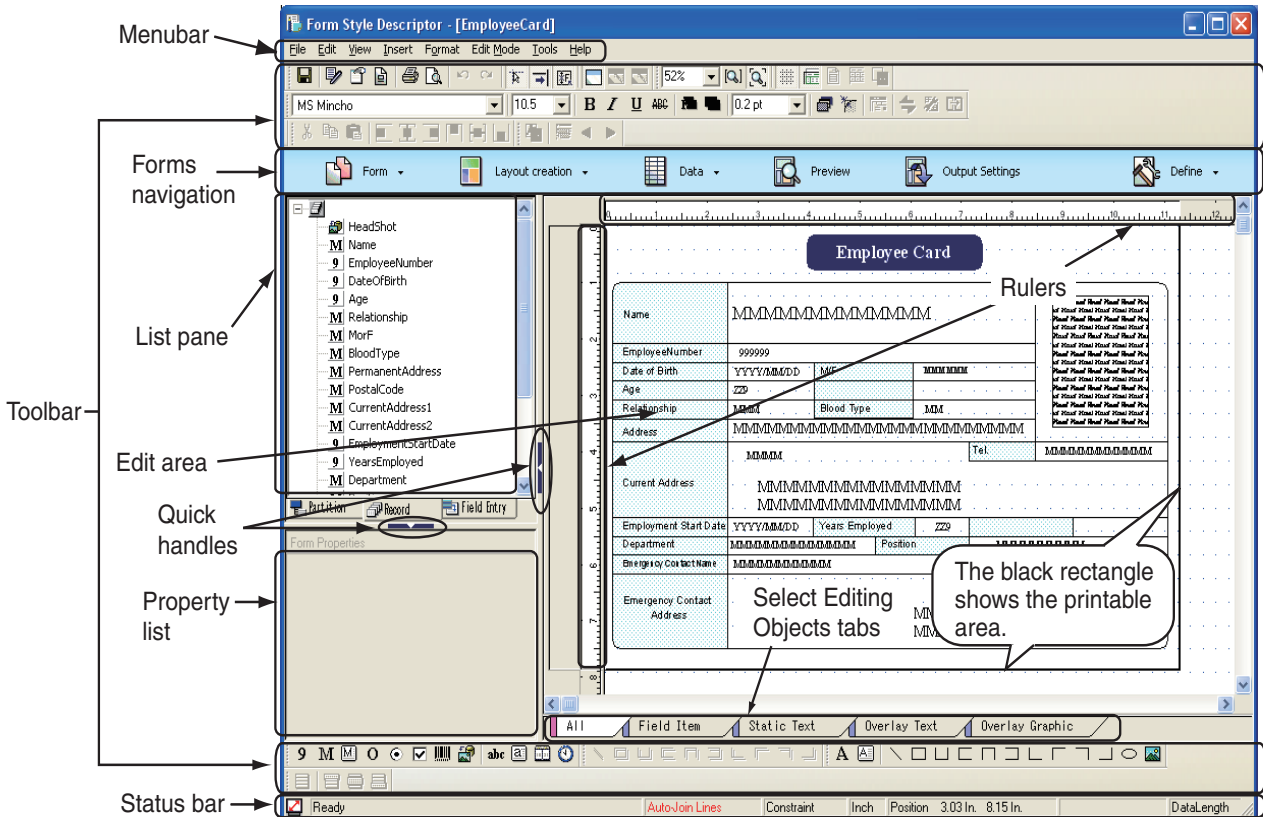
**Note**

- Display the Combination-Form Descriptor List from the [Form Descriptor List] window. For details on the Combination-Form Descriptor List, refer to the following:

=> "[10.1 Designing the Combination-Form](#)"

### 3.1.2 [Form Style Descriptor] Window

This is the window that is displayed when creating the new form style information (and the wizard for doing it) is complete. This window can also be viewed/updated from the [Form Descriptor List] window by selecting and opening the form style information that has been created.



\*\* "List pane" and "Property list" are collectively called "Side view".

#### 3.1.2.1 Menubar

The menubar displays all the commands for form style descriptor. Select and execute a command from pull-down menus or submenus in each menu, using a mouse or keyboard.

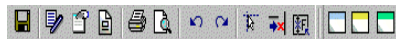
#### 3.1.2.2 Toolbar

There are the following types of toolbars:

- Standard toolbar



- View toolbar



- Format toolbar



- Editing toolbar



- Tools toolbar



- Fields toolbar





- Partition Graphics toolbar



- Overlay toolbar



- Frame toolbar



**Note**

- Immediately after installing, the following toolbars are not displayed:

- Editing toolbar
- Tools toolbar

To display, display Toolbar window from the [View] menu, and select the checkbox of the toolbar to be displayed.

For details of how to show/hide the toolbars, the toolbar types, and the icons of each toolbar, refer to the Designer Help guide.

### 3.1.2.3 Forms Navigation

Forms Navigation contains the main menus used for steps to create forms.

Forms can be created easily by selecting icons sequentially from left to right to perform each task.

For the procedure of creating the forms using forms navigation, refer to the Online Manual "Form Designing Introduction Guide".

For details of the menu, refer to the Designer Help guide.

To switch between showing and hiding the forms navigation, select [View] - [Forms Navigation] on the [Form Style Descriptor] window.

### 3.1.2.4 List Pane

The list pane includes the following three lists:

- Partition list
- Record list
- Field entry list

To switch to another pane, click either tab below the list pane.



Click either tab to switch to another pane.

"List pane" and "property list" are collectively called "Side view".

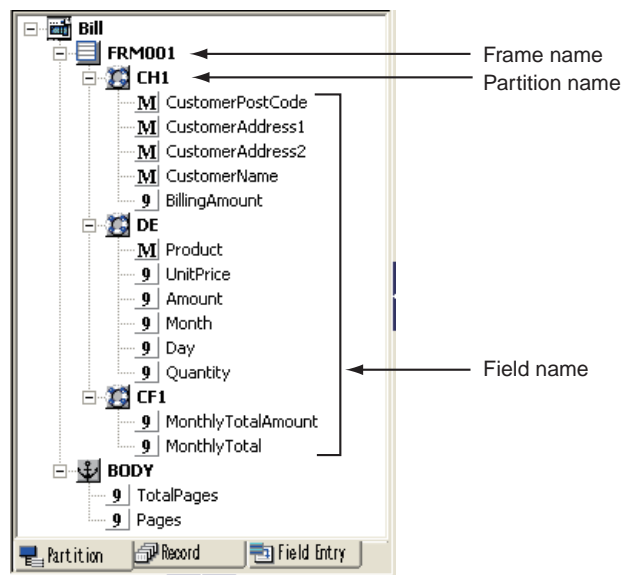
- **Partition list**

The list of frames, partitions, and fields defined in the [Form Style Information] is displayed.

The fields that can be viewed are shown below.

- Static field
- Block static field
- Numeric field
- Text field
- Block text field
- OCR-B field
- Radio button
- Check box
- Barcode field
- Image field
- Date field
- Time field

The window shown below is a free frame format example. For operations in the partition list, refer to the Designer Help guide.



**Note**

- You can choose to show/hide the following fields from the pop-up menu of the partition list:
  - Static field
  - Block static field
  - Date field
  - Time field

- **Record list**

The list of fields defined in [Form Style Information] is displayed.

The fields that can be viewed are shown below.

- Numeric field
- Text field
- Block text field
- OCR-B field
- Radio button
- Check box
- Barcode field (not including barcode (U.S. Postal FIM) fields)
- Image field
- Post code field
- Hidden field (only for record)
- Group item

The window shown below is a record list example.



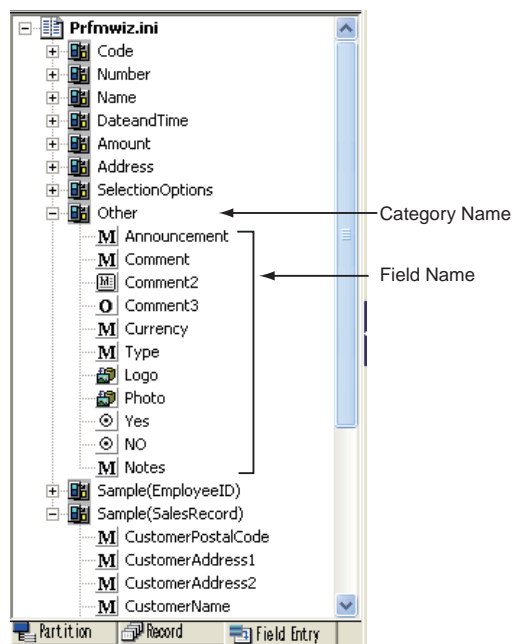
**Note**

- When the input data format is "XML Format", the data list is displayed.

- **Field entry list**

List Creator Designer provides a file (Prfmwiz.ini) that contains some frequently used files. The field entry list shows the list of fields registered in this file. The fields in the list can be defined by dragging and dropping them into the edit area on the right.

The Prfmwiz.ini file can be edited and used in accordance with the form style information you hope to create. For methods of editing the Prfmwiz.ini file, refer to the Designer Help guide.



### 3.1.2.5 Property List

You can switch between showing and hiding the property list by clicking on the quick handle.

To view the properties of each field on the property list, show the property list and perform any of the following operations:

- Double-clicking the field on the Partition list or Record list.
- Selecting the field in the edit area

| Form Properties(Static Field) |            |
|-------------------------------|------------|
| Properties                    |            |
| Field Name                    | Static0024 |
| Position(Ver.)                | 1.17"      |
| Position(Hoz.)                | 6.98"      |
| Size(Field Length)            | 1.70"      |

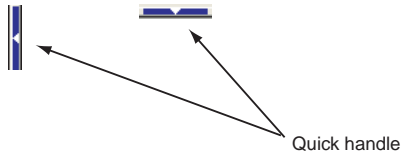
The field's properties on the property list can be changed.

The attributes displayed on the property vary depending on the field type. For details, refer to the Designer Help guide.

"List pane" and "property list" are collectively called "Side view".

### 3.1.2.6 Quick Handle

You can toggle between showing and hiding the side view or the property list.



Clicking either of the quick handles switches showing and hiding of the side view or the property list.

### 3.1.2.7 Edit Area

"Edit area" is the place where areas such as partitions, and objects such as fields and overlays are aligned. This is the main area to design form layout information.

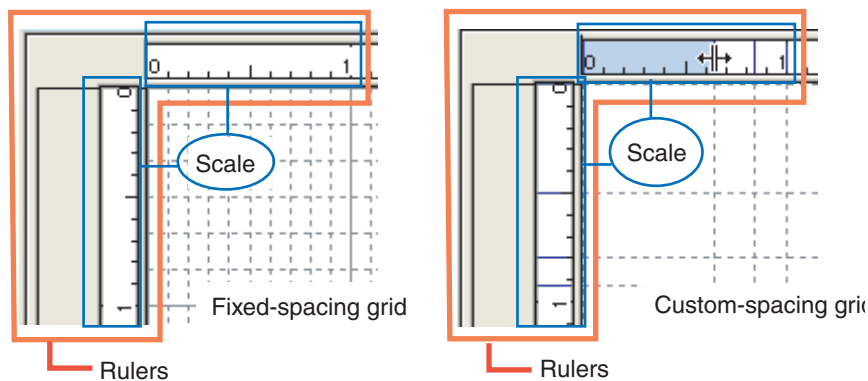
"Rulers", "grid" and "Select Editing Objects" tabs are displayed in the edit area.

#### • Rulers

"Rulers" are located on the top and left of the edit area, and show the partition name and the scale which can be used as a guide for editing.

To show or hide rulers, check or do not check the [Rulers] of the [View] menu.

Rulers may vary according to the grid types. For a "custom-spacing grid", additional gridline handles are shown on the ruler segment, and you can select a gridline handle or grid spacing.



For details about grid types, refer to the following:

=> "3.3 Grid Function"

Additionally, the rulers (vertical for multipart table format, vertical and horizontal for free frame format) have the area to show the partition name (partition label).

- Grid

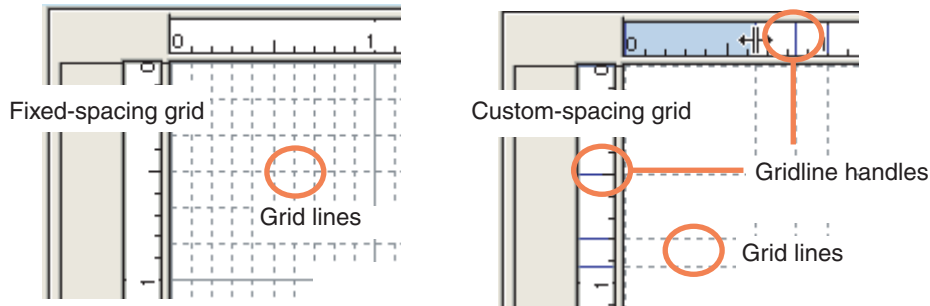
Grid is a pattern of lines displayed in the area where the objects are aligned. Although the grid is not actually output to the form, it is displayed as a guide for adjusting position and size of the objects.

To show or hide grid, check or do not check the [Grid] of the [View] menu.

The line style and color of gridlines and the grid spacing can be set in the [Grid Settings] window.

For details, refer to the Designer Help guide.

Grid may vary according to the grid types. For "custom-spacing grid", any gridlines and grid spacing can be set.

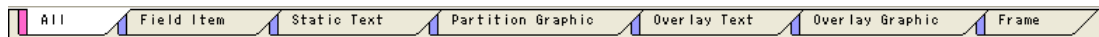


For details about grid types, refer to the following:

=> "3.3 Grid Function"

- Select Editing Objects Tabs

Select editing objects tabs are the tabs to select objects to be edited.



To select more than one tabs, hold down the [Ctrl] key and select them.

Only the objects selected on the select editing objects tabs can be edited in the edit area.

There are the following select editing objects tabs.

- All  
All objects displayed in the edit area can be edited, except for the post code fields, hidden fields, and group items.
- Field  
The following fields can be edited:
  - Numeric field
  - Text field
  - Block text field
  - OCR-B field
  - Radio button
  - Check box
  - Barcode field
  - Barcode field (U.S. Postal FIM)
  - Image field
- Literal  
The following fields can be edited:
  - Static field
  - Block static field
  - Date field
  - Time field
- Partition graphic  
The following partition graphics can be edited:

- Partition graphic (Line)
- Partition graphic (Rectangle)
- Partition graphic (Fixed pattern)

Note that this tab is only displayed for free frame format or multipart table format.

- Overlay text

The following overlay texts can be edited:

- Overlay text
- Block Overlay text

- Overlay Graphic

The following overlay graphics can be edited:

- Overlay line
- Overlay rectangle
- Overlay fixed pattern
- Overlay ellipse
- Overlay image

- Frame

The following frames and frame partitions can be edited:

- Frame
- Frame partition

Note that this tab is only displayed for free frame format.

### 3.1.2.8 Status Bar

The status bar displays information regarding the selected field or the field pointed at by the mouse pointer.



For details of the status bar, refer to the Designer Help guide.

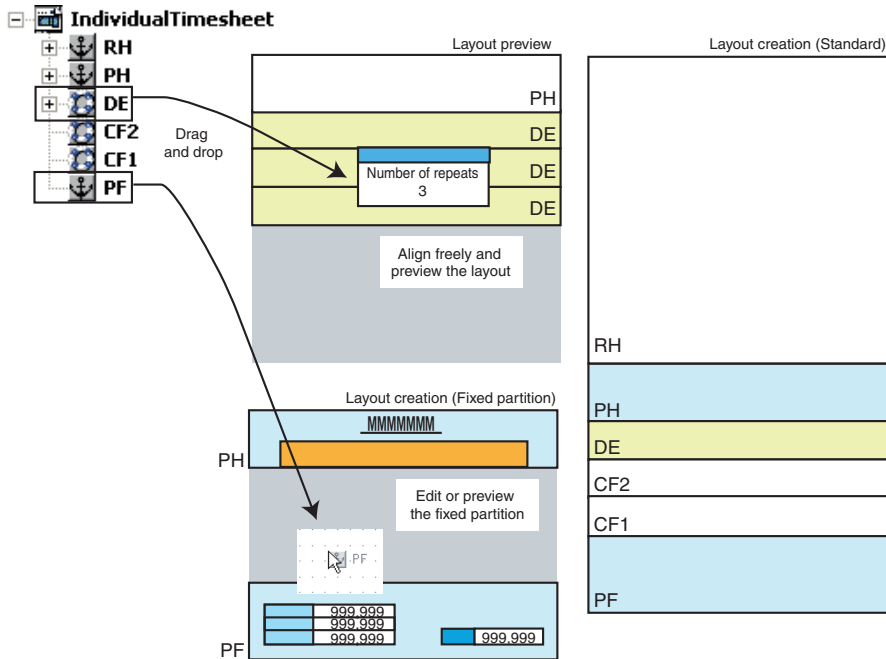
## 3.2 Operating Environment for Form Designing

This section explains the operating environment of the form style descriptor window for form designing.

### 3.2.1 Layout Creation and Layout Preview Windows

[Form Style Descriptor] window is a window to create form layout data. The display mode of this window can be switched for multipart table format or free frame format.

For example, in case of designing a multipart table format, you can design the form while previewing the layout of the defined objects and partitions, by switching the display mode of the window as shown in the figure below.



The following sections explain the display mode of the [Form Style Descriptor] window when creating the layout.

- **Layout Creation (Standard)**

Regardless of the form format, this display mode can be used for designing all forms.

In this mode, you can review and edit all the components (frames, fixed partitions, floating partitions and objects) in the form layout data.

Also, you can display only the partitions that need to be edited, while other partitions are hidden.

For multipart table format or free frame format, this view displays all the defined partitions in the order in the partition list. For fixed partition of free frame format, since it is displayed at a fixed position in the frame, you can edit it in the layout creation window.



**Note**

- For the fixed partition in multipart table format, the start position of the defined partition may differ from the printed position (displayed in a position different from the actual print position). Therefore, when editing the fixed partition in multipart table format while previewing the layout, edit the form in "Layout creation (Fixed partition)" mode.
- The [Form Style Descriptor] windows in "Layout creation (Standard)" mode and "Layout creation (Fixed partition)" mode may be referred to as "Layout creation window".

- **Layout Creation (Fixed Partition)**

This display mode can be used only for multipart table.

This view displays only the fixed partitions in the positions where they are actually printed, enabling you to check the layout.

Each object defined in the fixed partition can also be edited.

If the positions (lines) of the fixed partitions overlap (for example, a RH partition overlaps a PH partition etc.), you cannot preview or edit both partitions simultaneously, however, you can easily switch between the two partitions by selecting the partition to be displayed from the partition list and dragging it to the edit area.



**Note**

- The [Form Style Descriptor] windows in "Layout creation (Standard)" mode and "Layout creation (Fixed partition)" mode may be referred to as "Layout creation window".

- **Layout Preview**

This display mode can be used for multipart table format and free frame format.

In this mode, you can align all partitions in the edit area freely, and preview the form layout.

Also, you can check the layout while you specify Dittoed output.

To place partitions on the edit area, just select the target partition from the partition list and drag it to the edit area.

When a floating partition is displayed, a single floating partition can be displayed multiple times by specifying the number of repeats for the partition.

Partition can be displayed in the edit area using the Layout Editor.



**Caution**

- In free frame format, a partition cannot be displayed by dragging it in the edit area. For the methods of previewing the layout of the free frame format, refer to the Designer Help guide.
- The [Form Style Descriptor] window in Layout preview mode is also called "Layout preview window".

### 3.2.2 Customizing Operating Environment

When designing forms, it may be necessary to customize the operating environment in the form style descriptor such as unit settings, selection of grid types, and so on.

The following sections explain the operating environment that should be confirmed before form designing.

- **Unit Settings**

Inch, mm, or dot can be selected as the basic unit for operations during form designing. Default value is "Inch".



**Note**

- In List Creator, a value in dots is specified as the number of dots at 720 dpi (Dot Per Inch) conversion (1 inch = 25.4 mm = 720 dots).
- If "Custom-spacing grid" is specified for the grid type, specifying "Dot" is recommended. For details about custom-spacing grid, refer to the following:  
=> "[3.3.2 Custom-Spacing Grid](#)"

- **Selection of Grid Types**

Select "Fixed-spacing grid" or "Custom-spacing grid" as the grid type used during form design.

For the grid function, refer to the following:

=> "[3.3 Grid Function](#)"



**Note**

- If the form is output to an Excel file, specifying "Custom-spacing grid" is recommended. With custom-spacing grid, you can design a form by corresponding the grid settings (position of fields or graphics etc.) to the applicable cells in an output Excel file.



• **Default Value Setting**

The following default value can be set before designing forms.

- Default value of each properties of fields, overlay (graphics/text), partition graphics
- Default value of field color palette/ overlay color palette
- Default value of Form properties
- Default value of unit settings
- Default value of grid (only when the fixed-spacing grid is specified)

Some default value settings are enabled immediately after the change, and others are enabled after restarting. For details, refer to the Designer Help guide.

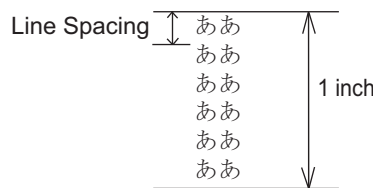
• **Line spacing**

In order to obtain an output similar to the existing form such as that in the form of paper, confirm how many lines are output per inch (25.4 mm) in the existing form.

If six lines are output per inch, it is called "1/6 inch line feed".

Line spacing is also expressed by the unit of "LPI (Line Per Inch)", so the above example is also called "6 LPI".

[For 1/6 inch line feed]



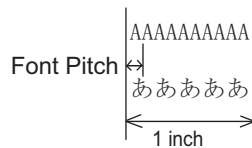
Specify the above line spacing value in the [Line spacing] of the [Form Properties] on the [Form Style Descriptor] window. For details regarding the Line spacing, refer to the "[Form Descriptor] tab" on the [Form Properties] window in the Designer Help guide.

• **Character Spacing**

In order to obtain an output similar to the existing form such as that in the form of paper, confirm how many half-width characters are output per inch (25.4 mm) in the existing form.

The number of characters output per inch are generally expressed by the unit of "CPI (Character Per Inch)".

[For 10 CPI]



In the [Grid Settings] window of the [Form Style Descriptor] window, specify the line spacing value in [Grid Spacing : Vertical] and character spacing value in [Grid Spacing : Horizontal] as above mentioned, and select [Snap to Grid]. For details, refer to the "[Grid Settings] Window" in the Designer Help guide.

Then, after checking [Non-Standard Pitch] on the [Font] tab of the object properties, specify the value of character spacing in [Font Pitch] and specify half of that value in [Japanese Font Pitch]. For details, refer to the "[Font] Tab" in the Designer Help guide.

These settings are not required if you specify character spacing value in [Horizontal] on the [Grid Settings] window, and select [Fit Character spacing] on the [Grid Options] window.

If the output environment (operating system etc.) is changed, the size of one character may be subtly different even with the same font name. However, if you specify the character spacing, the character will be output to a fixed position in any environment.

### 3.3 Grid Function

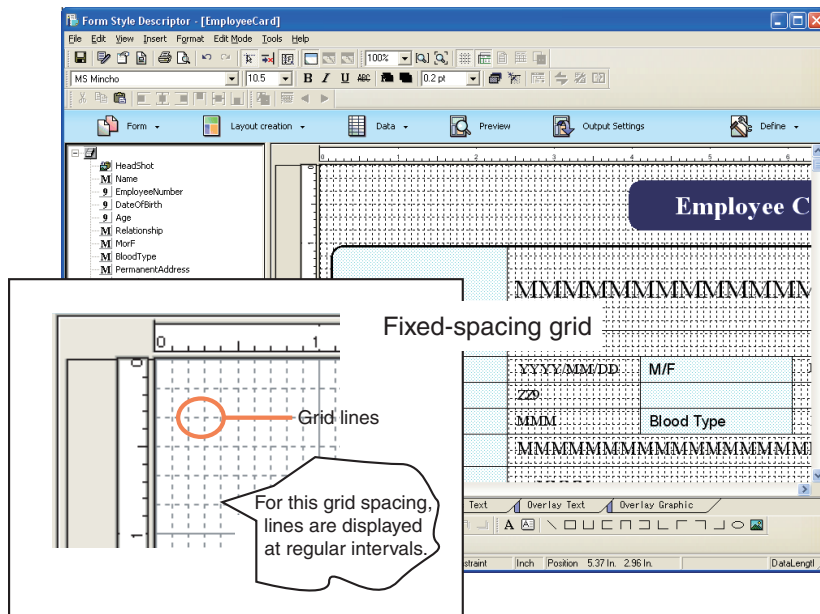
Grid is a pattern of lines displayed on the window for designing form layout. In List Creator, it can be used as a guide to align fields and graphics.

List Creator supports two types of grid, "Fixed-spacing grid" and "Custom-spacing grid", and either grid can be selected depending on the form to be created.

The following explains the fixed-spacing grid, custom-spacing grid and the snap to grid function.

#### 3.3.1 Fixed-Spacing Grid (Default)

Fixed-spacing grid is a pattern of lines displayed with fixed spacing both vertically and horizontally, like a graph paper. To align objects in the form with fixed character spacing (character size) and fixed line spacing (line height), designing can be done smoothly using this grid.



In List Creator, the grid default value is "fixed-spacing".



## Part 2 Form Designing

---

List Creator allows designing and outputting of multi-language forms, including Japanese and other languages. For details on working with multi-language forms, refer to the section about the multi-language form operation in the Online Manual "Setup & Operation".

This part explains the details when designing forms. When setting fields in the setting window, refer to the explanations in the Designer Help guide. This is displayed by clicking the [Help Topics] button in the window.



**Note**

- ["Part 2 Form Designing"](#) explains the operation methods using the menu.
  - Depending on the environment, output methods or resources used during the form output, you may need to pay attention when designing forms. For notes when designing forms, refer to the following.
    - => ["Appendix B Cautions for Designing Forms"](#)
    - => ["Appendix C Cautions regarding Input Data"](#)
- For more information regarding operations or functions of form designing, and constraints etc., also refer to the Designer Help guide.

# Chapter 4

## *Preparing for Form Designing*

---

The following tasks must be performed before you start form designing:

- Determining the Form Format
- Determine Whether to Use Unicode characters
- Determining the Methods for Creating Form Layout Data
- Preparing the Required Files

If you do not use existing resources, there is no need to prepare any files.

This chapter explains the procedures for each task.



The following describes each form format. When you use "XML Format" as an input data format, also refer to the following.

=> "D.1 Terms regarding XML Format Files"

=> "D.3 Specifying the Form Format responding to the XML Data Configuration"



**Note**

- The format of a form already created can be changed to another format. Formats can be changed in the following way:

- Simple Table -> Free frame format
- Single page -> Free frame format
- Multipart table -> Free frame format
- Free frame format -> Simple Table
- Free frame format -> Single page
- Free frame format -> Multipart table

For form change method, refer to the Designer Help guide

## 4.1.2 Information Required for Designing Forms in each Format

Before creating form style information in your selected form format, you are recommended to collect information, such as settings required by the wizard, so that you can create the form style information efficiently.



**Note**

- The following paper sizes are supported in List Creator:

- A3, A4, A5, A6
- B4, B5
- Postcard
- Letter, Legal
- Custom size (Any size between 0.01 and 22.5 inches can be set)

Note that some paper sizes may not be available depending on a combination of the operating system on the List Creator print server and selected output method.

The following sections describe information that should be collected for each form format.

### 4.1.2.1 Information Required for Free Frame Format

The following information is required:

- Paper size and orientation
- Types of partitions to be used
- Whether frames are used or not
- Types of frame partitions to be used in a frame
- Fields and overlays (graphics) to be used for outputting each form element
- Stacking of details
- Positions of fields and overlays to be used

On a form in free frame format, you can combine frames as you like for output.

You need to determine definition of each frame, arrangement of frame partitions, and types of frame partitions.

### 4.1.2.2 Information Required for Single Page Format

The following information is required:

- Paper size and orientation
- Fields and overlays (graphics) to be used for outputting each form element
- Positions of fields and overlays to be used

#### 4.1.2.3 Information Required for Simple Table Format

The following information is required:

- Paper size and orientation
- Whether the head or tail partition is used or not
- Number of details to be output to one page
- Whether the number of columns or lines is specified or not
- Fields and overlays (graphics) to be used for outputting each form element
- Positions of fields and overlays to be used

#### 4.1.2.4 Information Required for Multipart Table Format

- Paper size and orientation
- Types of partitions to be used
- Fields and overlays (graphics) to be used for outputting each form element
- Positions of fields and overlays to be used

On a form in multipart table format, you can define and combine partitions as you like for output. You need to determine arrangement and types of partitions.

#### 4.1.2.5 Information Required for Block Repeat

- Paper size and orientation
- Whether the head or tail partition is used or not
- Block size
- Number of repeated blocks and repeat direction
- Fields and overlays (graphics) to be used for outputting each form element
- Positions of fields and overlays to be used

#### 4.1.2.6 Information Required for Sticky Label Format

- Paper size and orientation
- Label size
- Number of repeated labels and repeat direction
- Fields and overlays (graphics) to be used for outputting each form element
- Positions of fields and overlays to be used



## 4.2 Determine Whether to Use Unicode characters

Determine whether Unicode characters are to be used in the strings defined in Form definition information.

This document explains form design when Unicode characters are not used.

For the method of creating form descriptor information when using Unicode characters, refer to the Designer Help guide. For the cautions, refer to the section about the multi-language form operation in the Online Manual "Setup & Operation".

## 4.3 Determining the Methods for Creating Form Layout Data

You need to determine the method for creating form layout data (form layout).

This section explains the methods that facilitate the creation of form layout data by utilizing existing resources (such as input data and existing forms).

If you do not have or use existing resources for the data creation, create your form layout data without using the methods explained here. In that case, proceed to ["5.1.1 Using Field Entry List"](#).

You can use the following methods that utilize existing resources:

- Using layout data from an existing form or preprinted paper
- Using Input Data Field Information
- Using a DTD file/XML file
- Using XBRL data support function/Navigator links function

The following describes each creation method.

For the procedures for preparing files required for creating form style information in each method, refer to the following:

=> ["4.4 Preparing the Required Files"](#)

### 4.3.1 Using Layout Data from an Existing Form or Preprinted Paper

This method can be used when you have an existing form.

A form, such as that in the form of paper, is scanned to an image file. Information on the form layout is imported from the image file to create form layout data.

The form image file is imported through the wizard. The imported image file is displayed as a template for the form layout data you are creating. Fields and overlays are defined based on the template. Moreover, when overlay conversion function is used, the lines, net and characters from the image below are changed into overlay and an item is automatically created as per the overlay (Automatic creation of item). When an image file has color, template can be displayed in color or can also be changed into overlay in color.


| Existing resource                       | Files to be prepared before designing a form | Timing of file import                             |
|---|--|---|
| Preprinted paper                        | Image file (.bmp)                            | - During the wizard process                       |
| Paper form                              |  | - When wizard ends                                |
| Electronic data (Word or PDF etc.) (*1) | Overlay (*2)                                 | When wizard ends (*3)                             |
|   | Form (*2)                                    | Fetch is not required (*4)                        |
|   | Image file (.bmp)                            | - During the wizard process<br>- When wizard ends |

\*1: Can be used only for windows. Use overlay utility with using electronic data.

\*2: Create overlay and form by using overlay utility.

\*3: When overlay is fetched after ending the wizard, the lines, net and characters of electronic data are converted in overlay and are displayed on form style display screen.

\*4: The file is already converted to a form that can be copied to the form storage directory and used.



**Caution**

- The following fields cannot be generated automatically. However, image files can be converted to overlays.
  - Free frame format where partition or frame is defined
  - Simple table format
  - Multipart table format
  - Block repeat format
  - Sticky label format
- When partition or frame is defined in a free frame format, convert the image file to overlay without automatically creating fields, and then define fields according to the converted overlay.

### 4.3.2 Using Input Data Field Information



This method uses input data field information, which is defined on a form.

This method can be used when input data used for outputting a form has been determined before creating form style information.

Field info files, in which input data field information is defined, are created in advance, and imported during the wizard process. The imported field information is used for defining fields in the [Form Style Descriptor] window by dragging and dropping it into the edit area.

| Existing resource                                     | Files to be prepared before designing a form | Timing of file import                        |
|---|--|--|
| Data files in variable length/fixed length/CSV format | Field info file                              | During wizard operation and form design (*1) |
| Database containing required data                     | Schema file (schema.ini)                     | During the wizard process                    |

\*1: The field info file can also be imported from the [Form Style Descriptor] window during form designing. For details, refer to the Designer Help guide.

### 4.3.3 Using DTD/XML File

This method uses an XML file as input data.

The data structure and element names in a DTD file and the data in an XML file are used.

The DTD file or XML file is imported directly during the wizard process. Information generated from the imported information is used for defining fields in the [Form Style Descriptor] window by dragging and dropping it into the edit area.

| Existing resource | Files to be prepared before designing a form | Timing of file import     |
|-------------------|--|---------------------------|
| DTD file          | -  | During the wizard process |
| XML file          | -  |                           |

### 4.3.4 Using XBRL Data Support Function/Navigator Links Function

It is a method of creating form layout information by using XBRL data support function or Navigator links function. For the creation procedure that uses XBRL data support function or Navigator links function, refer to the Japanese version Online Manual.

## 4.4 Preparing the Required Files

This section explains the procedure for preparing files required for the method you have selected in ["4.3 Determining the Methods for Creating Form Layout Data"](#).

### 4.4.1 Creating an Image File

When you use layout data from a paper form or preprinted paper, a bitmap image file is required. The bitmap image file is created by scanning a paper form or preprinted paper and saving it in bitmap format. This is called as "Overlay conversion".



**Note**

- For the points to note when creating items automatically, refer to the "Cautions when Automatically Generating Fields" in the Designer Help guide.

- **Resolutions when converting to overlays**

When you convert a bitmap image file to overlays, use the following resolution when scanning:

**Monochrome image file:**

- To recognize texts: 400dpi recommended
- To recognize lines: 200 - 600dpi (400dpi recommended)
- To automatically generate fields: 400dpi

**Color image file:**

- To recognize texts: 200dpi recommended
- To recognize lines: 200 - 600dpi (200dpi recommended)
- To automatically generate fields: 200dpi recommended

- **Cautions when converting color image data to overlays**

When converting color image data to overlays, note the following points:

- When performing color recognition of image data, the data is recognized in 24 bit color. The color of overlays after recognition is as follows:

| Condition   | Color of overlays after recognition  |
|---|--|
| [Create a KOL5 Format Overlay] is selected in the Form Creation Wizard:     | Basic 16 colors only   |
| [Create a KOL5 Format Overlay] is not selected in the Form Creation Wizard: | Basic 16 colors plus Custom 16 colors (of the given RGB values), 32 colors in total. |

- When recognizing colors of compressed image, some colors may be omitted..
- Compensation for image skew is disabled if specified.

If the color image is reduced to monochrome and displayed with compensation for image skew, overlays that are recognized as colored do not match with the displayed image because compensation for image skew is disabled for the overlays.

### 4.4.2 Creating Field Info Files

This section explains the procedure for creating field info files.

When you use input data in variable length, fixed length, or CSV format, create field info files by performing the procedure described here.

- **Definition of field info file**

A field info file is a file in which you define information of fields to be defined in form style information in advance. You can define the following field information in a field info file:

- Numeric field
- Text field
- Block text field
- OCR-B field
- Radio button
- Check box
- Barcode field
- Image field

Fields defined in the field info file can be viewed in the record list in the [Form Style Descriptor] window when creating new form style information. You can define the fields by dragging and dropping them into the edit area.



**Note**

- Changes made to field information in the [Form Style Descriptor] window are not saved to the field info file.
- The information of fields defined in the field info file is saved in the form even if it is not defined in the edit area (such field is saved as "hidden field (record)").

- **Method for creating field info files**

One field info file is created for each piece of input data.  
Use a text editor such as Notepad to create field info files.

- **File Name**

Any file name (any extension) can be used. The file can also be stored in any location.

- **Selection of Field Type**

You need to determine which field type is used to define each input data record. Also, detail information for each record needs to be determined.

**Numeric field**

This field is used for outputting numeric data.

In addition to field type (Numeric), data length (1 to 18) and decimal places (0 to 18 but less than the data length) can also be entered in a field info file.

**Text field**

This field is used for outputting string data in one line.

In addition to field type (MText), data length (1 to 2997) can also be entered in a field info file.

**Block text field**

This field is used for outputting string data in multiple lines.

In addition to field type (RectMText), data length (1 to 9999) can also be entered in a field info file.

**OCR-B field**

This field is used for outputting alphanumeric data using an OCR-B font.

Input data must be alphanumeric data.

In addition to field type (OCRB), data length (1 to 999) can also be entered in a field info file.

**Radio button**

This field is used for making a single selection from the choices available.

Input data must be a string used when the radio button is selected.

In addition to field type (Radio), value for selected state (1 to 255 byte string) can also be entered in a field info file.

**Check box**

This field is used for toggling an option or for selecting one or more options, by "Checking/Unchecking" it.

Input data must be a string used when the checkbox is selected.

In addition to field type (Check), value for selected state (1 to 255 byte string) can also be entered in a field info file.

#### Barcode field

This field is used for outputting a barcode.

Input data must correspond with barcode types.

In addition to field type (Barcode), data length (1 to 9999) can also be entered in a field info file.

#### Image field

This field is used for outputting media data such as a bitmap image.

Input data must be media data file name (file name only).

In addition to field type (Bitmap), data length (1 to 240) can also be entered in a field info file.

### • Description Format of Field Info Files

"•Selection of Field Type" Field info files are created according to the contents determined in .

A semicolon (;) at the start of a line indicates a comment line.

The following is the description format for the field info file.

```
[ITEMINFO]
NumericFieldName=Numeric,DataLength,DecimalPlaces
TextFieldName=MText,DataLength
BlockTextFieldName=RectMText,DataLength
OCR-BFieldName=OCRB,DataLength
RadioButtonName=Radio,ValueForSelectedState
CheckBoxName=Check,ValueForSelectedState
BarcodeFieldName=Barcode,DataLength
ImageFieldName=Bitmap,DataLength

[INFOMNG]
BASIC=1
```

#### [ITEMINFO]

This is the section where you enter field information.

Enter the field information according to the order of input data fields. For the structure of input data, refer to the following:

=> ["8.2.1 Understanding the Data Configuration"](#)

When entering a field name, information after the field name may be omitted except an "equal (=)" sign. The equal sign must be specified. Also, information other than the field name and "equal (=)" may be omitted; however, a "comma (,)" must be used when specifying additional information after omitted information so as not to change the order of the information.

For example, when omitting the data length for the numeric field in the above example:

```
NumericFieldName=Numeric,,DecimalPlaces
```

#### Field name

Enter a field name according to the following naming conventions.

For details of the naming rules, refer to the Designer Help guide.

- Specify a name within 30 characters which consisted of Japanese characters (available symbols are only hyphen ( - ), katakana-hiragana prolonged sound mark (ー), recursive symbol (々) and minus (-)) and alphanumeric characters (A-Z, a-z, 0-9, hyphen (-), and underscore (\_)).
- Spaces cannot be used in a name.
- Minus (-), hyphen ( - ) or underscore ( \_ ) cannot be used at the head or end of a field name.
- A field name containing only half-width numerics cannot be used.
- Two or more consecutive underscores ( \_ ) cannot be used in a field name.
- Mixture of Japanese and half-width alphanumeric characters cannot be used.
- Field name should not be duplicated.

**Field type**

Enter a field type. It is entered after a field name, separated with an equal sign (=) from the field name.

If the field type is omitted, the setting is regarded as a text field (MText).

**Data length**

Enter a data length within the maximum length that can be specified for each field type. Place a comma (,) between this value and other setting value.

If the data length is omitted, the setting is regarded as "12".

If a length that exceeds the maximum length is specified, the maximum length is set.

If a length that exceeds the minimum length is specified, the minimum length is set.

For radio buttons or check boxes, data length cannot be specified.

**Decimal places**

For numeric fields, enter the number of decimal places within the data length. Place a comma (,) between this value and other setting value.

If the number of decimal places is omitted, the setting is regarded as "0".

If a length that exceeds the data length is specified, a length the same as that of the data length is set.

If a length less than minimum value is specified, "0" is set.

**Value for selected state**

For radio buttons and check boxes, enter a string used when the field is selected. Place a comma (,) between this value and other setting value.

If the value for selected state exceeds 255 bytes, or if the value for selected state is omitted, "Select" is specified for the radio button. For a check box, "Check" is specified.

**[INFOMNG]**

This is a section where you enter the number of basic information.

**BASIC=1**


Enter the number of basic information. Be sure to specify "BASIC=1".

**Table 4.1** Information that can be entered in the ITEMINFO section in the field info file


| Field type   |           | Data Length | Decimal places                           | Value For Selected State |
|--------------|-----------|-------------|--|--------------------------|
| Numeric      | Numeric   | 1 to 18     | 0 to 18<br>(DataLength >= DecimalPlaces) | -                        |
| Text         | MText     | 1 to 2997   | -  | -                        |
| Block Text   | RectMText | 1 to 9999   | -  | -                        |
| OCR-B        | OCRB      | 1 to 999    | -  | -                        |
| Radio Button | Radio     | -           | -  | 1 to 255 byte string     |
| Check Box    | Check     | -           | -  | 1 to 255 byte string     |
| Barcode      | Barcode   | 1 to 9999   | -  | -                        |
| Image        | Bitmap    | 1 to 240    | -  | -                        |

-.: Cannot be entered.



 **Caution** - The following field types cannot be defined in the field info file:

- Static field
- Block static field
- Barcode field (U.S. Postal FIM)
- Date field
- Time field
- Post code field
- Hidden field (Record)
- Group item

 **Note** - Field information can be entered up to 2000 fields. However, due to restrictions on the size of the file that can be saved, the form style information may not be saved even if the number of fields is less than 2000. To reduce the file size without decreasing the number of fields, shorten the strings specified in the file, such as those in field names and values for selected state for radio buttons and check boxes.

• **Description example of field info files**

The following is a description example of a field info file.

```
[ITEMINFO]
IDPhoto=Bitmap,12
Name=MText,6
EmployeeNumber=Numeric,6,0
EmploymentStartDate=Numeric,8,0

[INFOMNG]
BASIC=1
```

**4.4.3 Using Schema File (schema.ini)**

You can acquire required information from a database schema file (schema.ini) and use it as field information.

• **Definition of Schema File (schema.ini)**

Schema file (schema.ini) is an ini file in text format, in which database fields and structure are defined. Fields defined in the schema file (schema.ini) can be viewed in the record list in the [Form Style Descriptor] window when creating new form style information. You can define the fields by dragging and dropping them into the edit area.

• **Creation of field information from a schema file (schema.ini)**

Field information can be created from information in a schema file (schema.ini). The following explains the description format of the schema file (schema.ini), and field information created from the schema file (schema.ini).

```
[Section name]
:
:
Coln="ColumnName" type [width n]
```

[Section name]  
 The name of a database table.  
 This name can be viewed by the wizard when creating new form style information.

Coln="ColumnName" type [width n]  
 This is the section where you enter field information ("n" is a numeric character).

**ColumnName**

A field name.  
 If the same field name exists, a four-digit number is added to the original name.  
 Text -> Text0001

**type**

The field type is generated according to "type".  
 Field types that can be created are shown below:

**Table 4.2** Field types created from the schema file (schema.ini)

| "type" in the schema file (schema.ini) |   | Field type                   |
|--|---|------------------------------|
| Bit                                    | Bit type                                    | Numeric field (Numeric)      |
| Byte                                   | Byte type                                   |                              |
| Short                                  | Integer type                                |                              |
| Integer                                |   |                              |
| Long                                   | Long integer type                           |                              |
| Currency                               | Currency type                               |                              |
| Single                                 | Single-precision floating point number type |                              |
| Double                                 | Double-precision floating point number type |                              |
| Float                                  |   |                              |
| DateTime                               | Date/Time type                              |                              |
| Date                                   | Date type                                   |                              |
| Text                                   | Text type                                   | Text field (MText)           |
| Char                                   |   |                              |
| Memo                                   | Memo type                                   | Block text field (RectMText) |
| LongChar                               |   |                              |

**[width n]**

Data length ("n" is a numeric character).  
 When this value is omitted, "12" is specified.  
 When the value of "width" exceeds the maximum data length, the maximum data length for the corresponding field type is specified.



**Caution**

- When a value specified in ColumnName is not in accordance with the naming conventions for List Creator, it is modified to conform to the conventions. If all characters in a value do not conform to the naming conventions, the value is modified as follows:
    - When ColumnName is numeric field  
 4 digits serial number is attached to "Numeric" field.  
 Ex. Numeric0001
    - When ColumnName is text field  
 4 digits serial number is attached to "Text" field.  
 Ex. Text0001
- For details regarding the naming standards for List Creator, refer to the Designer Help guide.

#### 4.4.4 Preparing DTD/XML File

When a DTD/XML file is used, the DTD or XML file must be created in a format that List Creator supports.  
 For the supported format, refer to the following:

=> ["D.2 Conditions under which XML Files and DTD Files can be Used in List Creator"](#)

# Chapter 5

## Creating Form Layout Data

This chapter explains the following methods for creating form layout:

| Using resources (*1)                                    | Methods for Creating  | Reference   | Remarks  |
|---|---|---|--|
| No existing resources, or do not use existing resources | Locating fields from field entry list                       | => <a href="#">"5.1.1 Using Field Entry List"</a>                                   | -  |
| Use existing resources                                  | Using layout data from an existing form or preprinted paper | => <a href="#">"5.1.2 Using Layout Data from Existing Form or Preprinted Paper"</a> | This section explains the method for converting a template image to overlay data, and automatically generating fields and overlays to create form layout data. |
|   | Using Field Information of Input Data                       | => <a href="#">"5.1.3 Using Field Information of Input Data"</a>                    | This section explains the method for importing field information from field information files or schema files (schema.ini) to create form layout data.         |
|   | Using DTD/XML File  | => <a href="#">"5.1.4 Using DTD/XML file"</a>                                       | This section explains the method for importing the data from DTD files or XML files to create form layout data.  |

\*1: Resource is the data that is related to the form you want to create.

For the method of allocating areas (parts) such as partitions, or the method of defining fields and overlays, refer to the Designer Help guide.

## 5.1 Creating Form Layout Data

Form layout data can be created either by using or not using the existing resources. This section explains the following methods.

- If there are no existing resources, or if you do not use existing resources, the following method is used.
  - Creating form layout data using field entry list
- If you use existing resources, the following method is used.
  - Using layout data from an existing form or preprinted paper
  - Using input data field information
  - Using a DTD file/XML file

Form layout data can easily be created by using the Form Creation Wizard (Select wizard, Specify whether to use Unicode character).

The first screen displayed in the form creation wizard to select the following options is called the Form Creation Wizard (Select wizard, Specify whether to use Unicode character).

- Whether the wizard is used or not
- Whether Unicode characters are defined or not

- **Display the Form creation Wizard (Select wizard, Specify whether to use Unicode character)**

The procedure for displaying Form creation Wizard (Select wizard, Specify whether to use Unicode character) is shown below.

**= Procedure =**

- 1) Select [Start] - [Programs] - [List Creator] - [Form Descriptor List] to start the [Form Descriptor List].  
=>The [Form Descriptor List] window is displayed.



**Note**

- The form style information you are creating will be stored in the form storage directory currently displayed. When you want to change the form storage directory, refer to the following:  
=> "[12.3 Specifying the Form Storage Directory](#)"

- 2) Select [File] - [New].  
=>The [New] window is displayed.
- 3) Enter a name and comments for the form descriptor information, and click the [OK] button.  
=>[Select Wizard, Specify whether to use Unicode character] window is displayed.



**Caution**

- When you specify a form name, note the following point:  
=> "[B.1.1 Cautions regarding Specification of Form Name](#)"

### 5.1.1 Using Field Entry List

The following explains the method for creating form layout data using the field information registered in the field entry list (Prfmwiz.ini file) without using field information.

Separate explanations are made for the free frame format and for the other formats of forms. This is because the fields registered in the field entry list cannot be selected and defined in the wizard in the free frame format, generating a difference in wizard operations between the free frame format and the other formats.

#### 5.1.1.1 Creating Form Layout Data in Free Frame Format

Use the following procedure to create form layout data in free frame format.

**= Procedure =**

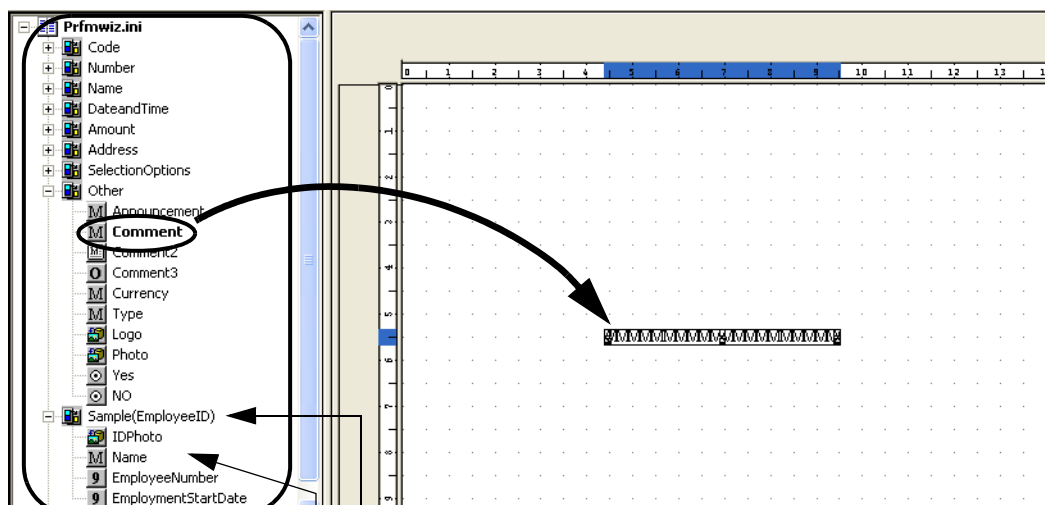
- 1) Display the Form creation Wizard (Select wizard, Specify whether to use Unicode character).
- 2) Select [New Default Form] and click the [Finish] button.  
=>The wizard is omitted and the [Form Style Descriptor] window is displayed.
- 3) Place the header/footer/frame/frame partition.  
For details of the operation method, refer to the Designer Help guide.
- 4) Place the field. Click the [Field Entry] tab at the bottom of the field list to display the field entry list.



=>The field entry list is displayed.

The field entry list displays the field information defined in the Prfmwiz.ini file.

- 5) Select a field in the field entry list and place it by dragging and dropping it to the edit area.  
=>When a field from the field entry list is placed in the edit area, its field name in the field entry list is displayed in bold.



Field entry list

Category names defined in the Prfmwiz.ini file

Field names in the category defined in the Prfmwiz.ini file

**Note** - The field entry list includes the fields defined in the Prfmwiz.ini file. They are candidates to be defined in form layout data. Unless the fields in the field entry list are actually defined in the edit area, they will not be saved in the form. For details regarding the Prfmwiz.ini file, refer to the Designer Help guide.

- 6) Define headings, fixed strings and rectangles to create a format.  
For details of the operation method, refer to the Designer Help guide.

### 5.1.1.2 Creating Form Layout Data for Other Formats

Use the following procedure to create form layout data for the formats other than the free frame format. The case of creating form layout data in single page is used here as an example.

**= Procedure =**

- 1) Display the Form creation Wizard (Select wizard, Specify whether to use Unicode character).
- 2) Select the form format (for this instance, specify "Single Page") and click the [Next] button.  
=>The paper setting window is displayed.
- 3) Set the information regarding the paper, and click the [Next] button.  
Since template image is not used here, specify nothing in the [Template Image].  
=>A details setting window is displayed for each form format.  
This window will differ according to the selected format.
- 4) Set the form style descriptor details for each form format, then click the [Next] button.  
=>The file referencing window is displayed.

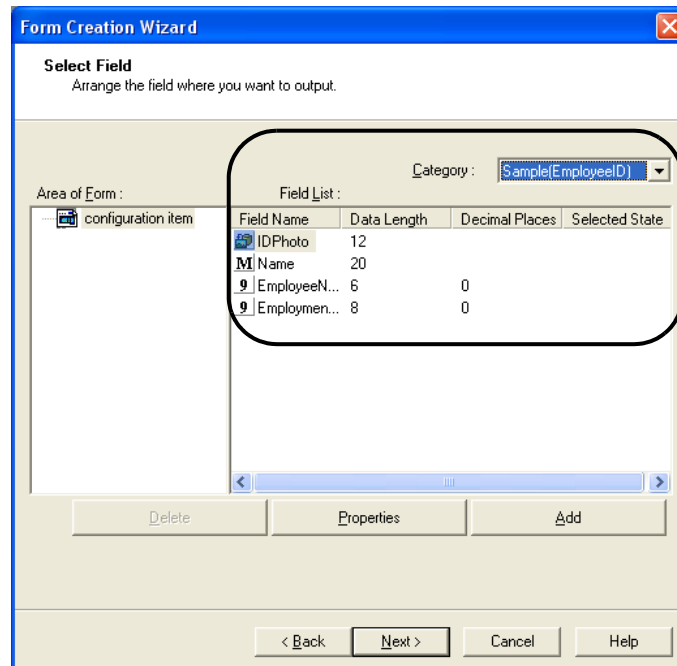
**Note**

- The second details setting window may be displayed depending on the selected format. Specify necessary information and click the [Next] button, and the file referencing window is displayed.
- For the forms in the simple table, block repeat or sticky label format, specify the repeat count in body (details, block layout, number of labels). The numbers set here can be changed after the wizard is closed. They can be changed in the [Repeat Properties] tab in the [Group Item] window which is displayed by selecting "BODY" in the record list and clicking the [Properties] button on the format toolbar. For details of the operation method, refer to the Designer Help guide.

- 5) Select "None", then click the [Next] button.  
=>The field selection window is displayed.
- 6) Select the fields to be located in the form. Select the category in the [Category] list box.  
=>A list of the fields defined in the specified category is displayed in the "Field List".

**Note**

- The categories defined in the Prfmwiz.ini file are displayed in the [Category] list box. The fields listed in the "Field List" are those defined in each category. The categories displayed in the [Category] list box and the "Field" defined in each category can be edited and used according to the form being designed. For the editing method, refer to the Designer Help guide.



- 7) Add necessary fields to "Area of Form". Select a field to be defined in the form style information from the "Field List", and click the [Add] button.  
 =>The selected field is added to the "Area of Form" on the left of the window.

**Note**

- You can also add a field with either of the following operations:
  - Select a field in "Field List" and drag and drop it to "Area of Form" on the left of the window.
  - Double click a field in "Field List".
- Select a field to change, and click the [Properties] button. The [Properties] window is displayed and the properties can be changed. The information changed in the [Properties] window will be saved in the Prfmwiz.ini file. (When designing another form style information, the changed Prfmwiz.ini file can be used for defining a field.) Set access privileges to the Prfmwiz.ini file for users to read/change the file. The user may not be able to design a form if the user does not have access privileges to read and change the file.

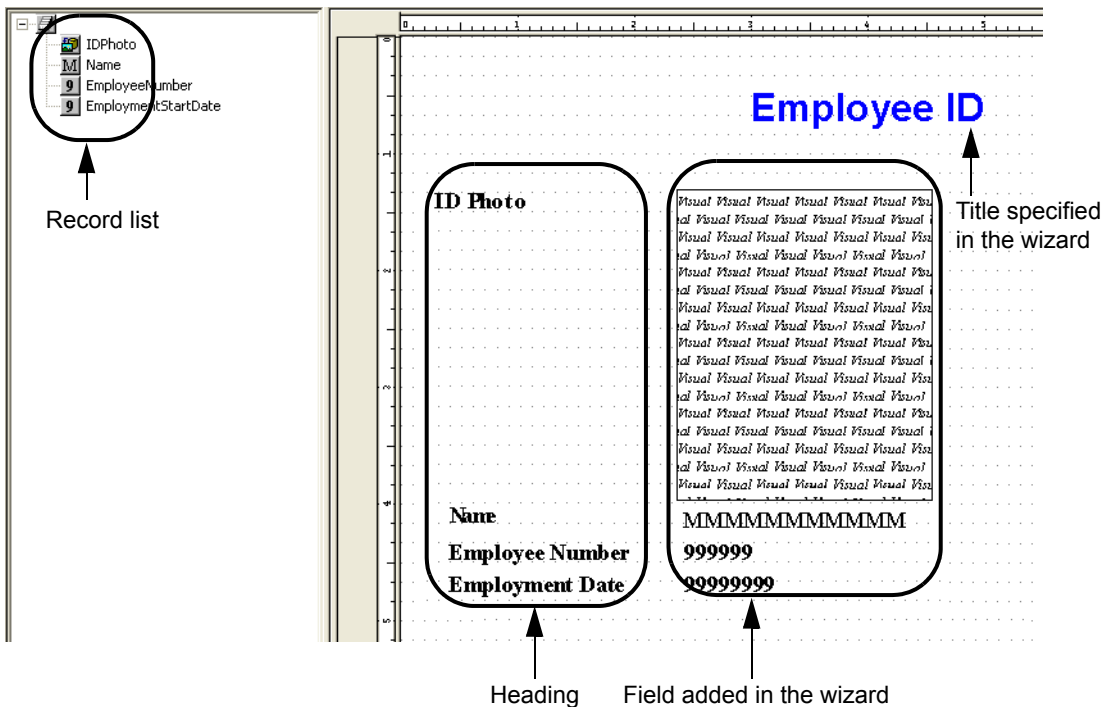
**Caution**

- Fields that cannot be defined in the field selection window are set according to the default values for those fields. Set the appropriate default values for fields beforehand. The default values for fields can be set in the window displayed by, on the [Form Style Descriptor] window, clicking [Tools] - [Default Values] - [Fields].

- 8) Add necessary fields, then click the [Next] button.  
 =>The finish window is displayed.
- 9) Check the "Add field names to the form" checkbox as necessary.  
 If the "Add field names to the form" checkbox is checked, a static field using the field name added in step (7) as its heading is created automatically.
- 10) Click the [Finish] button.  
 =>The [Form Style Descriptor] window is displayed after the wizard is closed.

The [Form Style Descriptor] window appears as follows:

- Titles specified in the wizard and the fields selected in the wizard are automatically created.
  - If the "Add field names to the form" checkbox is checked in the last window of the wizard, a static field using the field name as its heading is also automatically created.
  - The record list is displayed.
- Hidden fields (records) and group items can be added from the record list.



Click the [Partition] tab below the list to display the partition list. The list of fields in the partitions configured in the form is displayed.

Partition list

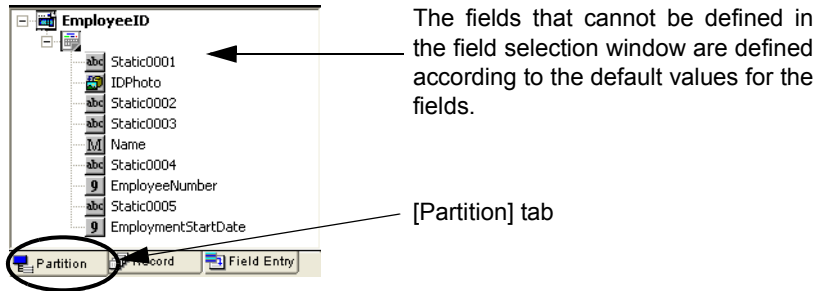


Figure 5.1 Partition list

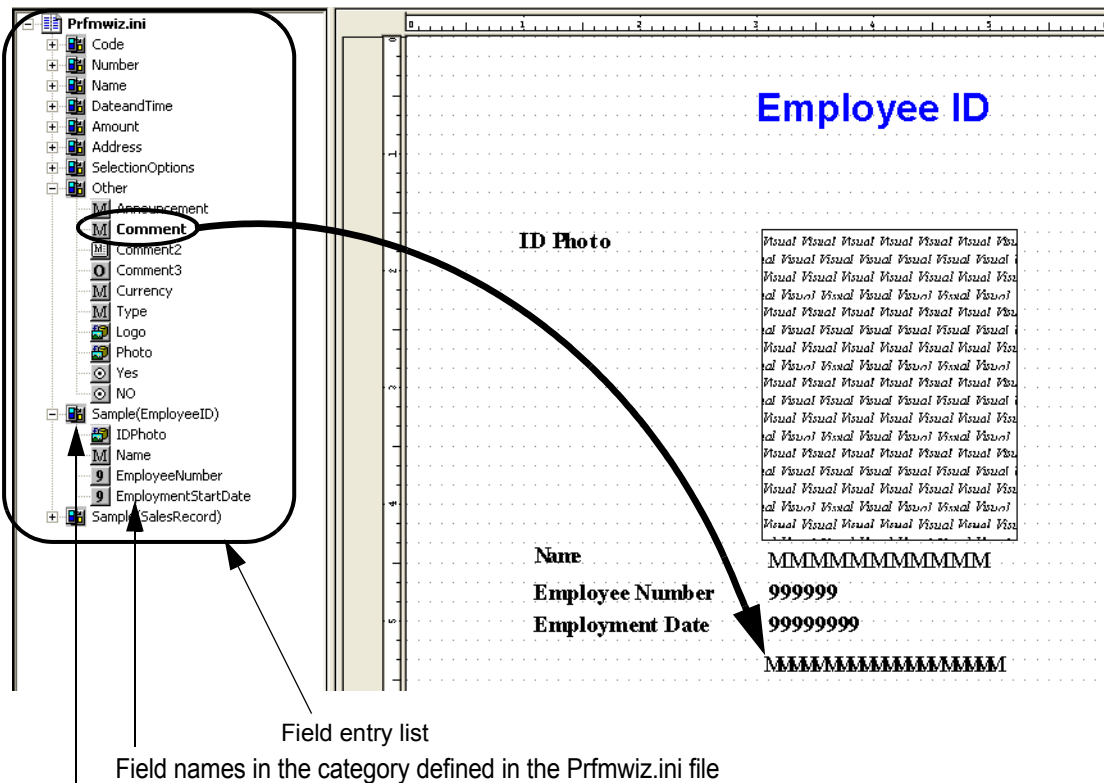
- To add a field, locate the field from the field entry list. Click the [Field Entry] tab at the bottom of the field list to display the field entry list.



=>The field entry list is displayed.

The field entry list displays the field information defined in the Prfmwiz.ini file.

- Select a field in the field entry list and place it by dragging and dropping it to the edit area on the right. =>When a field from the field entry list is placed in the edit area, its field name in the field entry list is displayed in bold.



Category names defined in the Prfmwiz.ini file

**Note** - The field entry list includes the fields defined in the Prfmwiz.ini file. They are candidates to be defined in form layout data. Unless the fields in the field entry list are actually defined in the edit area, they will not be saved in the form. For details regarding the Prfmwiz.ini file, refer to the Designer Help guide.



- 13) Move Prfmwiz.ini headings and visible fields as necessary. Also, define fixed strings and rectangles to create a format. For details of the operation method, refer to the Designer Help guide.

The image shows a screenshot of a form design tool interface. At the top, there is a horizontal ruler with markings from 0 to 7. On the left side, there is a vertical ruler with markings from 0 to 6. The main area is a grid with a dotted pattern. At the top center, there is a rounded rectangular box containing the text "Employee ID" in blue. Below this, there is a larger rounded rectangular box containing the following text:  
**MMMMMMMMMMMMMM**  
**Employee Number : 999999**  
**Employment Date : DD-MMM-YYYY**  
To the right of the "Employment Date" field, there is a small rectangular box containing the text "Visual Visual Visual Visual Visual Visual". Below the large rounded rectangular box, the text "Fujitsu Limited" is centered. The entire form is set against a light gray background with a dotted grid.

## 5.1.2 Using Layout Data from Existing Form or Preprinted Paper

This method creates form layout data using layout data from an existing form or preprinted paper. The following explains the method for creating a template image of an existing form or preprinted paper, and automatically generating overlays from the template image with the "Convert template image to overlay" function.

For the method for creating a template image, refer to the following:

=> ["4.4.1 Creating an Image File"](#)



**Caution**

- When using the overlay conversion function, refer to the following for the template image resolution.

=> ["4.4.1 Creating an Image File"](#)

Use the image file created from the preprinted paper as shown below to create form layout data.

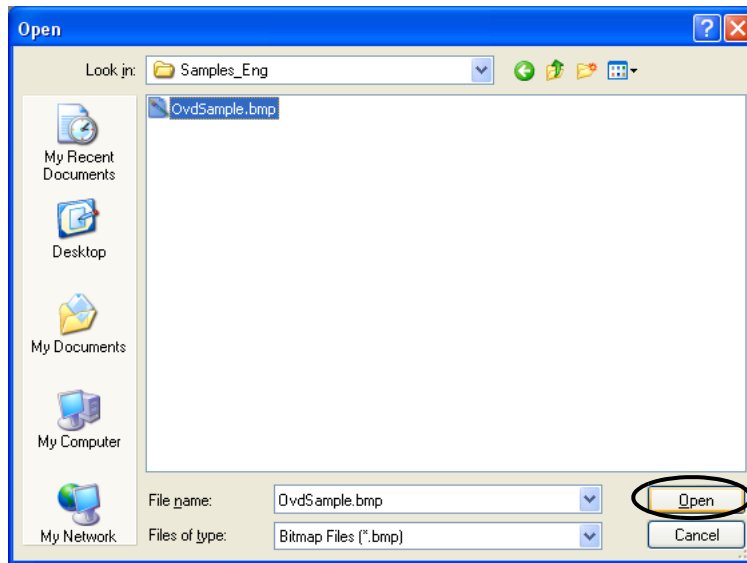
The case of creating form layout data in free frame format using a monochrome template image is used here as an example.

|              |                           |      |  |
|--------------|---------------------------|------|--|
| Test Results |                           | Name |  |
| Score        | 1 <sup>st</sup> term exam |      |  |
|              | Mid-term exam             |      |  |
|              | End of year exam          |      |  |
| Total        |                           |      |  |

### = Procedure =

- 1) Display the Form creation Wizard (Select wizard, Specify whether to use Unicode character).
- 2) Select "Free Frame", then click the [Next] button.  
=>The paper setting window is displayed.
- 3) Set the information regarding the paper, and click the [Define] button in the "Template image".  
=>The [Open] window is displayed.

4) Select an image file to be displayed as the template image, and click the [Open] button.



**Note**

- When a paper form (preprinted paper) is scanned in, the scanned image may be slightly tilted. To correct the tilt and display the image as the template image, check the [Compensate for image skew] checkbox.
- If a colored image is selected, a message is displayed asking whether to display the template image in color or in monochrome. Click the [Yes] button to display the template image in color. Click the [No] button to display the template image in monochrome. When displaying the template image in color, specification in [Compensate for image skew] will be invalid.

=>Return to the [Paper settings] window.

5) Click the [Next] button.

=>The file referencing window is displayed.

**Note**

- When the form format is not the free frame format, a details setting window is displayed before the file referencing window is displayed. Two details setting windows may be displayed for some form formats. Specify necessary information and click the [Next] button, and the file referencing window is displayed.

6) Select "None", then click the [Next] button.

=>The finish window is displayed.

**Note**

- When the form format is not the free frame format, the field selection window is displayed. Select the category from the [Category] list, select the field to be placed in the form from the field names displayed in the [Field list], and then click the [Add] button. Add all the necessary information and click the [OK] button, and the finish window is displayed. For details, refer to the following:  
=> "5.1 Creating Form Layout Data"

7) Select [Convert the Template image to an Overlay pattern], and click the [Finish] button.

**Note**

- If you do not use the overlay convert function, click the [Finish] button without checking the [Convert the Template image to an Overlay pattern] checkbox. In this case, the template image is displayed in the edit area in the [Form Style Descriptor] window.

=>The [Form Style Descriptor] and [Convert Template Image To Overlay] windows are displayed. The template image is displayed in the [Form Style Descriptor] window.

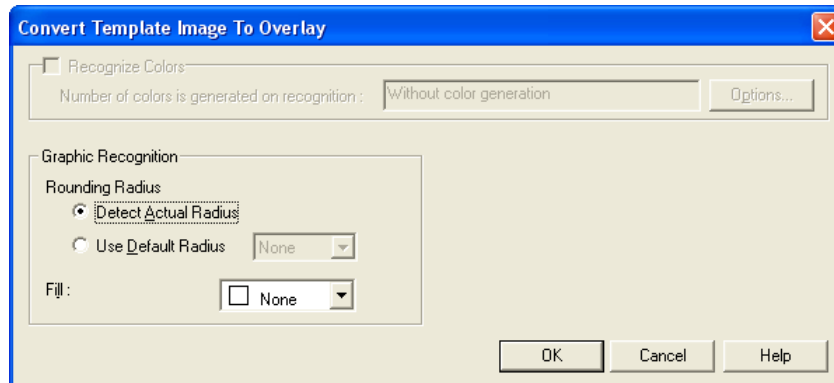
8) Configure the overlay conversion and specify the fields to be generated automatically, and click the [OK] button.

Here, you will set the followings:

Graphic Recognition  
 Rounding : Detect Actual Radius  
 Fill : None

**Note**

- [Recognize Colors] checkbox cannot be specified for a monochrome template image. It can be specified only for a color template image. To convert a color image to overlays in color, check this checkbox.
- For the points to note when converting an image file to overlays, refer to the Designer Help guide.



=>The template image is converted to overlays, and [Automatic Correction Position] window is displayed.

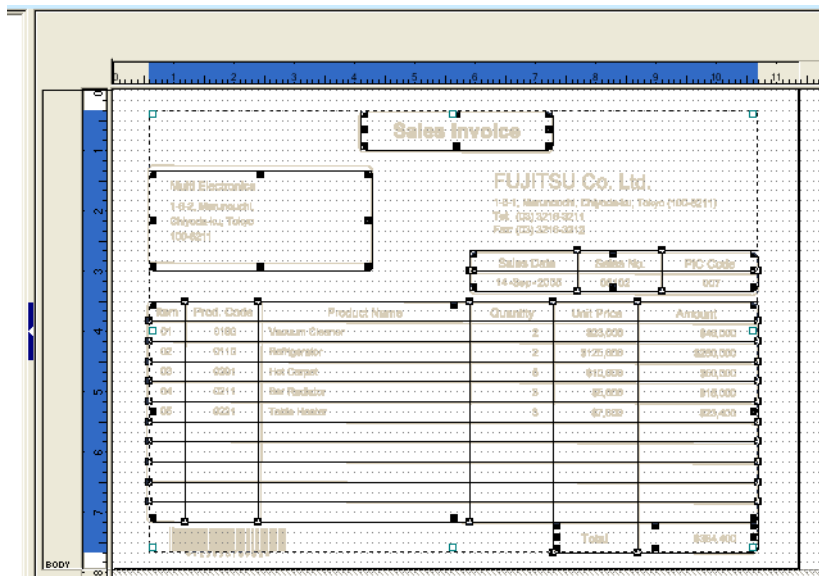
9) To correct the positions of the fields by aligning it to the gridlines, specify "Snap to Grid" on the [Automatic Correction Position] window and click the [OK] button.

Here, set nothing and click the [Cancel] button.

=>The display returns to the [Form Style Descriptor] window.

The [Form Style Descriptor] window appears as follows:

- The template image is converted to overlays and displayed in the edit area.



10) Place the header/footer/frame/frame partition.

For details of the operation method, refer to the Designer Help guide.

**Note**

- Place frames and frame partitions only when the form format is the free frame format. In the example, there is no need to place frames or frame partitions. Align them as necessary.

- 11) Create headings and visible fields as necessary. Also, define fixed strings and rectangles to create a format. For details of the operation method, refer to the Designer Help guide.



**Note**

- To hide the template image, click the [Template Image] button on the View toolbar while the template image is displayed.
- If the template image contains the margins of the preprinted paper, subtract the margin value when specifying the position to display the template image in the template image window. Also, if the bottom right of the template image exceeds the boundaries of the paper, adjust the position of the template image so that it fits within the paper.
- For Windows, the form images designed using favorite tools such as Microsoft Word or Microsoft Excel, or the form images of PDF files published on the Internet can be converted to overlays using the List Creator Overlay Utility, and easily read into List Creator forms.

### 5.1.3 Using Field Information of Input Data

The following explains the method for creating form layout data based on the field information of input data.

When using field information from schema files (schema.ini), the files are necessary.

For schema files (schema.ini), refer to the following:

=> ["4.4.3 Using Schema File \(schema.ini\)"](#)

When using other methods, field info files are necessary.

For the method for creating a field info file, refer to the following:

=> ["4.4.2 Creating Field Info Files"](#)

The case of creating form layout data in free frame format is used here as an example.

#### = Procedure =

- 1) Display the Form creation Wizard (Select wizard, Specify whether to use Unicode character).
- 2) Select "Free Frame", then click the [Next] button.  
=>The paper setting window is displayed.
- 3) Set the information regarding the paper, and click the [Next] button.  
Since template image is not used here, specify nothing in the [Template Image].  
=>The file referencing window is displayed.



#### Note

- When the form format is not the free frame format, a details setting window is displayed before the file referencing window is displayed. Two details setting windows may be displayed for some form formats. Specify necessary information and click the [Next] button, and the file referencing window is displayed.

- 4) Select "Refer to the File", and specify one of the following for the file to browse, and then click the [Next] button.

#### Field info file

Select this when the input data is Variable Length, Fixed Length or CSV Format. Specify the field info file name with its full path.

#### Schema file (schema.ini)

Select this when creating field info from a schema file (schema.ini). Specify the schema file name with its full path.

File names can also be specified from the open window, which is displayed by clicking the [Browse] button to the right of the file name input area.

The following operations differ depending on the selected file.

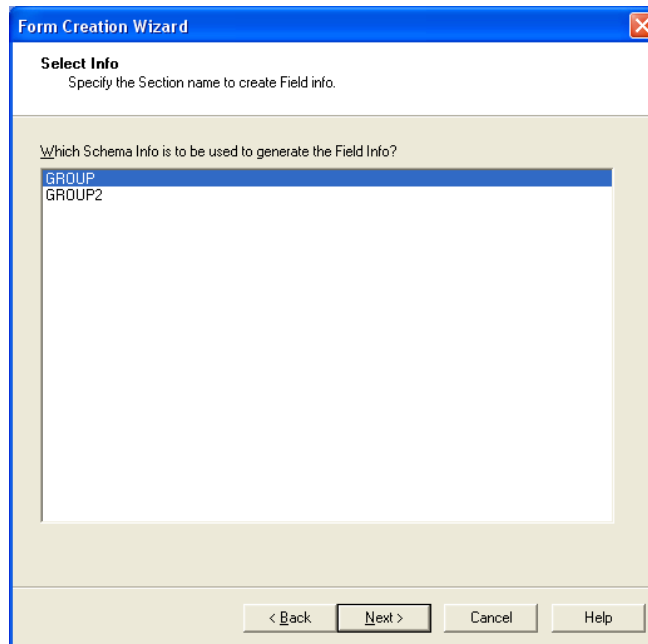
#### When using a Field Info File

Proceed to Step (6)

#### When using a schema file (schema.ini)

Proceed to Step (5).

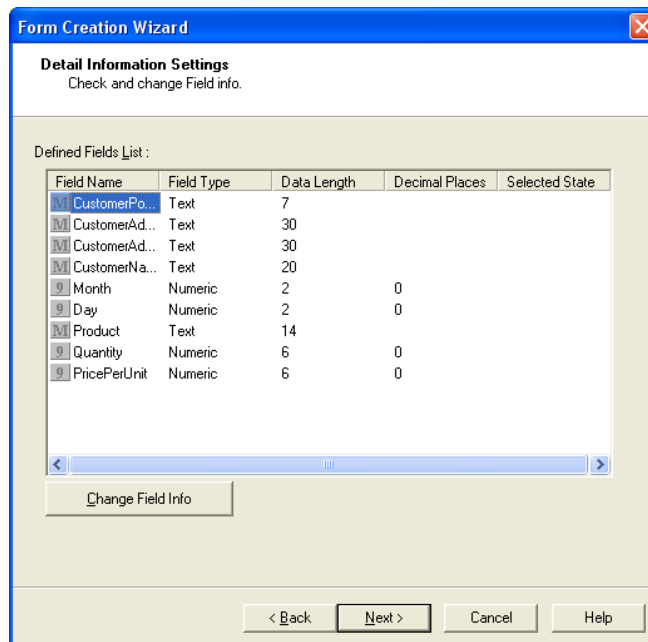
- 5) Select the table from which the field information is to be created, and click the [Next] button.



=>The details setting window for field information is displayed.

**Note** - When the format is a simple table or block repeat, two details setting windows for field information are displayed. Specify which field is placed in which area (head/body/tail), and click the [Next] button. Place one field in the body area. Fields for which no location was specified are automatically located in the body area.

6) Click the [Next] button.



**Note** - Field information can be changed in the Change Field Info window. Selecting the field name of the field to be changed and then clicking the [Change Field Info] button displays the [Change Field Info] window. The changes made to the field information in the [Change Field Info] window are not reflected in the field info file and schema file (schema.ini). For details of the change field info window, refer to the Designer Help guide.



- Information that cannot be defined in the field info file is set according to the default values for those fields. Set the appropriate default values for fields beforehand. The default values for fields can be set in the window displayed by selecting [Tools] - [Default Values] - [Fields] in the [Form Style Descriptor] window.

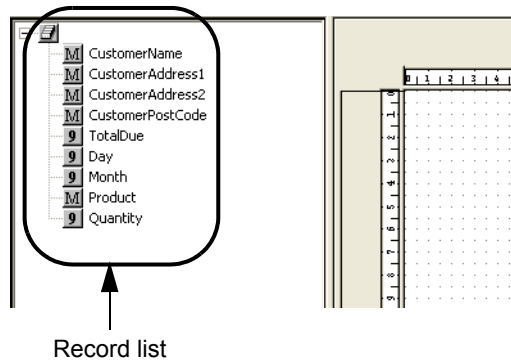
=>The finish window is displayed.

7) Click the [Finish] button.

=>The [Form Style Descriptor] window is displayed after the wizard is closed.



- When the form format is the free frame or sticky label format, the title specified in the wizard is displayed in the edit area.



8) Place the header/footer/frame/frame partition.

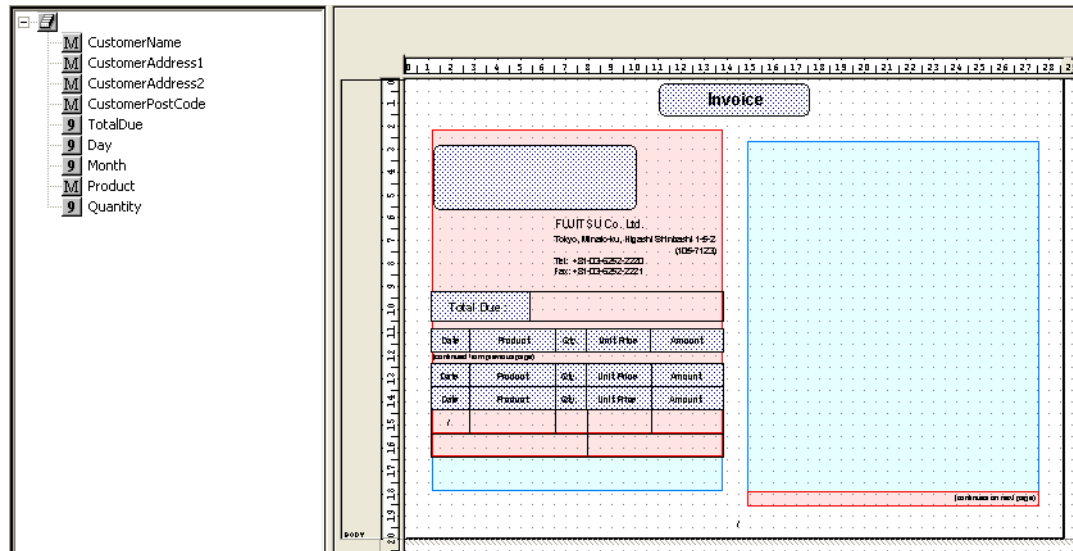
For details of the operation method, refer to the Designer Help guide.



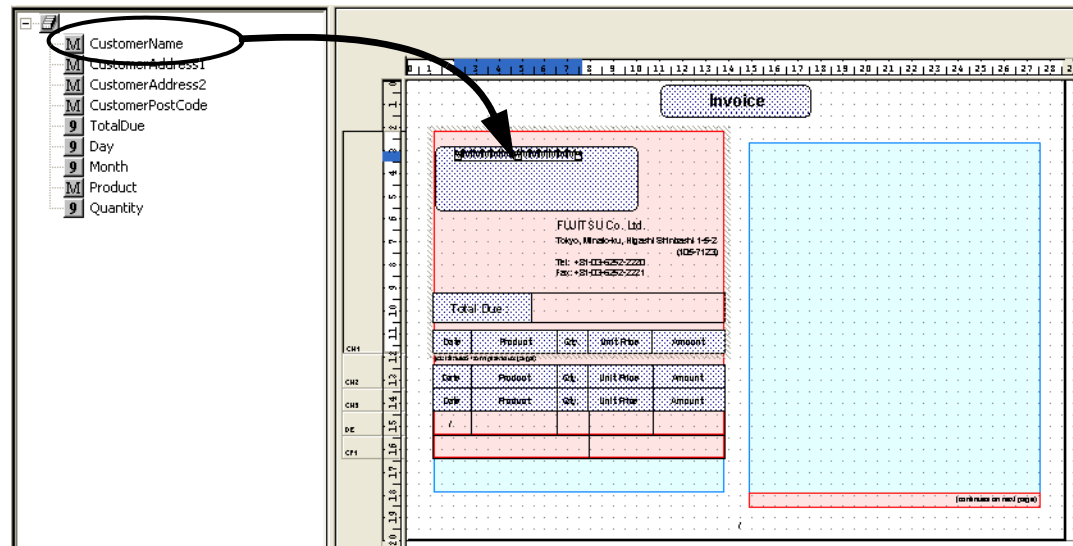
- Place frames and frame partitions only when the form format is the free frame format.



9) Define fixed strings and rectangles to create a format. For details of the operation method, refer to the Designer Help guide.



10) Place fields by dragging and dropping them from the record list to the edit area.



**Caution**

- To return a field defined in the edit area to an undefined state, select the field in the edit area by clicking on it and select [Format] - [Hide Field]. A field deleted from the edit area is also deleted from the record list.
- If the [Form Style Descriptor] window is closed, fields not located in the edit area are saved as hidden fields (records).

### 5.1.4 Using DTD/XML file

The following explains the method to create form layout data using a DTD file/XML file and to extract those files to data list. Also, the data structure of form layout data and of input data in XML format is explained.

### 5.1.4.1 Using DTD/XML File

The method for creating form layout data referring to a DTD file or input data in XML format is explained below.



**Note**

- For DTD files and XML files that can be used in List Creator, refer to the following:  
=> ["D.2 Conditions under which XML Files and DTD Files can be Used in List Creator"](#)
- When creating a form in the free frame format with input data in "XML format", it is recommended to refer to a DTD file in designing form layout data.
- For the conditions under which an element can be specified from the [Specify the Data Root] window of the Form Creation Wizard, refer to the following:  
=> ["D.4 Conditions under which Elements can be Specified as a Data Root"](#)
- In the [Contents Tree of the selected Data Root] in the [Specify the Data Root] window of the Form Creation Wizard, the structure of the specified data root is displayed. If a data root element is changed, the element is redisplayed as the data root. For the method for extracting the files to the data list from the [Form Style Descriptor] window after closing the wizard, refer to the following:  
=> ["5.1.4.2 Extracting to Data List"](#)
- For the conditions under which an element can be specified as body area group items in the [Body Area Group Item] window of the Form Creation Wizard, refer to the following:  
=> ["D.5 Conditions under which Elements can be Specified as Body Group Items"](#)
- For the conditions under which an element can be specified as details group items in the [Details Group Item : Select] window of the Form Creation Wizard, refer to the following:  
=> ["D.6 Conditions under which Elements can be Specified as Details Group Items"](#)

When using an XML file for the input data, also refer to the following:

=> ["Appendix D XML Files Used in List Creator"](#)

The case of creating form layout data in free frame format is used here as an example.

**= Procedure =**

- 1) Display the Form creation Wizard (Select wizard, Specify whether to use Unicode character).
- 2) Select the form format to be created and click the [Next] button.  
=>The paper setting window is displayed.



**Note**

- For form format selection, refer to the following:  
=> ["D.3 Specifying the Form Format responding to the XML Data Configuration"](#)

- 3) Set the information regarding the paper, and click the [Next] button.  
Since template image is not used here, specify nothing in the [Template Image].  
=>The file referencing window is displayed.



**Note**

- When the form format is not the free frame format, a details setting window is displayed before the file referencing window is displayed. Two details setting windows may be displayed for some form formats.  
Specify necessary information and click the [Next] button, and the file referencing window is displayed.

- 4) Select one of the following, then click the [Next] button:

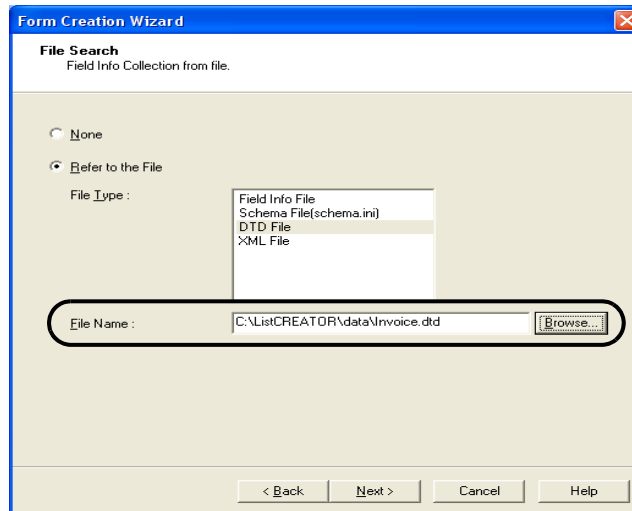
**DTD file**

Select this when creating form layout data referring to a DTD file. Specify the DTD file name in the "File Name" with its full path.

**XML file**

Select this when creating form layout data referring to a XML file. Specify the XML file name in the "File Name" with its full path.

File names can also be specified from the window displayed by clicking the [Browse] button to the right of the file name input area.

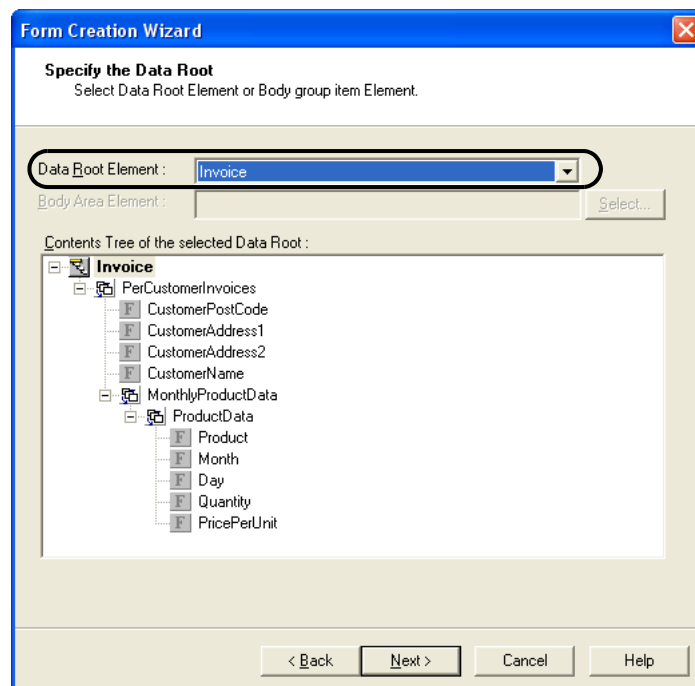


- 5) Click the [Next] button.  
=>The data root specification window is displayed.

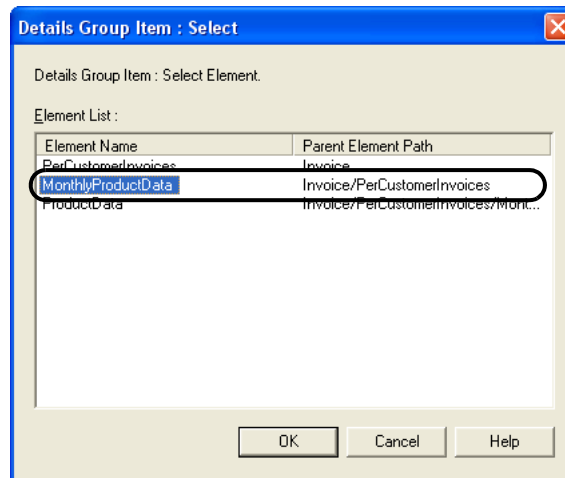
- 6) Specify the data root element.  
The following operations differ depending on the form format.

For the free frame or single page format  
Proceed to Step (9).

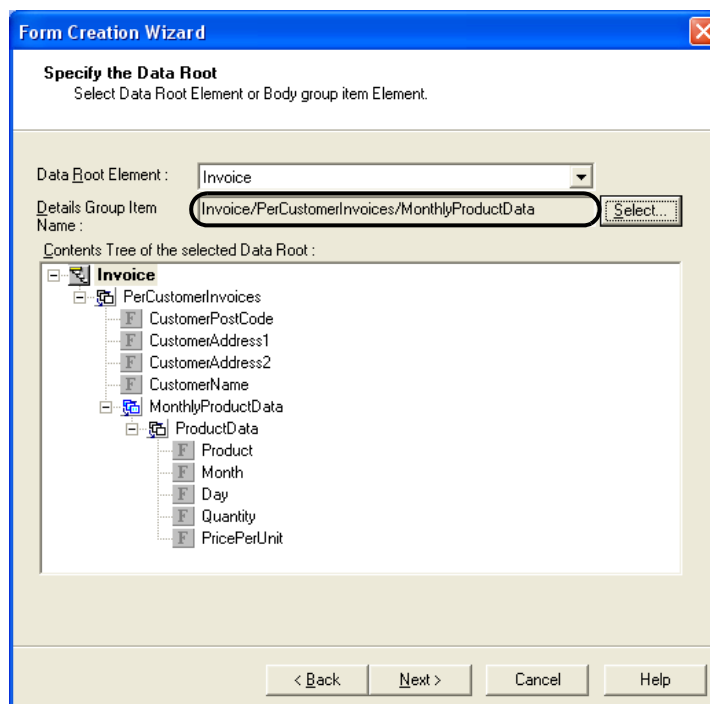
For the simple table, multipart table, block repeat, or sticky label format  
Proceed to Step (7). Also specify the body area element or details area element.



- 7) When the format is simple table, multipart table, block repeat or sticky label, select an element to be used as the body area group item or details group item, from the [Body Area Group Item] window or the [Details Group Item : Select] window displayed by clicking the [Select] button.



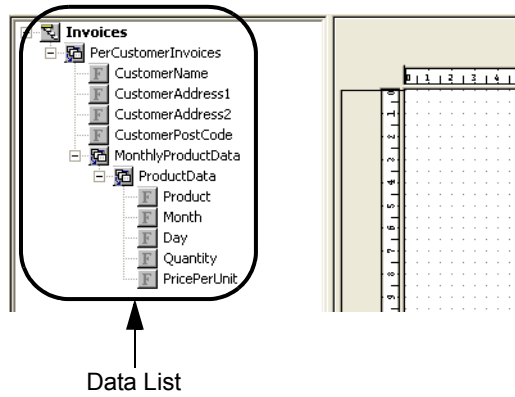
- 8) Click the [OK] button.  
=>The display returns to the data root specification window.



- 9) Check the structure under the data root, then click the [Next] button.  
=>The finish window is displayed.
- 10) Click the [Finish] button.  
=>The [Form Style Descriptor] window is displayed after the wizard is closed. The [Form Style Descriptor] window appears as follows:
- The data structure and XML tags from the DTD file or the XML file are extracted to the data list in the left.
  - All XML tags are displayed as hidden fields (records) or a group item.



- When the form format is the free frame or sticky label format, the title specified in the wizard is displayed in the edit area.



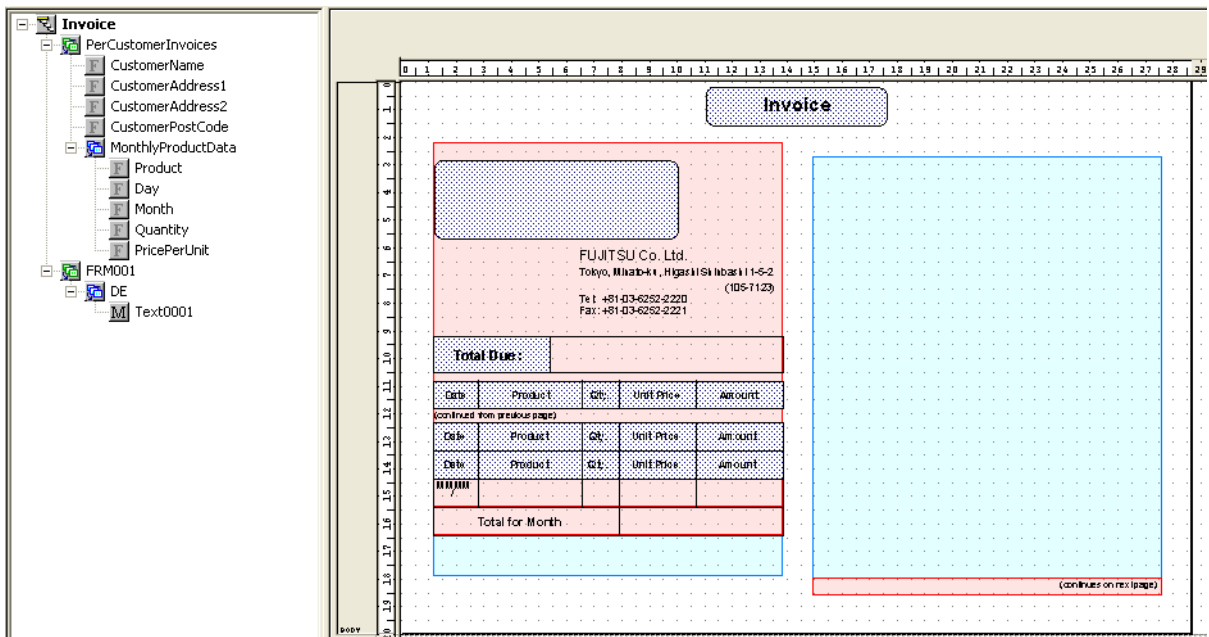
**Note** - The following shows the rule for allocating elements to each field when referring to a DTD file.

| Element-type declaration          | Element name | Field type   |
|-----------------------------------|--------------|--------------|
| <!ELEMENT Group1 (Field1,Field2)> | Group1       | Group item   |
| <!ELEMENT Field1 (#PCDATA)>       | Field1       | Hidden field |
| <!ELEMENT Field2 EMPTY>           | Field2       | Hidden field |


11) Place the header/footer/frame/frame partition. For details of the operation method, refer to the Designer Help guide.

**Note** - Place frames and frame partitions only when the form format is the free frame format.

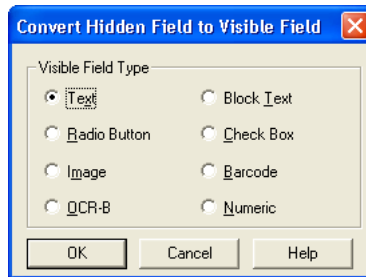
12) Define fixed strings and rectangles to create a format. For details of the operation method, refer to the Designer Help guide.



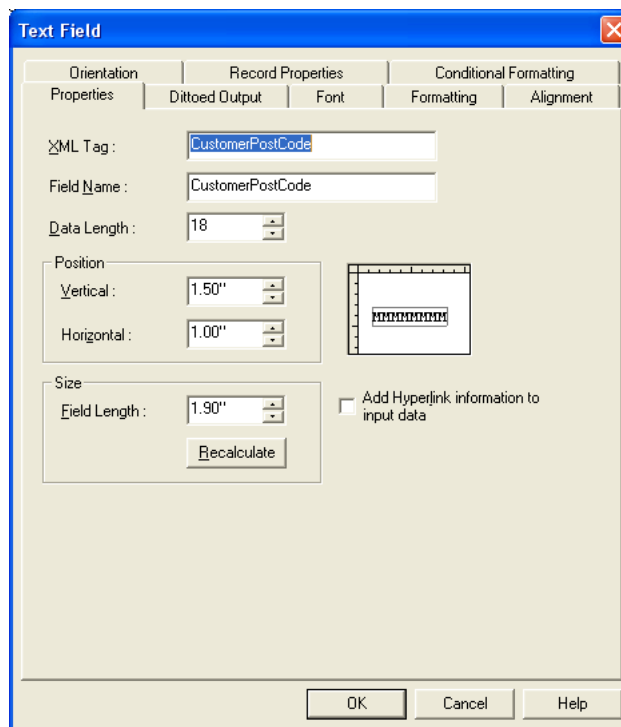
- 13) Change the data list structure.  
 For details, refer to the following:  
 => "5.1.4.4 Changing Data List Structure (for Free Frame Format Only)"

 **Note** - Change the data list structure only when the form format is the free frame format.


- 14) Place XML names by dragging and dropping them from the data list to the edit area.  
 Drag and drop XML names to the edit area.  
 =>The [Convert Hidden Field to Visible Field] window is displayed.



- 15) Specify the field type in [Visible Field Type] and click the [OK] button.  
 =>The properties window of the field is displayed.
- 16) Specify the information such as data length and formatting, and click the [OK] button.



=>The field is inserted in the location specified by the drag-and-drop operation in the edit area.

 **Note** - The location can also be specified in the field's [Properties] window. When the location is changed in the field's [Properties] window, the field is inserted in the changed location.

For the points to note when adding a new field or moving a field from the [Form Style Descriptor] window, refer to the following:  
 => "D.9 Points"

### 5.1.4.2 Extracting to Data List

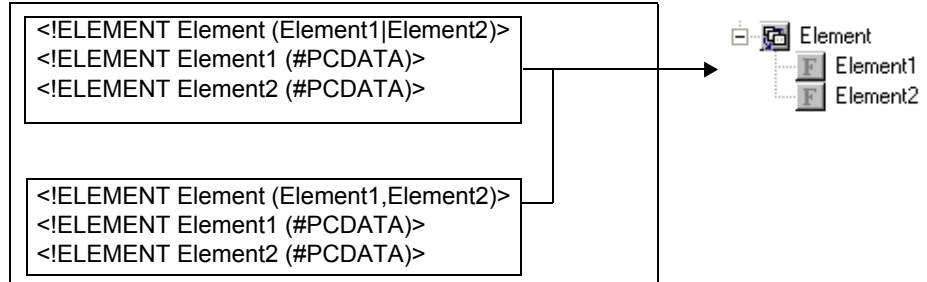
The following explains how DTD files or XML files are extracted to the data list.

- DTD files

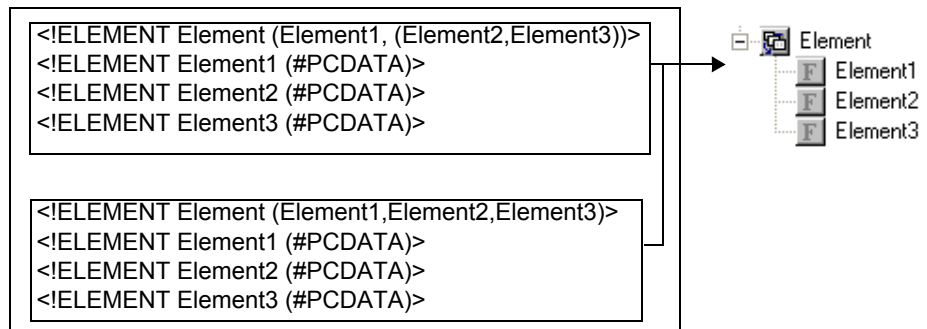
DTD files are extracted to the data list according to the following rules.

**Principle**

- Difference between "|" and ",", which indicate the appearance order, does not make difference in extracting rules.

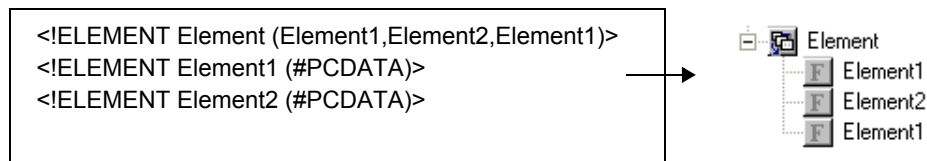


- There is no difference in extracting rules between an element enclosed in parentheses and one not enclosed.



**When the same element is defined**

- When the form format is the single page format, the same number of the elements is extracted to the data list as the number of times the element is defined in the DTD file.



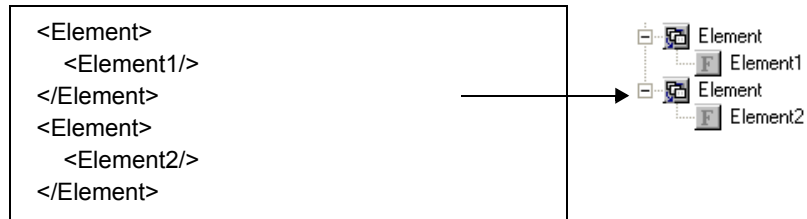
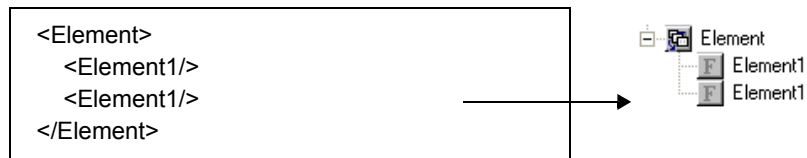
- When the form format is not the single page format, you cannot refer to a file containing an element defined more than once in one element hierarchy.

- XML files

XML files are extracted to the data list according to the following rules.

For the single page or free frame format

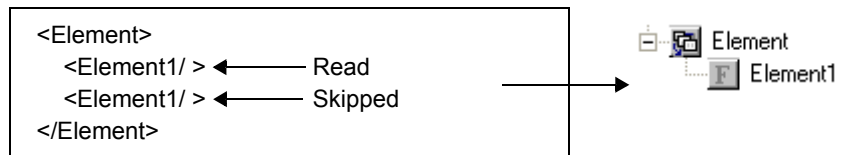
The data structure of the XML file referred to is extracted to the data list as it is.



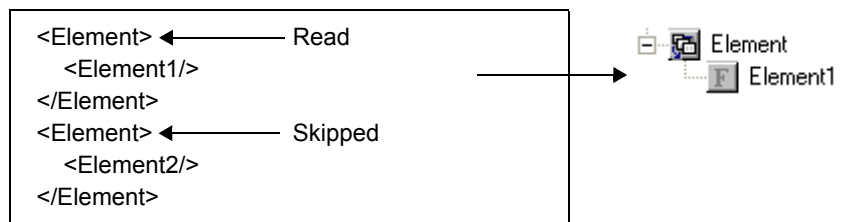
Formats other than single page or free frame format

If the same element appears more than once in one element hierarchy in the data structure of the XML file referred to, the first element that appears is considered to be valid and is read. The second elements and on are skipped.

When XML tag <Element> includes two <Element1> child tags, the first <Element1> is read. The second <Element1> is skipped.



When XML tag <Element> has <Element1> as its child element and another one has <Element2>, the first <Element> is read and the second <Element> is skipped.





### 5.1.4.3 Data Structure of Form Layout Data and XML Input Data

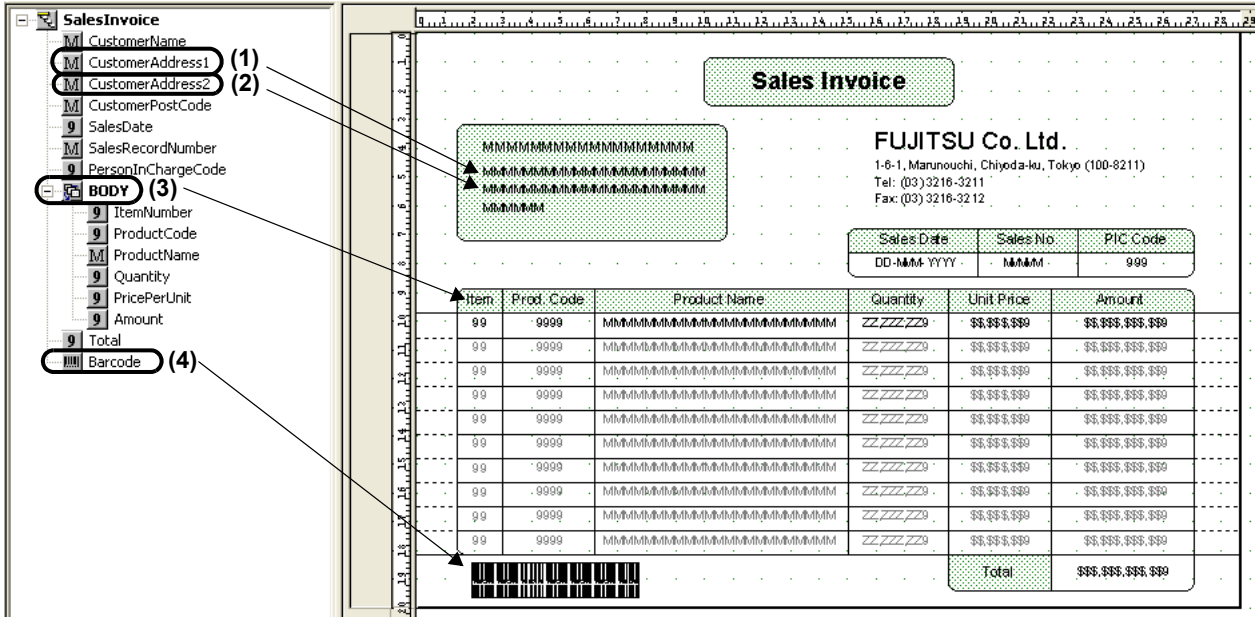
The data structure in data list can have a different order of tags from that in the input data in XML format. Even if the order is different, the data is output based on the tag name.

However, when the same tag names are defined in one hierarchy, the order of tags in the data list in the [Form Style Descriptor] window must be the same as in the input data.

An example is shown below.

- Form layout

The <Barcode> tag is defined after the <Detail> tag as shown in the form layout below.



- Input data

The following XML file is specified as the input data.

```

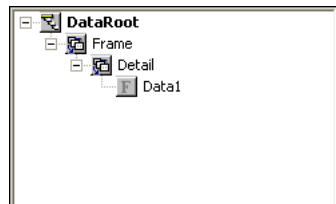
<?xml version="1.0" encoding="UTF-8" ?>
<SalesInvoice>
  <CustomerPostCode>0000000</CustomerPostCode>
  <CustomerAddress>Madison Avenue 411</CustomerAddress> (1)
  <CustomerAddress>Des Moines IA</CustomerAddress> (2)
  <CustomerName>ABC Electronics</CustomerName>
  <SalesDate>20031221</SalesDate>
  <SalesRecordNumber>03101</SalesRecordNumber>
  <ContactCode>005</ContactCode>
  <Barcode>1234567890</Barcode> (4)
  <Detail> (3)
    <ProductCode>0020</ProductCode>
    <ProductName>Color television</ProductName>
    <Quantity>3</Quantity>
    <UnitPrice>120000</UnitPrice>
  </Detail>
</SalesInvoice>
    
```

Two <CustomerAddress> tags are described as child elements of the <SalesInvoice> tag.

The <Barcode> tag is described before <Detail> tag.



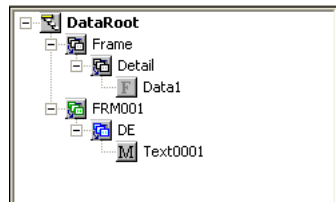
=>The data list is created as follows:



2) Place a frame.

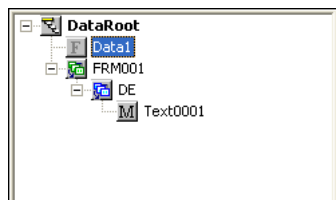
For details, refer to the Designer Help guide.

=>The following data list is displayed.



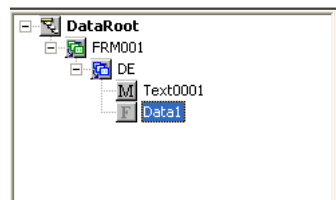
3) Delete group item "Frame" and "Detail".

=>The group items are deleted.



4) Move the hidden field "Data1" into the details group item "DE".

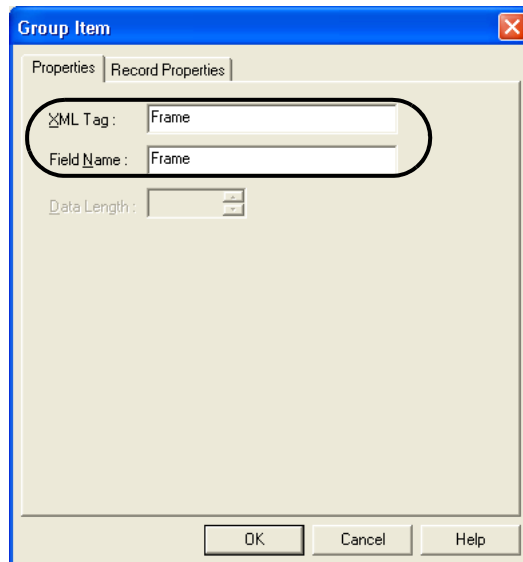
=>The following data list is displayed.



5) Select the frame group item "FRM001" from the data list, and select [Format] - [Properties].

=>The [Properties] window of the group item is displayed.

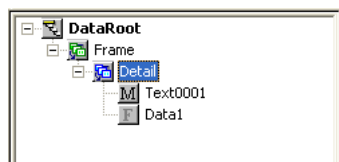
6) Change the XML tag to "Frame" and the field name to "Frame" in the [Properties] tab, and click the [OK] button.



=>The XML tag and the field name of "FRM001" are changed to "Frame".

7) Change XML tag in the details group item "DE" to "Detail" and the field name to "Detail" in the same way.

=>The following data list is displayed.



# **Chapter 6**

## *Definition of Association Data*

---

This chapter explains the procedures for defining association data between form layout data and input data.

## 6.1 Definition of Association Data

Association data contains definition of data to be overlapped on the form layout data (input data) and that of information for generating output data by automatically performing calculation using input data. The following information needs to be defined.

### Input data type

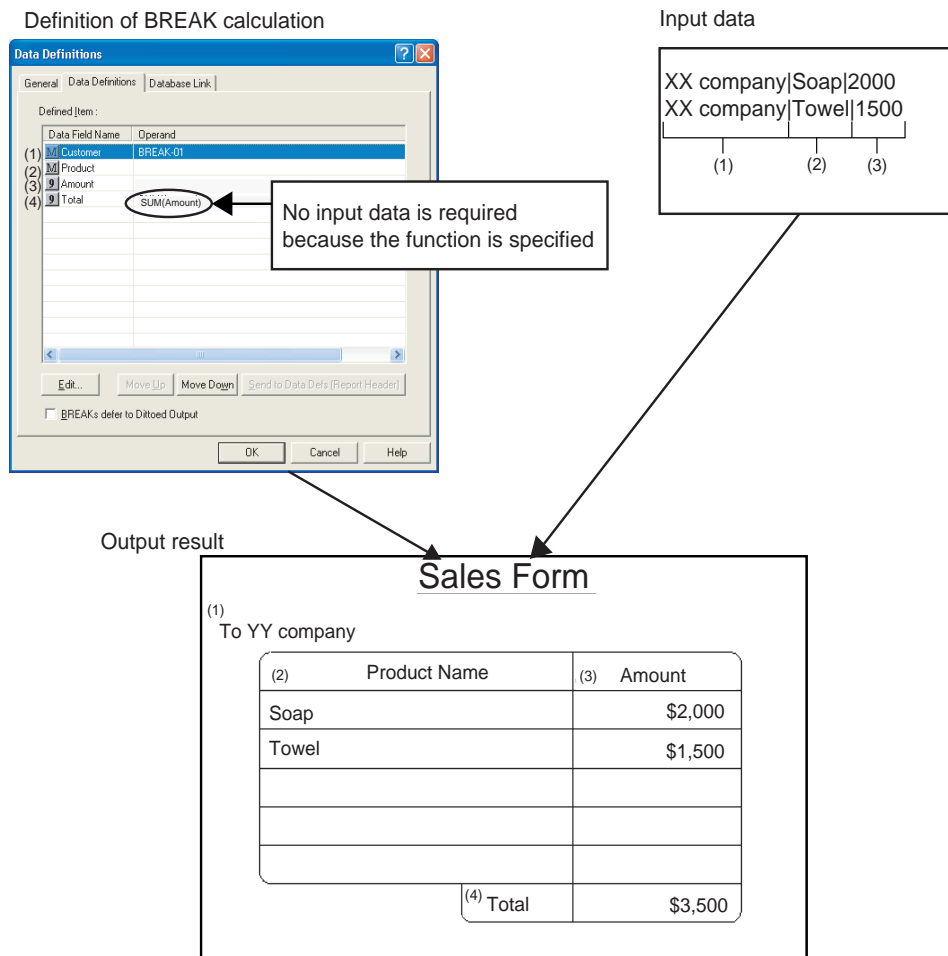
Define the type of input data. For input data types, refer to the following:  
=> "6.2.1 Determining the Type of Input Data"

### Output order of fields

Define the order of the fields in the form layout data so that it matches the order of the input data (refer to Figure 6.1).

### Definitions of calculations (automatic calculation of output data)

Define settings for automatically calculating a value for a specified field (such as a total) using input data (refer to Figure 6.1 and Figure 6.2).



**Figure 6.1** Relationship between the order of the fields in the form layout data and that of input data

Definition of BREAK control

Define BREAKs for specified fields.

BREAK control performs operations such as the following when outputting a form: creates a new page or performs a calculation, such as a total, when the value of the input data for the field with a BREAK is changed (when a BREAK occurs).

A field where a BREAK is defined is called a "BREAK field". A BREAK occurs when the data value of a BREAK field changes.

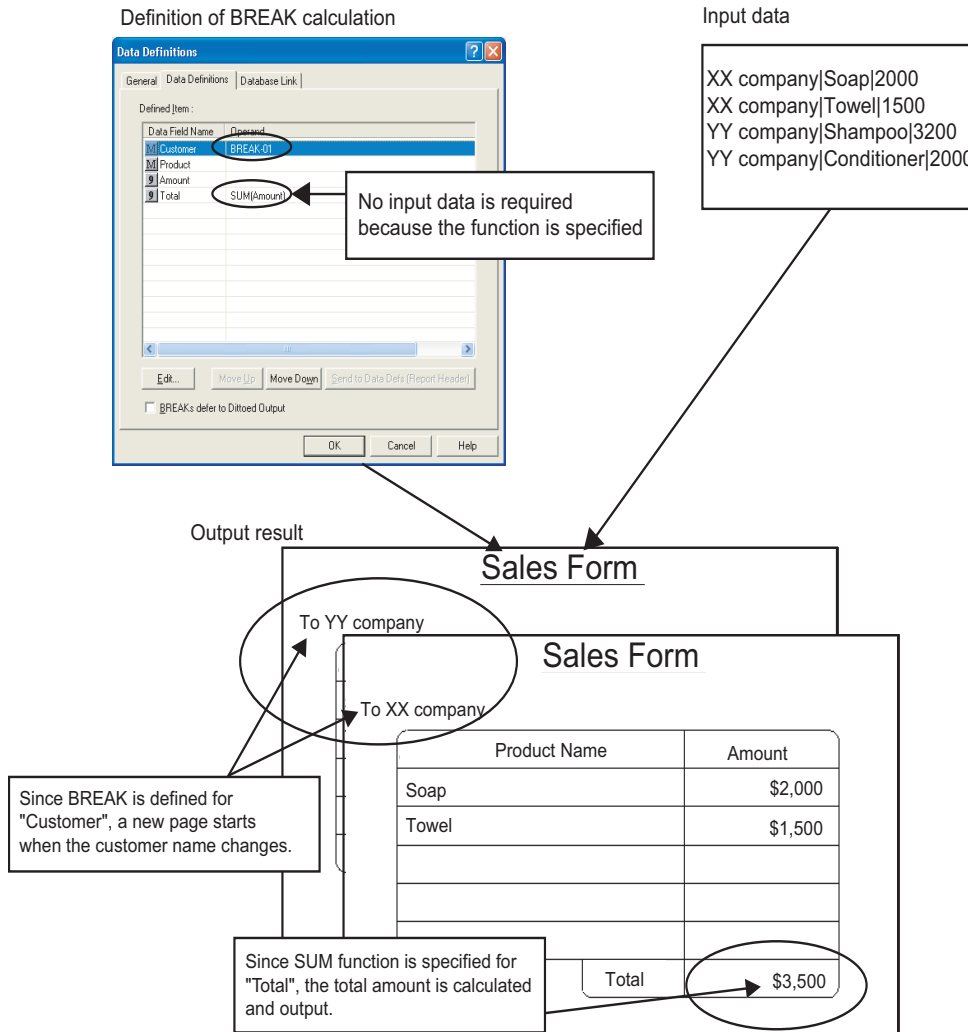


Figure 6.2 Relationship between BREAK and calculation

## 6.2 Selecting Information to be Defined for Association Data

This section explains the procedure for selecting information that associates form layout data with input data.

- Determining the Type of Input Data
- Checking Fields and Contents to which Operators are Specified
  - Check each field where a BREAK or calculation is defined.
  - Media data type of each image field
  - Fields where strings are operated
  - Fields where conditions are judged (CASE data output).
- Selecting the Output Order and Output Conditions of Frames and Partitions, and Timing of Calculations
  - For the free frame format, select the output order and output conditions of frames and partitions, and timing of calculations.
  - For the multipart table format, select the output order and output conditions of partitions, and timing of calculations.

The following describes each task.

### 6.2.1 Determining the Type of Input Data

Determine the type of input data to be used when outputting a form.

List Creator uses text data files such as in CSV or XML format, or data extracted from a linked database, as input data.

Select an input data type that is appropriate for your purpose.

For details such as input data types and format, refer to the following:

=> "[1.5 Input Data Type and Format](#)"

### 6.2.2 Checking Fields and Contents to which Operators are Specified

Check the following:

- Fields where calculations are defined
- Fields where BREAKs are defined
- Media data type of each image field
- Fields where strings are operated
- Fields where conditions are judged (CASE data output).



6.2.2.1 Types of Operators

Table 6.1 "Types of operators" in page 105 shows types of operators.

For the procedures and notes for specifying each operator, refer to the Designer Help guide.

**Table 6.1** Types of operators (1 / 3)

| Type               | Operator  | Definition   | Form format |             |                     |                                 |                                       |                     |              | Input data   |
|--------------------|---|--|-------------|-------------|---------------------|---------------------------------|---------------------------------------|---------------------|--------------|--------------|
|                    |   |  | Free frame  | Single page | Simple table format | Multipart table (Report header) | Multipart table (Other report header) | Block repeat format | Sticky label |              |
| BREAK              | BREAK- <i>nn</i>  | Detects break limit <i>nn</i> : Control level  | DD          | -           | DD                  | -                               | DD                                    | DD                  | -            | Required     |
| Media data         | MEDIA- <i>x</i>   | Outputs media data <i>x</i> : Media data type  | DD          | DD          | DD                  | DD                              | DD                                    | DD                  | DD           | Required     |
|                    | XMEDIA- <i>x</i> ( <i>y</i> )   | Outputs <i>y</i> as media data <i>x</i> : Media data type<br><i>y</i> : Data field, character constant (file name)   | E           | DD (*1)     | DD                  | E                               | E                                     | DD                  | DD           | Not required |
| String operation   | HEADSTR( <i>x</i> )   | Adds <i>x</i> before the field<br><i>x</i> : Data field, character constant  | E           | DD (*1)     | DD                  | E                               | E                                     | DD                  | DD           | Required     |
|                    | TAILSTR( <i>x</i> )   | Adds <i>x</i> after the field<br><i>x</i> : Data field, character constant   | E           | DD (*1)     | DD                  | E                               | E                                     | DD                  | DD           |              |
|                    | CONNECTSTR ( <i>x</i> 1, <i>x</i> 2[, <i>x</i> 3...])                               | Links <i>x</i><br><i>x</i> : Data field, character constant, half-width space "_SPACE_"  | E           | DD (*1)     | DD                  | E                               | E                                     | DD                  | DD           | Not required |
| Condition judgment | CASE( <i>x</i> 1, <i>y</i> 1) [CASE( <i>x</i> 2, <i>y</i> 2)..] DEFAULT( <i>z</i> ) | Outputs value <i>y</i> if condition <i>x</i> is TRUE<br>When no condition is met, outputs value <i>z</i> specified as DEFAULT<br><i>x</i> : Judged condition<br><i>y</i> , <i>z</i> : Data field, constant, omitted value "_NULL_" | E           | DD (*1)     | DD                  | E                               | E                                     | DD                  | DD           | Not required |

**Table 6.1** Types of operators (2 / 3)

| Type     | Operator            | Definition   | Form format |             |                     |                                 |  |                     |              | Input data   |
|----------|---------------------|--|-------------|-------------|---------------------|---------------------------------|--|---------------------|--------------|--------------|
|          |                     |  | Free frame  | Single page | Simple table format | Multipart table (Report header) | Multipart table (Other than report header) | Block repeat format | Sticky label |              |
| Function | ADD(x,y)            | $x + y$<br>x, y: Data field, numeric constant                              | E           | DD          | DD                  | E                               | E  | DD                  | DD           | Not required |
|          | SUB(x,y)            | $x - y$<br>x, y: Data field, numeric constant                              | E           | DD          | DD                  | E                               | E  | DD                  | DD           |              |
|          | MUL(x,y)            | $x * y$<br>x, y: Data field, numeric constant                              | E           | DD          | DD                  | E                               | E  | DD                  | DD           |              |
|          | DIV(x,y)            | Quotient of $x / y$<br>x, y: Data field, numeric constant                  | E           | DD          | DD                  | E                               | E  | DD                  | DD           |              |
|          | CAL(Equation)       | Four arithmetic expressions  | E           | DD          | DD                  | E                               | E  | DD                  | DD           |              |
|          | SET(x)              | Transfer of x<br>x: Data field, constant, variable "%"                     | E           | DD (*1)     | DD                  | E                               | E  | DD                  | DD           |              |
|          | SUM(x)              | Sum total of x<br>x: Data field  | E           | DD (*1)     | DD                  | -                               | E  | DD                  | DD           |              |
|          | TSUM(x)             | Sum total of x through all data (whole data)<br>x: Data field              | E           | DD (*1)     | DD                  | E                               | E  | DD                  | DD           |              |
|          | BSUM- <i>nn</i> (x) | Sum total of x at each BREAK<br><i>nn</i> : Control level<br>x: Data field | E           | -           | DD                  | -                               | E  | DD                  | -            |              |
|          | ALL(x)              | Sum total of x<br>x: Data field with repeats specified                     | -           | DD (*1)     | -                   | -                               | -  | -                   | -            |              |
|          | MAX(x)              | Maximum for x<br>x: Data field   | E           | DD (*1)     | DD                  | -                               | E  | DD                  | DD           |              |
|          | MIN(x)              | Minimum for x<br>x: Data field   | E           | DD (*1)     | DD                  | -                               | E  | DD                  | DD           |              |
|          | MEAN(x)             | Average for x<br>x: Data field   | E           | DD (*1)     | DD                  | -                               | E  | DD                  | DD           |              |
| CNT      | Number of processes | E  | DD (*1)     | DD          | -                   | E                               | DD   | DD                  |              |              |

**Table 6.1** Types of operators (3 / 3)

| Type              | Operator   | Definition  | Form format |             |                     |                                 |  |                     |              | Input data   |
|-------------------|------------|---|-------------|-------------|---------------------|---------------------------------|--|---------------------|--------------|--------------|
|                   |            |   | Free frame  | Single page | Simple table format | Multipart table (Report header) | Multipart table (Other than report header) | Block repeat format | Sticky label |              |
| Internal variable | PAGE       | Page number   | DD          | DD (*1)     | DD                  | DD                              | DD   | DD                  | DD           | Not required |
|                   | TPAGE      | Total number of pages   | DD          | DD (*1)     | DD                  | DD                              | DD   | DD                  | DD           |              |
|                   | PAGE-ALL   | Page number through all forms when performing multi-form output           | DD          | DD (*1)     | DD                  | DD                              | DD   | DD                  | DD           |              |
|                   | TPAGE-ALL  | Total number of pages through all forms when performing multi-form output | DD          | DD (*1)     | DD                  | DD                              | DD   | DD                  | DD           |              |
|                   | BPAGE-nn   | Page number per BREAK<br>nn: Control level                                | DD          | -           | DD                  | -                               | DD   | DD                  | -            |              |
|                   | BTPAGE-nn  | Total number of pages per BREAK<br>nn: Control level                      | DD          | -           | DD                  | -                               | DD   | DD                  | -            |              |
|                   | BODY       | Detail number in one page of a form                                       | -           | -           | DD                  | -                               | -  | DD                  | -            |              |
|                   | BODYT      | Sequence numbers for BODY (when a body is output over multiple pages)     | -           | -           | DD                  | -                               | -  | DD                  | -            |              |
|                   | SHARED     | Record area sharing   | -           | -           | -                   | DD                              | DD   | -                   | -            |              |
|                   | USERID     | User ID   | DD          | DD          | DD                  | DD                              | DD   | DD                  | DD           |              |
| Other             | - (Hyphen) | Input data cannot be taken  | DD          | DD (*1)     | DD                  | DD                              | DD   | DD                  | DD           |              |

E (Extended): Can be specified. (Specify the timing of calculation in the COMPUTE statement on the [Extended] tab.)

DD (Data Defs): Can be specified. (List the data field names on the [Data Definitions] tab in the order of calculation.)

-: Cannot be specified or specification invalid.

\*1: Cannot be specified for data fields where repeat is specified.

• **BREAK**

Check whether it is necessary to execute calculation, such as a total, or start a new page when data for a certain field changes. If necessary, determine which operations you want to perform when the input data values for certain fields change.

The form formats and areas in which a BREAK can be specified are shown below:

- Partitions in the form layout data in free frame format
- Partitions other than RH (Report Header) in the form layout data in multipart table format
- Head and tail partitions in the form layout data in simple table format
- Head and tail partitions in the form layout data in block repeat format

Only one BREAK can be specified in the form layout data in simple table format and block repeat format.

Since up to 16 BREAKs can be specified in the form layout data in multipart table format and free frame format, the priority (control level) of the BREAKs should be considered. For example, to calculate monthly subtotals and annual totals on the "Sales Spreadsheet", specify control levels of 2 for "Month" and 1 for "Year".



**Caution** - The fields specified with BREAKs are processed only up to their specified data lengths during form output. Take into consideration BREAKs when setting the data lengths of fields specified with the BREAKs.



**Note** - You can set to perform dittoed output according to priorities for dittoed output even if a BREAK occurs. For details, refer to the Designer Help guide.

- **Media data**

If an image field is defined in form layout data, determine the type of media data. The media data that can be used varies depending on the output method.



**Caution** - When you transfer a media data file from the application server to the List Creator print server for distributed output, be sure to specify the media data file name when outputting the form.



**Note** - When outputting a form in Windows, you can output OLE2 object if an application supporting OLE2 objects is installed on the List Creator print server. For example, you can embed a graph created by Microsoft Excel in a form.  
- If you always output the same bitmap data, you can also import the bitmap data in the form layout data in advance (overlay image).

For the support status of media data for each operating system running on the List Creator print server, refer to the Online Manual "Setup & Operation".

- **String operation**

Check whether it is necessary to edit input data values (strings). If necessary, determine which strings are added to the input data values (strings) for which fields.



**Caution** - When performing string operations, specify data lengths and field lengths with consideration for the addition of strings to the fields of which strings are output. If the data/field lengths are too short, the data will not be output correctly.

- **Condition judgment (CASE data output)**

Check the fields where output contents need to be changed depending on the conditions when outputting the form. When such fields exist, determine the method for judging input data and values you want to output.

- **Function**

Check whether the results calculated based on input data need to be output to the form. For example, check whether daily, monthly, and annual totals on the "Sales Spreadsheet" are calculated from individual sales records in input data, or calculated in advance and provided by input data.

- **Internal variable**

Check whether there are any fields in which you want to output internal information, such as page numbers and detail numbers.

### 6.2.3 Selecting the Output Order and Output Conditions of Frames and Partitions, and Timing of Calculations

For form layout data in free frame format, select the output order and output conditions for defined frames and partitions, and what calculations are performed for which data fields before outputting each frame and partition.

For form layout data in multipart table format, select the output order and output conditions for defined partitions, and what calculations are performed for which data fields before outputting each partition.

## 6.3 Defining Association Data


Define settings selected in "6.2 Selecting Information to be Defined for Association Data". Use the [Data Definitions] window to define the settings.

### = Procedure =

- 1) In the [Form Style Descriptor] window, select [File] - [Data Definitions].  
=>The [Data Definitions] window is displayed.



**Note**

- When the [Form Style Descriptor] window is closed, select the form descriptor information defining association data in the [Form Descriptor List] window, and click the [Open] button  on the toolbar. The [Form Style Descriptor] window is displayed. You can also select [File] - [Open], or double click the name of the form to display the [Form Style Descriptor] window.

- 2) Define the settings in the [Data Definitions] window. Definitions will vary according to form formats. Check the displayed tabs in the following table, and define the settings by referring to the explanations for the tabs that appear below the table.  
=> "Table 6.2 Form format and the tabs in the [Data Definitions] window"

- 3) Once defining is complete, click the [OK] button in the [Data Definitions] window to save the definitions.  
=>The display returns to the [Form Style Descriptor] window.
- 4) In the [Form Style Descriptor] window, select [File] - [Close]. A message asking if you want to save the file is displayed. Click [Yes].  
=>The definitions are saved, and the display returns to the [Form Descriptor List] window.

The tabs displayed in the [Data Definitions] window for each form format are shown below. Refer to the descriptions of each tab according to form format you are using. For details of each tab, refer to the Designer Help guide.

**Table 6.2** Form format and the tabs in the [Data Definitions] window

| Form format \ Tab      |   | General | Data Definitions (Report Header) | Data Definitions | Extended | Database Link |
|------------------------|---|---------|----------------------------------|------------------|----------|---------------|
| Free Frame format      | When frames are defined                           | Y       | -                                | Y                | Y        | Y             |
|                        | When no frame is defined                          | Y       | -                                | Y                | -        | Y             |
| Single page format     |   | Y       | -                                | Y                | -        | Y             |
| Simple table format    |   | Y       | -                                | Y                | -        | Y             |
| Multipart table format | When the report header (RH) partition is used     | Y       | Y                                | Y                | Y        | Y             |
|                        | When the report header (RH) partition is not used | Y       | -                                | Y                | Y        | Y             |
| Block repeat format    |   | Y       | -                                | Y                | -        | Y             |
| Sticky label format    |   | Y       | -                                | Y                | -        | Y             |

Y: Displayed  
 -: Not displayed



**Caution**

- In the [Data Definitions] window, field names are used even if the input data format is "XML Format" (tag names cannot be used).
- The [Data Definitions] window is not displayed unless one or more of the following types of fields are defined.  
 For form layout data that includes the report header (RH) partition in multipart table format, this window is not displayed unless one or more of the following types are specified in each of the RH partition and other partitions.
  - Numeric field
  - Text field
  - Block text field
  - OCR-B field
  - Radio button
  - Check box
  - Barcode field (not including barcode (U.S. Postal FIM) fields)
  - Image field
- Whenever adding, renaming, cutting, or deleting a field in the [Form Style Descriptor] window, be sure to check its association data in the [Data Definitions] window. If an operand contains a field name that does not exist, an error will occur when saving the form layout data.

# **Chapter 7**

## *Confirming the Output Image*

---

This chapter explains how to confirm the output image of form layout data.

The following three methods are available for checking the output images of form layout data:

- Layout print
- Preview
- Test print

The following describes each method.

## 7.1 Layout Print

Using the layout print function, the form layout data can be checked even while the form is being designed. When performing the layout print, the form layout data is printed with scales. To check the actual form output image, perform a test print.

**= Procedure =**

- 1) In the [Form Style Descriptor] window, select [File] - [Layout print].  
=>The print dialog box is displayed.
- 2) Click the [OK] button.  
=>The form layout is printed with scales.

**Note** - The layout print is performed according to the paper size and orientation specified in the printer settings window, which is displayed by selecting [File] - [Print Setup]. Therefore, if the paper size specified in the printer settings is smaller than that specified for a form, the form will be reduced during output.

| Product Name         | Amount                |
|----------------------|-----------------------|
| MMMMMMMMMMMMMMMMMMMM | \$\$, \$\$\$, \$\$9   |
| MMMMMMMMMMMMMMMMMMMM | \$\$, \$\$\$, \$\$9   |
| MMMMMMMMMMMMMMMMMMMM | \$\$, \$\$\$, \$\$9   |
| MMMMMMMMMMMMMMMMMMMM | \$\$, \$\$\$, \$\$9   |
| MMMMMMMMMMMMMMMMMMMM | \$\$, \$\$\$, \$\$9   |
| Total                | \$\$\$, \$\$\$, \$\$9 |

**Figure 7.1** Output example of layout print



## 7.2 Preview

Using the preview function, the output image of form layout data can be checked even while the form is being designed.

The following methods are available for previewing:

- Previewing when designing a form

It is useful when checking the output image of the form being designed. You can use both the actual input data and an input data format to preview.

- Previewing with Utility (Preview)

It is useful when checking the form output image using the created form style information and the actual input data.



**Note**

- You cannot preview when designing a form, in the following cases: Preview using Utility (Preview) or with the preview command.
  - When using a data file for multi-form output
  - When using a data file with field name label

Preview is a function for checking the form output image. Therefore, fields are displayed in different positions and sizes from the output results of the actual form. To check the actual form output image, perform a test print.

## 7.2.1 Previewing when Designing a Form

This section explains how to perform a preview when designing a form.

The following two methods are available for previewing when designing a form:

- Previewing from the [File] menu of the [Form Style Descriptor] window
- Previewing from the [Form Descriptor List] window

This section explains how to perform a preview from the [File] menu of the [Form Style Descriptor] window, with an example in which "Print" is specified for the output method.



### Note

- For notes during preview, refer to "4.4.1 preview Command" of the Online Manual "Setup & Operation".
- When previewing from the [Form Descriptor List] window, select the form descriptor information, and select [File] - [Preview] to display the [Preview Start] window. You can specify output printer name when previewing from the [Form Descriptor List] window. Other operations are same as the procedure of previewing from the [File] menu of the [Form Style Descriptor] window.

### = Procedure =

- 1) In the [Form Style Descriptor] window, select [File] - [Preview].  
=>The [Preview Start] window is displayed.

- 2) Select "Preview Type". For this instance, select "Print".
- 3) Select "Code Type".




### Caution

- This is valid only when the input data format is "Variable Length", "Fixed length", or "CSV Format". When the format of the input data is "XML format", the character code is specified in XML declaration in the XML file. If the XML declaration in the XML file is omitted, or if the character code is not specified in the XML declaration, it is regarded as UNICODE (UTF8).

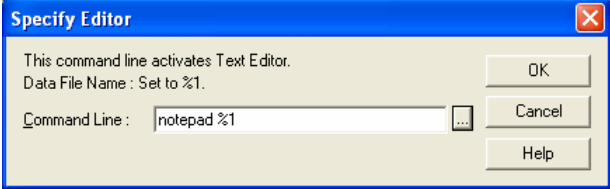
4) Specify the "Data File Name" to be created as a format.

You can also use the default file name.



**Note**


- The actual input data can be specified. Click the [...] button to the right of the input area, and specify the input data in the window that is displayed.
- The specified format data file can be edited by clicking the [Edit] button to start a text editor. Edit it as necessary.  
To change the text editor you are using, click the [Editor] button. When the Editor Specification window is displayed, specify the command line to start the text editor.



- Immediately after installing the product, "notepad %1" is specified.
- For the command line of the text editor, specify the full path to the command or a command name that resides under the directory specified for the environment variable PATH.
- "%1" is replaced with the format data file when the command line is executed. Modify the command line as needed in accordance with the command syntax of the specified text editor.
- If "XML Format" is specified for the input data format, and the format data file is created with UNICODE (UTF8), specify the text editor that supports the UNICODE (UTF8) character code.

- When using the following functions, you cannot use the actual input data to preview. To perform a preview, a format data file should be created.
  - XBRL data support function
  - Navigator links function

5) Select a type in the "Format Data Type". For this instance, select "Record Length Confirmation".




**Note**

- For the forms for which BREAK or dittoed output is specified, "BREAK Confirmation" should be selected.

6) Click the [Format Creation] button.

=>A format data file is created.

If you select "Record Length Confirmation" in the "Format Data Type", dummy data such as shown below is used for each field type in the format data file. For single page and free frame format without defined frames, a one page input data format is created. For all other formats, a single detail input data format is created.



**Note**

- If "XML Format" is specified for the input data format, format data files are created with UNICODE (UTF8).

| Field type                | Record Field Type | Formatting | Output format of dummy data |
|---------------------------|-------------------|------------|-----------------------------|
| Numeric field             | Numeric           | Value      | "9999...." (*1)(*2)         |
|                           |                   | Currency   | "9999...." (*1)(*2)         |
|                           |                   | Separator  | "9999...." (*1)             |
|                           |                   | Date       | Current date                |
| Text field                | Text              | Standard   | "XXXX...." (*1)             |
|                           |                   | Mail       | "9999...." (*1)             |
| Block text field          | Text              | -          | "XXXX...." (*1)             |
| OCR-B field               | Text              | -          | "XXXX...." (*1)             |
| Radio Button<br>Check box | Text              | -          | (Half-width space) (*1)     |
| Barcode field             | Text              | -          | "9999...." (*1)(*5)         |

| Field type            | Record Field Type | Formatting | Output format of dummy data  |
|-----------------------|-------------------|------------|--|
| Image field           | Text              | -          | prtest.bmp   |
| Post code field       | -                 | -          | "9999..." (*1)   |
| Hidden field (Record) | Text              | -          | "XXXX..." (*1)   |
|                       | Numeric           | -          | "9999..." (*1)(*2)   |
| Group item            | -                 | -          | If "Use Group Item Name" is specified for [Group Item Handling] in the [Data Definitions] window, a single dummy data is created from the data of the configuration fields linked with a fixed length format without separator characters, regardless of the input data format. (*3)<br>If "Use Configuration Item Name" is specified for [Group Item Handling], dummy data with separator characters is created from the data of the configuration fields according to the input data format and separator characters. (*4) |


\*1: Characters of the data length are output.


\*2: If the decimal place is specified, the decimal point is also output.

\*3: Data is created as shown below:  
XXXXXX999999XX9999

\*4: Data is created as shown below:  
XXXXXX,999999,XX,9999


\*5: For Code 3 of 9 (EIAJ Compliant), data is created as shown below:  
Z9999....  
For EAN-128 (Convenience Store), data is created as shown below:  
91999....


 **Caution** - If the record type of the barcode field add-on data is set to "Text", change the corresponding data in the generated format data file to a numeric value.

 **Note** - Edit the format data file for the following cases:


- Checking BREAK results
- Checking calculation results
- Outputting multiple details in forms such as free frame format, multipart table format, or simple table format  
Increase the number of lines in the format data file when the input data format is "Variable Length", "Fixed Length", or "CSV Format".  
Increase the number of elements in the body area group item or details group item when the input data format is "XML Format".
- Checking output partitions according to the specifications on the [Extended] tab of the [Data Definitions] window, with a multipart table format or free frame format  
For details, refer to the following:  
[=> "8.2 Designing the Input Data based on Input Data Reports"](#)

7) Check the "Confirm whether data exceeds field area" checkbox to see if data exceeds the field area. When the checkbox is checked, the entire field is replaced with a series of the substitution character "#" in the preview if the data exceeds the field area.

 **Caution** - When an output image is displayed in the Excel file format, this setting is disabled.

 **Note** - When the [Confirm whether data exceeds field area] checkbox is checked, the substitution character "#" is used in the preview, regardless of the [Not Output an Overflowing Portion] setting in the [Alignment] tab of the field's properties window.



**Caution**  - The [Preview Start] window is not displayed unless 1 or more of the following types of fields are defined.  
For form layout data that includes the report header (RH) partition in multipart table format, this window is not displayed unless one or more of the following types are specified in each of the RH partition and other partitions.

- Numeric field
- Text field
- Block text field
- OCR-B field
- Radio button
- Check box
- Barcode field (not including barcode (U.S. Postal FIM) fields)
- Image field

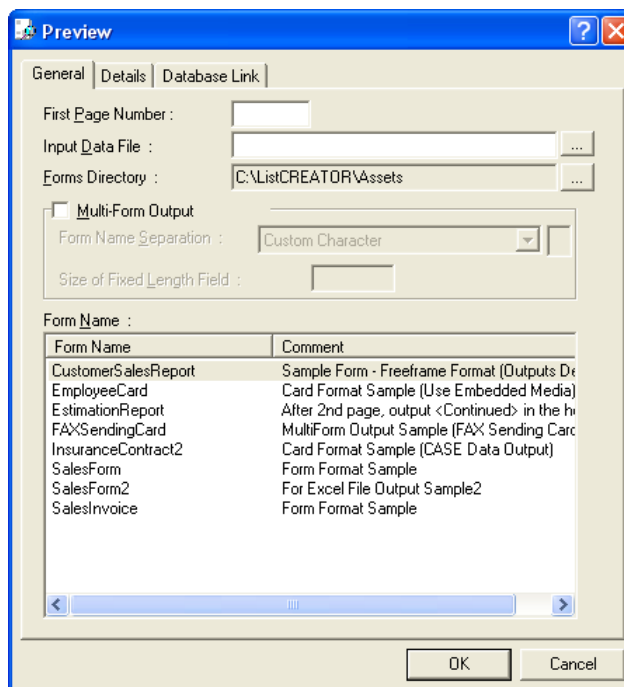
- The format data file created here will not be deleted even if the [Preview Start] window is closed.  
Delete the format data file when it is no longer required.


## 7.2.2 Previewing with Utility (Preview)

This section explains how to preview using the utility (Preview).

### = Procedure =

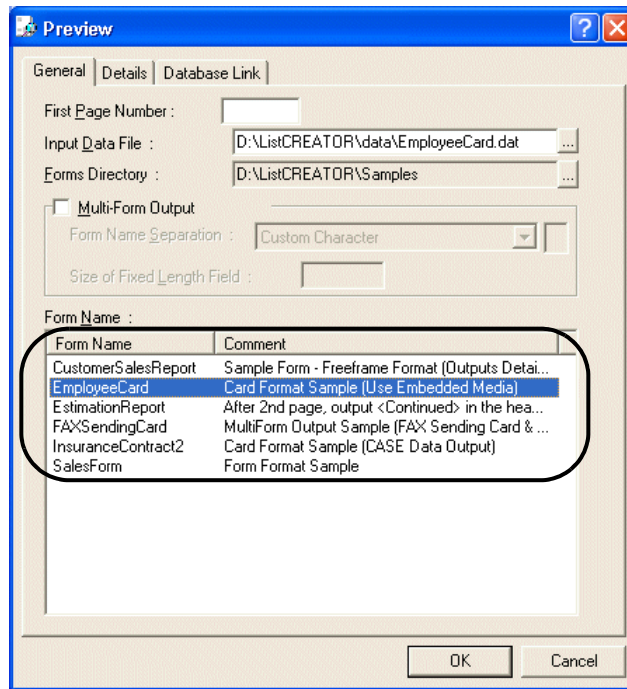
- 1) Select [Start] - [Programs] - [List Creator] - [Utility] - [Preview].  
=>The initial [Preview] window is displayed.



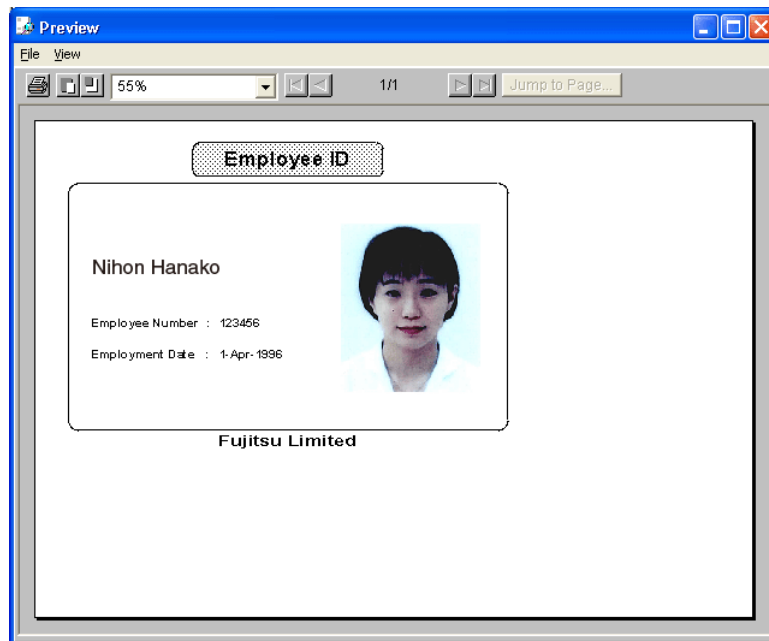
- Note**  - You can also perform a preview with the prview command. For operations and notes regarding the prview command, refer to the Online Manual "Setup & Operation".  
- For details regarding the [Preview] window, refer to the Online Help guide.

- 2) Specify the name of the created data file under [Data File Name] on the [General] tab.
- 3) Click the button to the right of the "Forms Directory" on the [General] tab, and switch the form storage directory in the window that is displayed.  
When browsing a directory, specify it within 259 bytes.

4) Select the form to be previewed from the list of form names on the [General] tab.




5) Click the [OK] button.  
=>The [Preview] window is displayed.



## 7.3 Test Print


The form output image can be checked by overlapping dummy data over the form style information created during form designing.

Since fields are printed in the specified position and size, the actual form output image can be verified.

 **Caution** - Perform a test print to check the form output image with the operating system and output device that are actually used for the form output.

**= Procedure =**


- 1) Select [Start] - [Programs] - [List Creator] - [Utility] - [Test Print].  
=>The [Test Print] window is displayed.

 **Note** - You can also perform a test print with the prtest command. For operations and notes regarding the prtest command, refer to the Online Manual "Setup & Operation".

- 2) Specify the printer name, and select the form style information to be printed.
- 3) Click the [Print] button.  
=>The data is printed by the specified printer.

| Sales Form                   |                 |
|------------------------------|-----------------|
| To<br>XXXXXXXXXXXXXXXXXXXX   |                 |
| Product Name                 | Amount          |
| XXXXXXXXXXXXXXXXXXXXXXXXXXXX | \$9 ,999 ,999   |
| XXXXXXXXXXXXXXXXXXXXXXXXXXXX | \$9 ,999 ,999   |
| XXXXXXXXXXXXXXXXXXXXXXXXXXXX | \$9 ,999 ,999   |
| XXXXXXXXXXXXXXXXXXXXXXXXXXXX | \$9 ,999 ,999   |
| XXXXXXXXXXXXXXXXXXXXXXXXXXXX | \$9 ,999 ,999   |
| Total                        | \$999 ,999 ,999 |

**Figure 7.3** Output example of test print

 **Note** - When printing a free frame format or multipart table format, each partition will be printed once.



# **Chapter 8**

## *Designing Input Data*

---

This chapter explains the methods for designing input data for the form style information created in the form designing process.

If the input data has already been determined, check that the data's format is appropriate for the form style information used.

## 8.1 Checking Input Data Reports

Input data reports mean the information about the input data in the form style information created.

Input data reports can be printed, or output to PDF file or Excel file by using printing reports function.

The input data reports provide the information such as the configuration, format, and length of data, which is required for the input data to be overlapped over form style information. Input data is designed based on the input data reports.

Output the input data report before designing the input data.

The procedure for outputting input data reports is shown below.

The following shows an example in which "Print" is specified for the output method.

### = Procedure =

- 1) Select [Start] - [Programs] - [List Creator] - [Form Descriptor List].  
=>The [Form Descriptor List] window is displayed.
- 2) Select the form descriptor information which you want to print the input data report of, and select [File] - [Form Contents Report].  
=>The [Form Contents Report] window is displayed.

- 3) Select the output method. Here, select "Print".
- 4) Specify the name of the printer to use for output.



#### Note

- "Printer Name" can be specified only when "Print" is selected.
- When you select "PDF File" or "Excel File", specify "Output Destination Directory".

- 5) Select [Output] for the [Input Data Report], and select [Do not Output] for the other output details.
- 6) Click the [OK] button.  
=>The input data report is printed by the specified printer.



- The input data report is printed on a cut-sheet of A4 paper according to the properties of the specified printer driver. Place A4 paper in the printer, then set the properties of the printer driver to print with A4 paper before printing.

Input Data Report

03 Mar 2005

| SalesForm            |                       | N                   |                  |             |                                   |             |           | Page           |               |               |
|----------------------|-----------------------|---------------------|------------------|-------------|-----------------------------------|-------------|-----------|----------------|---------------|---------------|
|                      |                       | B                   |                  |             |                                   |             |           | 1              |               |               |
| Input Data Format    | Variable Length       | Separator Character | Vertical Bar ( ) |             | Barcode Function Character (FNC1) | -           |           |                |               |               |
| <<Data Definitions>> |                       |                     |                  |             |                                   |             |           |                |               |               |
| #                    | Field Name            | Field Type          | Field Format     | Format Type | Field Info                        | Data Length | Byte Size | Decimal Places | Total Repeats | Control Level |
| 1                    | CustomerPostalCode    | Text                | Text             | Mail        | -                                 | 7           | -         | -              | -             | -             |
| 2                    | CustomerAddress1      | Text                | Text             | Standard    | -                                 | 30          | -         | -              | -             | -             |
| 3                    | CustomerAddress2      | Text                | Text             | Standard    | -                                 | 30          | -         | -              | -             | -             |
| 4                    | CustomerName          | Text                | Text             | Standard    | -                                 | 20          | -         | -              | -             | -             |
| 5                    | SalesDate             | Numeric             | Numeric          | Date        | -                                 | 8           | -         | 0              | -             | -             |
| 6                    | SalesRecordNumber     | Text                | Text             | Standard    | -                                 | 5           | -         | -              | -             | 1             |
| 7                    | PersonResponsibleCode | Numeric             | Numeric          | Value       | -                                 | 3           | -         | 0              | -             | -             |
| 8                    | ProductCode           | Numeric             | Numeric          | Value       | -                                 | 4           | -         | 0              | -             | -             |
| 9                    | ProductName           | Text                | Text             | Standard    | -                                 | 30          | -         | -              | -             | -             |
| 10                   | Quantity              | Numeric             | Numeric          | Value       | -                                 | 8           | -         | 0              | -             | -             |
| 11                   | PricePerUnit          | Numeric             | Numeric          | Currency    | -                                 | 7           | -         | 0              | -             | -             |
| 12                   | Barcode               | Barcode             | Text             | -           | Code 3 of 9                       | 10          | -         | -              | -             | -             |

Figure 8.1 Example of printed input data report

## 8.2 Designing the Input Data based on Input Data Reports

Design the input data based on the printed input data reports.

When the input data format is "XML Format", create the input data according to the data configuration determined.

Use input data reports only when you want to know the order of the fields defined on the [Data Definitions] window.

The method for understanding input data reports is explained in the following order:

- Understanding Data Configuration
- Understanding the Data Format for Each Field
- Understanding Data Length of Each Field
- Sorting the Data

The explanation corresponds with the numbers in "Input Data Report" below.

| Input Data Report |                       |                 |        |                     |              |                |             |                                   |           | 03 Mar 2005          |               |               |     |      |      |      |      |      |
|-------------------|-----------------------|-----------------|--------|---------------------|--------------|----------------|-------------|-----------------------------------|-----------|----------------------|---------------|---------------|-----|------|------|------|------|------|
| SalesForm<br>(1)  |                       |                 | N<br>B | (2)                 |              |                |             |                                   |           | Page                 |               |               |     |      |      |      |      |      |
| Input Data Format |                       | Variable Length |        | Separator Character |              | Vertical Bar() |             | Barcode Function Character (FNC1) |           | -                    |               |               |     |      |      |      |      |      |
| (3)               |                       |                 |        |                     |              |                |             |                                   |           | <<Data Definitions>> |               |               |     |      |      |      |      |      |
|                   |                       |                 |        |                     |              |                |             |                                   |           | (6)                  | (7)           | (8)           | (9) | (10) | (11) | (12) | (13) | (14) |
| (4) #             | Field Name (5)        |                 |        | Field Type          | Field Format | Format Type    | Field Info  | Data Length                       | Byte Size | Decimal Places       | Total Repeats | Control Level |     |      |      |      |      |      |
| 1                 | CustomerPostalCode    |                 |        | Text                | Text         | Mail           | -           | 7                                 | -         | -                    | -             | -             |     |      |      |      |      |      |
| 2                 | CustomerAddress1      |                 |        | Text                | Text         | Standard       | -           | 30                                | -         | -                    | -             | -             |     |      |      |      |      |      |
| 3                 | CustomerAddress2      |                 |        | Text                | Text         | Standard       | -           | 30                                | -         | -                    | -             | -             |     |      |      |      |      |      |
| 4                 | CustomerName          |                 |        | Text                | Text         | Standard       | -           | 20                                | -         | -                    | -             | -             |     |      |      |      |      |      |
| 5                 | SalesDate             |                 |        | Numeric             | Numeric      | Date           | -           | 8                                 | -         | 0                    | -             | -             |     |      |      |      |      |      |
| 6                 | SalesRecordNumber     |                 |        | Text                | Text         | Standard       | -           | 5                                 | -         | -                    | -             | 1             |     |      |      |      |      |      |
| 7                 | PersonResponsibleCode |                 |        | Numeric             | Numeric      | Value          | -           | 3                                 | -         | 0                    | -             | -             |     |      |      |      |      |      |
| 8                 | ProductCode           |                 |        | Numeric             | Numeric      | Value          | -           | 4                                 | -         | 0                    | -             | -             |     |      |      |      |      |      |
| 9                 | ProductName           |                 |        | Text                | Text         | Standard       | -           | 30                                | -         | -                    | -             | -             |     |      |      |      |      |      |
| 10                | Quantity              |                 |        | Numeric             | Numeric      | Value          | -           | 8                                 | -         | 0                    | -             | -             |     |      |      |      |      |      |
| 11                | PricePerUnit          |                 |        | Numeric             | Numeric      | Currency       | -           | 7                                 | -         | 0                    | -             | -             |     |      |      |      |      |      |
| 12                | Barcode               |                 |        | Barcode             | Text         | -              | Code 3 of 9 | 10                                | -         | -                    | -             | -             |     |      |      |      |      |      |

For (1) through (4) and (13), refer to the following:

=> ["8.2.1 Understanding the Data Configuration"](#)

For (5) through (9), refer to the following:

=> ["8.2.2 Understanding the Data Format for each Field"](#)

For (10) through (12), refer to the following:

=> ["8.2.3 Understanding the Data Length of each Field"](#)

For (14), refer to the following:

=> ["8.2.4 Sorting the Data"](#)

### 8.2.1 Understanding the Data Configuration

This section explains how to understand the data configuration.

Input data is entered in units of records (lines).

For the character code that can be handled in input data, refer to the following:

=> ["Appendix B Cautions for Designing Forms"](#)

### 8.2.1.1 How a Record of Input Data is Handled for each Form Format

How a record (line) of input data is handled for each form format is shown below.

For input data in the free frame format, refer to the following:

=> "8.2.1.2 How Input Data in Free Frame Format is Handled"

**Table 8.1** How a record (line) of input data is handled

| Form format            |   | 1 record (line) unit  | Remarks  |
|------------------------|---|-----------------------|--|
| Free Frame format      | When no frame is defined  | 1 page                | - All records (lines) require data for headers, bodies, and footers.   |
|                        | When one frame is defined   | 1 detail              | - All records (lines) require data for headers, bodies, and footers, and data for control header partitions, details partitions, and control footer partitions (CH, DE, CF).   |
|                        | When the multiple frames are defined and the all frames are linked in one |                       | - The data for headers, bodies, and footers should be the same until a new page starts.<br>- The data for control header and control footer partitions (CH, CF) should be the same until a BREAK occurs.   |
|                        | When the multiple frames are defined and the link is not set              | 1 output frame detail | - All records (lines) require data for headers, bodies, and footers, and data for frame partitions (CH, DE, CF) of a frame which will be output.   |
|                        | When multiple frames are defined and multiple links are set               |                       | - The data for headers, bodies, and footers should be the same until a new page starts.<br>- The data for control header and control footer partitions (CH, CF) of a frame which will be output should be the same until a BREAK occurs.<br>- Data is not required for frame partitions (CH, DE, CF) of a frame which will not be output.<br>- You should consider the number of the details allowed to be output in a page of each frame when you create the data.  |
| Single page format     |   | 1 page                | -  |
| Simple table format    |   | 1 detail              | - All records (lines) require data for heads, bodies, and tails.<br>- The data for heads and tails should be the same until a BREAK occurs.  |
| Multipart table format |   | 1 detail              | - If data for report header partitions (RH) is required, specify it in the first record (line).<br>- All records (lines) except those for report header partitions (RH) require data for page header, control header, details, control footer, and page footer partitions (PH, CH, DE, CF, PF).<br>- The data for page header and page footer partitions (PH, PF) should be the same until a new page starts.<br>- The data for control header and control footer partitions (CH, CF) should be the same until a BREAK occurs. |
| Block repeat format    |   | 1 block               | - All records (lines) require data for heads, bodies, and tails.<br>- The data for heads and tails should be the same until a BREAK occurs.  |
| Sticky label format    |   | 1 label               | -  |

**Caution**

- Specify input data within 32KB per record (line).
  - When using data files, specify either 0x0D0A or 0x0A as the linefeed code.
  - The upper limit value of input data for each operating system running on the application server/List Creator print server is as follows:
    - For Windows: 4GB
    - For Solaris: 8GB
    - For HP-UX: Less than 2GB
    - For Linux: Less than 2GB
- If the upper limit value of input data is exceeded during the form output, an error may occur.

The following provides explanations for each data of input data reports.

Fields are not output in input data reports if the fields do not require input data.

## • When using a data file

### (1) Input data format

This displays the format of the data to be created. Input data formats are as follows:

- Variable length
- Fixed length
- CSV format
- XML format

### (2) Separator character

This indicates the characters used for separating the data of each field. The character(s) specified as a separator character must not be used in the data. If no separator character is specified, separator characters are not required between data of each field. If the input data is "CSV Format", characters used as separator characters are handled as data when enclosed in "double quotation marks (")".

### (3) Data definitions

This indicates the units for creating one record (line).

#### Data definitions (report header)

This is created for forms where the report header partition (RH) is defined with a multipart table format.

Data for the report header partition (RH) is created in the first record (line).

#### Data definitions

This is for creating data other than report header partitions (RH).

If [Data Definitions (Report Header)] is specified, data is created from the second record (line); if not specified, data is created from the first record (line).

### (4) # (Number)

This indicates the order of the data of one record (line).

Data is created in numeric order for each record (line).

The numbers correspond to the data fields receiving the input data, among those fields defined in the [Data Definitions (Report Header)] tab and the [Data Definitions] tab of the [Data Definitions] window.

The following description examples show data files that use "commas (,)" as separator characters.

Data\_for\_#1,Data\_for\_#2,Data\_for\_#3, ...

**Caution**

- If the order of the data in a record (line) for input data is changed from the one shown in the input data report, be sure to change the order of the data field names in the [Data Definitions (Report Header)] tab and the [Data Definitions] tab on the [Data Definitions] window.

### (13) Number of Repeats

When repeats are specified for fields in a single page form, data for the number of repeats is required. The number of data for that field is indicated by the number of repeats.

If repeat is not specified, "-" (hyphen) is output. In such a case, the number of data for that field is 1.

The following is an example of a data file where the separator character is a "comma (,)", the number of repeats for the first and second fields are "-" (hyphen) and "3", respectively.

Data\_for\_#1,Data1\_for\_#2,Data2\_for\_#2,Data3\_for\_#2, ...

### 8.2.1.2 How Input Data in Free Frame Format is Handled

For a form in the free frame format, how a record of input data is handled differs depending on whether or not a frame is defined and the defined frame is linked.

The following explains an example of the output image of a free frame form and how a record of input data is handled.

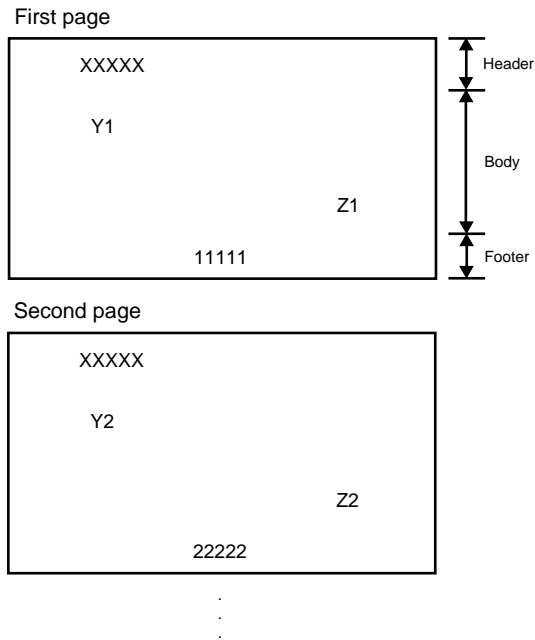
- When no frame is defined

Assume that you create form layout data with the fields defined as follows:

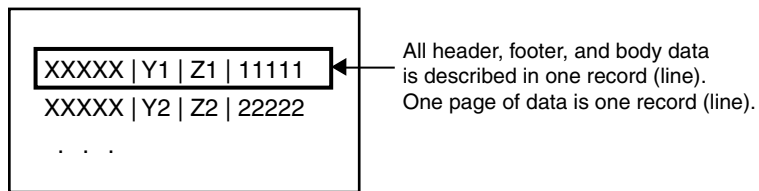
- One field is defined in the header
- Two fields are defined in the body
- One field is defined in the footer

An example of the output image of this form and how the input data is handled is shown below.

An output image example of the form:



An input data example:



- When a frame is defined or multiple frames are linked in one

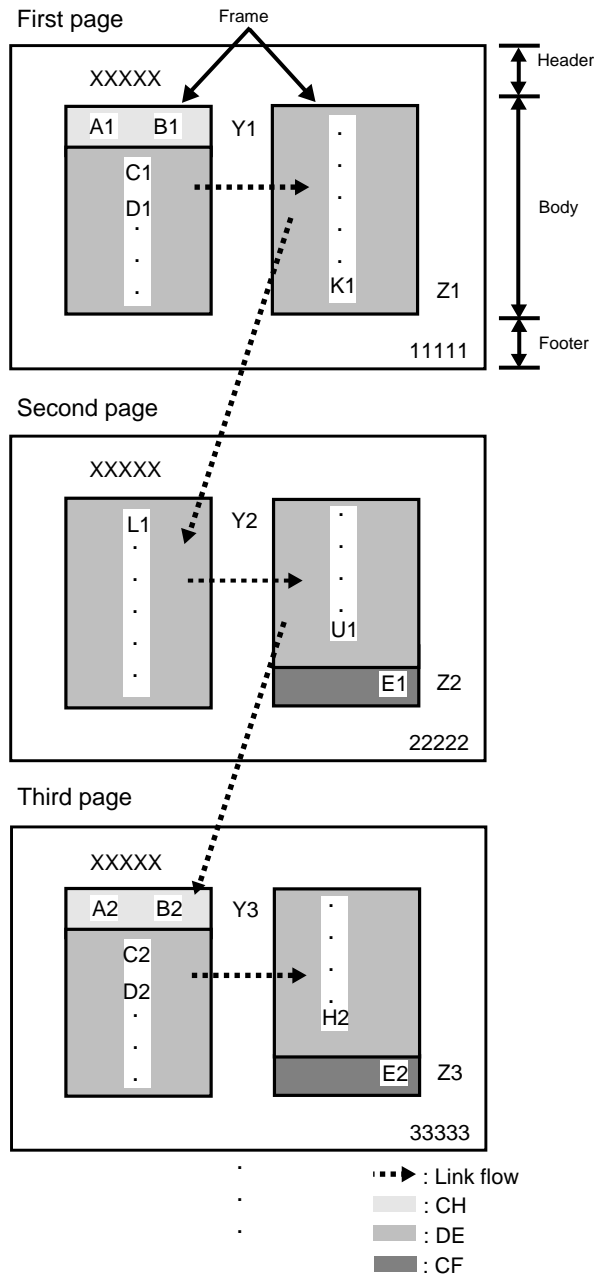
Assume that you create form layout data with the fields defined as follows:

- One field is defined in the header
- Two fields are defined in the body
- One field is defined in the footer
- Two frames are defined and linked
- Two fields are defined in the CH (Control Header) partition
- One field is defined in the DE (Details) partition
- One field is defined in the CF (Control Footer) partition



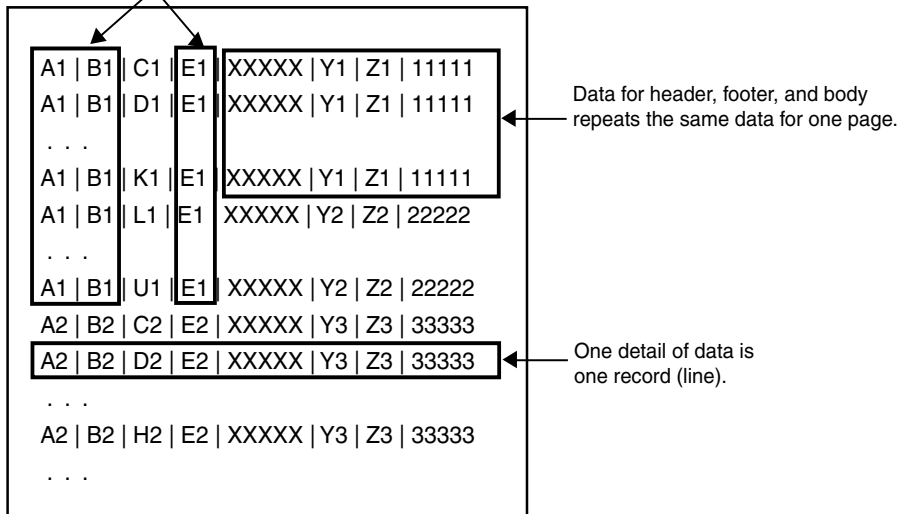
An example of the output image of this form and how the input data is handled is shown below.

An output image example of the form:



An input data example:

Data for CH and CF is repeated until BREAK occurs.



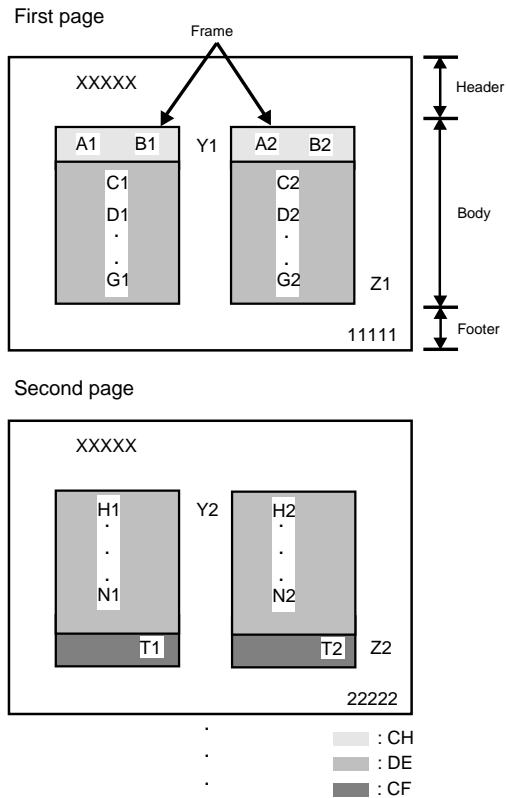
• When a frame is not linked or when multiple links are set

Assume that you create form layout data with the fields defined as follows:

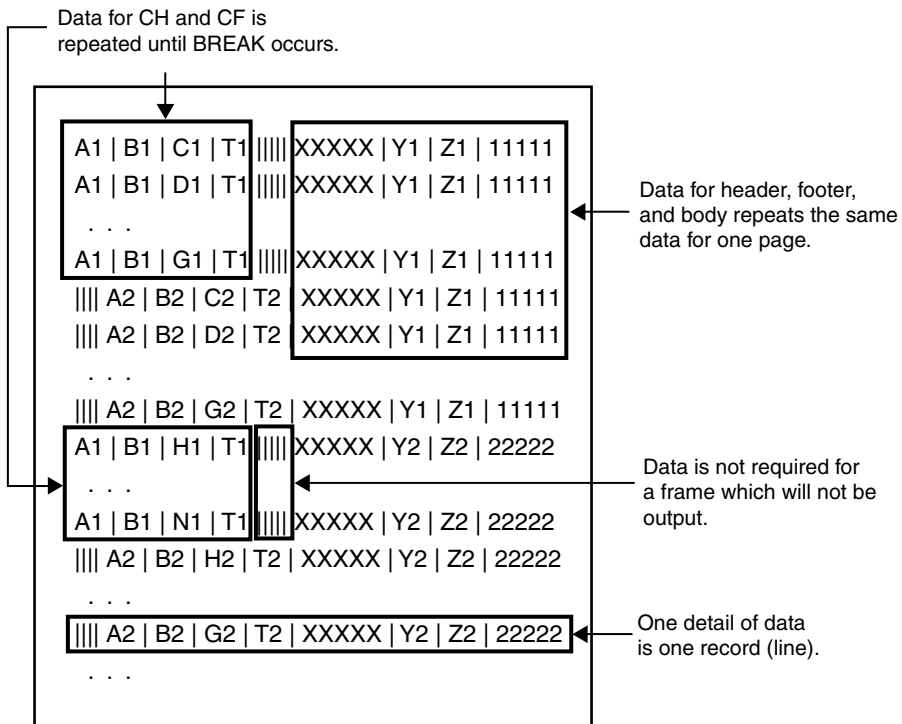
- One field is defined in the header
- Two fields are defined in the body
- One field is defined in the footer
- Two frames are defined but not linked
- Two fields are defined in the CH (Control Header) partition for each frame
- One field is defined in the DE (Details) partition for each frame
- One field is defined in the CF (Control Footer) partition for each frame

An example of the output image of this form and how the input data is handled is shown below.

An output image example of the form:



An input data example:



## 8.2.2 Understanding the Data Format for each Field

The data format for each field varies depending on the field type and its subtypes. When using a data file, you should create data in a format appropriate for the field type and its subtypes.

### (5) Field name

Field names that are output in the input data report require input data. Of the data fields in the [Data Definitions (Report Header)] tab and the [Data Definitions] tab of the [Data Definitions] window, if an operator requiring no input data is specified as an operand in a field, the field will not be output in the input data report.

For operators requiring no input data, refer to the Designer Help guide.



**Note**

- It is recommended that when form layout data is created, a field is named so that the field name represents the content of the field. It would be helpful when you create the data.

### (6) Field type

Data formats vary according to the field types.

#### Numeric

This field is used for outputting numeric data.

Data to be created will vary depending on the formatting types.

#### Text

This field is used for outputting string data in one line.

Data to be created will vary depending on the formatting types.

#### Block text

This field is used for outputting string data in multiple lines.

Data is expressed as string data.

#### OCR-B

This field is used for outputting alphanumeric data using an OCR-B font.

Data is expressed as alphanumeric data.

#### Radio button

This field is used for making a single selection from the choices available.

Data contains strings for the values for selected and unselected states.

#### Check box

This field is used for toggling an option or for selecting one or more options, by "Checking/Unchecking" it.

Data contains strings for the values for selected and unselected states.

#### Barcode

This field is used for outputting a barcode.

Data to be created will vary depending on the barcode type.

#### Image

This field is used for outputting media data such as a bitmap image.

Data contains the media data file name. The directory name is specified when outputting the form.

#### Post code

This field is used for outputting a post code to a fixed location of a postcard and an envelope.

Data is expressed as a 7-digit post code (not including hyphens (-)).

#### Hidden

This work field is not output to the form.

Data to be created will vary depending on the field format (record format) ("Numeric" or "Text").

Group

This is a group of multiple fields that are handled as one field.

If "Use Group Item Name" is specified for the group item handling, all fields that make up the group are specified as one data.

(7) Field format

This indicates the record format of the fields defined in the form.

Numeric

This field handles numeric data.

Text

This field handles string data.

Group

This is a group of multiple fields that are handled as one data unit.



- The field format of a hidden field (record) can be changed in the [Hidden Field Information] tab of the field properties.

(8) Formatting type

When the field type is "Numeric" or "Text", this indicates the data type in further detail.

When the field type is "Numeric"

Value

Numerals are specified in the data.

Currency

Numerals are specified in the data.

Separator

Numerals are specified in the data.

Date

Specify the data in YYYYMMDD date format (8-digit).

YYYY : Specify the western calendar year.

MM : Specify within the range of 01 to 12 (month).

DD : Specify within the range of 01 to 31 (day).

When the field type is "Text"

Standard

Data is expressed as string data.

Mail

Data is expressed as a 7-digit post code (not including hyphens (-)).

(9) Field detail

When the field type is "Barcode"

Barcode type is indicated. The value that can be used for data varies depending on the type of the barcode field.

Specify the "FNC1 Function Character" code for the function character "FNC1", which is used when the barcode type is "EAN-128".

When the field type is "Radio Button" or "Check Box"

This indicates the string used as the value for selected state when a radio button or checkbox is selected.

**Table 8.2** Field Type and Data Format (1 / 5)

| Field type                         | Type<br>Barcode type<br>Value for selected state | Input data required (*1) | Available characters (*2)                                     | Remarks   |
|------------------------------------|--|--------------------------|---|---|
| Static field<br>Block static field |  | -                        | -   |   |
| Numeric field (*3)                 | Value, currency, separator                       | Y                        | ¥ + - , .<br>0 to 9   | <ul style="list-style-type: none"> <li>- \$ (Currency symbols)<br/>Only one character can be specified at the beginning.<br/>Actually they are skipped.<br/>Even if currency symbols are specified when creating a form, only "\$" can be specified for the input data.</li> <li>- + - (Signs)<br/>Only one character can be specified at the beginning.<br/>When specifying currency symbols, specify a sign after the currency symbols.<br/>Even if minus sign is specified for "Negative Numbers" when creating a form, only "+" or "-" can be specified for the input data.</li> <li>- , (Comma)<br/>Actually it is skipped.</li> <li>- . (Decimal point)<br/>Only one character can be specified.</li> </ul> |
|                                    | Date   | Y                        | 0 to 9  | Specify in YYYYMMDD format.<br>YYYY: Western Calendar Year<br>MM: 01 to 12 (month)<br>DD: 01 to 31  |
| Text field                         | Standard   | Y                        | String  |   |
|                                    | Mail   | Y                        | 0 to 9  | Specify a 7-digit number, without hyphens (-).  |
| Block text field                   |  | Y                        | String  | When inserting linefeed in desired locations, it is necessary to specify the character code to be used as the linefeed code. 01 to 7F can be specified as the linefeed code. (There are some character codes that cannot be specified, depending on the input data format.)(*4)   |
| OCR-B field                        |  | Y                        | 0 to 9<br>A to Z<br>* + - = / . , ; :<br>" ( ) < > # &<br>¥ % |   |
| Radio button                       |  | Y                        | String  | Strings used as values for selected or unselected state are specified.  |
| Check box                          |  | Y                        | String  | Strings used as values for selected or unselected state are specified.  |

**Table 8.2** Field Type and Data Format (2 / 5)

| Field type    | Type<br>Barcode type<br>Value for selected<br>state | Input<br>data<br>required<br>(*1) | Available<br>characters<br>(*2)   | Remarks  |
|---------------|---|-----------------------------------|---|--|
| Barcode field | JAN Standard  | Y                                 | 0 to 9  | Specify in 12 digits.  |
|               | JAN Reduced   | Y                                 | 0 to 9  | Specify in 7 digits.   |
|               | Code 3 of 9   | Y                                 | 0 to 9<br>A to Z<br>+ - / \$ % .<br>Half-width<br>space   | Specify within the number of digits<br>specified for the field (1 to 999).(*5) (*6)  |
|               | Code 3 of 9 (EIAJ<br>Compliant)                     | Y                                 | 0 to 9<br>A to Z<br>a to z<br>!"#\$%&'()+,-<br>/<br>;<=>?@[^\`<br>{ }~?.<br>Half-width<br>space<br>。」「、・<br>ヲイイイ<br>ヤヨヅ<br>ア to ャ<br>。。 | Specify either of the following at the start of<br>the input data:<br>- Z<br>- 3N<br>- 1P<br>Specify within the number of digits<br>specified for the field (1 to 999). (*6)   |
|               | Industrial 2 of 5                                   | Y                                 | 0 to 9  | Specify in the number of digits specified for<br>the field (1 to 999).   |
|               | Interleaved 2 of 5<br>(ITF)                         | Y                                 | 0 to 9  | Specify in the number of digits specified for<br>the field (1 to 999).   |
|               | NW-7  | Y                                 | 0 to 9<br>A to D<br>+ - \$ / : .  | Specify from 3 to 34 digits.<br>"A to D" is used as the START/STOP code<br>(included in the number of digits for the<br>data).   |
|               | Custom  | Y                                 | 0 to 9<br>A to Z<br>-   | Specify with 7 digits for the post code and<br>13 digits for the address. A 7-digit post<br>code must be specified. If the length is less<br>than 20 digits, the missing portions will be<br>filled with a barcode suitable for CC4. |
|               | CODE-128  | Y                                 | ASCII code<br>(*7)  | Specify within the number of digits<br>specified for the field (1 to 999).(*5)   |

**Table 8.2** Field Type and Data Format (3 / 5)

| Field type           | Type<br>Barcode type<br>Value for selected<br>state | Input<br>data<br>required<br>(*1) | Available<br>characters<br>(*2)   | Remarks  |
|----------------------|---|-----------------------------------|---|--|
| Barcode field        | EAN-128<br>(*8)                                     | Y                                 | ASCII code<br>(*7)<br>4 function<br>characters  | Specify within the number of digits<br>specified for the field (1 to 999).(*5)<br>"FNC1" to "FNC4" can be used as function<br>characters. The function character is<br>specified with the following character<br>codes:<br>FNC1: The character code defined in the<br>form<br>FNC2: 0xFC<br>FNC3: 0xFD<br>FNC4: 0xFE |
|                      | EAN-128<br>(Convenience<br>store)                   | Y                                 | ASCII code<br>(*7)<br>4 function<br>characters  | Specify in 44 digits. (*9)<br>Specify "91" at the start of the input data.<br>"FNC1" to "FNC4" can be used as function<br>characters. The function character is<br>specified with the following character<br>codes:<br>FNC1: The character code defined in the<br>form<br>FNC2: 0xFC<br>FNC3: 0xFD<br>FNC4: 0xFE     |
|                      | UPC version A<br>(UPC-A)                            | Y                                 | 0 to 9  | Specify in 11 digits.<br>Add-on data can be specified. (*10)   |
|                      | UPC version E<br>(UPC-E)                            | Y                                 | 0 to 9  | Specify in 6 digits.<br>Add-on data can be specified. (*10)  |
|                      | EAN-13  | Y                                 | 0 to 9  | Specify in 12 digits.<br>Add-on data can be specified. (*10)   |
|                      | U.S. POSTNET<br>(Delivery Point<br>Code)            | Y                                 | 0 to 9  | Specify in 11 digits.  |
|                      | U.S. POSTNET<br>(ZIP + 4 Code)                      | Y                                 | 0 to 9  | Specify in 9 digits.   |
|                      | U.S. POSTNET<br>(5-Digit ZIP Code)                  | Y                                 | 0 to 9  | Specify in 5 digits.   |
|                      | QR Code<br>(Model 1)                                | Y                                 | (*11)   | Specify within the number of digits<br>specified for the field (1 to 9999). (*5)(*12)  |
| QR Code<br>(Model 2) | Y   | (*11)                             | Specify within the number of digits<br>specified for the field (1 to 9999). (*5)(*12) |  |



**Table 8.2** Field Type and Data Format (4 / 5)

| Field type                         | Type<br>Barcode type<br>Value for selected<br>state | Input<br>data<br>required<br>(*1)   | Available<br>characters<br>(*2)   | Remarks  |
|------------------------------------|---|---|---|--|
| Barcode field                      | QR Code<br>(Micro QR)                               | Y   | ASCII code,<br>8-bit binary<br>data,<br>Japanese<br>(*13)                                   | Specify within the number of digits<br>specified for the field (1 to 999).(*5)(*12)  |
|                                    | PDF417  | Y   | ASCII code,<br>extended<br>ASCII code,<br>8-bit binary<br>data                              | Specify within the number of digits<br>specified for the field (1 to 9999). (*5)(*18)  |
|                                    | MaxiCode  | Y   | ASCII code,<br>extended<br>ASCII code,<br>8-bit binary<br>data                              | Specify within the number of digits<br>specified for the field (1 to<br>999).(*5)(*18)(*19)  |
|                                    | Intelligent Mail<br>Barcode                         | Y   | 0 to 9  | Specify in the number of digits specified for<br>the field (20, 25, 29 or 31).   |
|                                    | GS1 DataBar<br>Omnidirectional                      | Y   | 0 to 9  | Specify in 13 digits.  |
|                                    | GS1 DataBar<br>Truncated                            | Y   | 0 to 9  | Specify in 13 digits.  |
|                                    | GS1 DataBar<br>Stacked                              | Y   | 0 to 9  | Specify in 13 digits.  |
|                                    | GS1 DataBar<br>Stacked<br>Omnidirectional           | Y   | 0 to 9  | Specify in 13 digits.  |
|                                    | GS1 DataBar<br>Limited                              | Y   | 0 to 9  | Specify in 13 digits. (*20)  |
|                                    | GS1 DataBar<br>Expanded                             | Y<br>(*21)  | 0 to 9<br>A to Z<br>a to z<br>! "% & ' ? * + , -<br>./ : ; < > = ? _<br>SPACE<br>FNC1(0x0a) | Specify within the number of digits<br>specified for the field (16 to 74). (*22)   |
| GS1 DataBar<br>Expanded<br>Stacked | Y<br>(*21)  | 0 to 9<br>A to Z<br>a to z<br>! "% & ' ? * + , -<br>./ : ; < > = ? _<br>SPACE<br>FNC1(0x0a) | Specify within the number of digits<br>specified for the field (16 to 74). (*22)            |  |
| Barcode field<br>(U.S. Postal FIM) | FIM A<br>(U.S.Postal FIM)                           | -   | -   |  |
|                                    | FIM B<br>(U.S.Postal FIM)                           | -   | -   |  |
|                                    | FIM C<br>(U.S.Postal FIM)                           | -   | -   |  |
| Image field                        |   | Y   | File Name   | Specify the file name of the media data to<br>be output (file name only). The directory<br>name is specified when outputting the form. |
| Date field                         |   | -   | -   | -  |

**Table 8.2** Field Type and Data Format (5 / 5)

| Field type                 | Type<br>Barcode type<br>Value for selected<br>state | Input<br>data<br>required<br>(*1) | Available<br>characters<br>(*2) | Remarks  |
|----------------------------|---|-----------------------------------|---------------------------------|--|
| Time field                 |   | -                                 | -                               | -  |
| Post code field            |   | Y                                 | 0 to 9                          | Specify a 7-digit number, without hyphens (-).   |
| Hidden field (Record) (*3) |   | Y                                 | (*14)                           |  |
| Group item                 |   | Y                                 | (*15)                           | If "Use Group Item Name" is specified for [Group Item Handling] on the [Data Definitions] window, specify the data for the configuration items with a fixed length format without separator characters, regardless of the input data format. (*16)<br>If "Use Configuration Item Name" is specified for [Group Item Handling], specify separator characters for each configuration item. (*17) |

\*1: Y: Input data is required. However, it may not be required depending on the specified operator.

-: Input data is not required.

For operators, refer to the following:

=> "[Table 6.1 Types of operators](#)"

- \*2: Character codes are described for the Shift-JIS. When a character code other than Shift-JIS is used, specify corresponding characters.
- \*3: Be sure to specify data of 1 digit or more for the fields in which the ADD, SUB, MUL, DIV, and CAL operators are specified. For hidden fields (records), this applies only when the field format (record format) is "Numeric".
- \*4: The following character codes cannot be used when the input data format is "Variable Length", "Fixed Length", or "CSV Format":  
0A 0D 1A 1F 20  
The following character codes cannot be used when the input data format is "XML Format":  
0D 1A 1F 20
- \*5: If the data is not of sufficient data length, spaces are inserted.
- \*6: Do not specify start and stop codes in data because List Creator automatically adds the codes.
- \*7: If the input data of the barcode is encoded during the form output, specify as shown below:
- Data length of the barcode field  
The input data length before encoding + 2
  - Byte size in the [Data Definitions] window (only if the input data format is "Fixed Length")  
Byte size after encoding
- Encoded input data cannot be used in the following circumstances:
- When an operand is specified for the data field in the [Data Definitions (Report Header)] tab and the [Data Definitions] tab of the [Data Definitions] window
  - When a data field name is specified for the operand in the [Data Definitions (Report Header)] tab and the [Data Definitions] tab of the [Data Definitions] window
  - When a data field name is specified in the [Extended] tab of the [Data Definitions] window
- If the input data of the barcode is not encoded during the form output, the following character codes cannot be used:
- 00 0A 0D 1A 1F
- \*8: When adding the function character "FNC1" to distinguish between EAN-128 and CODE-128, specify the function character "FNC1" code at the beginning of the data.
- \*9: If the data is of 43 digits, the entire check digit is automatically added.
- \*10: Add-on data cannot be used in the following circumstances:
- In the body of a simple table form

- In the body of a block repeat form
- In a sticky label form
- For configuration items of a group item, when "Use Group Item Name" is specified for [Group Item Handling]
- If repeats are specified

\*11: The types of characters that can be used are shown below. When a character code other than Shift-JIS is used, specify corresponding characters.

- Numeric : 0 to 9
- Alphabet : A to Z, a to z
- Symbol : \* + - \$ / : . % Half-width space
- JIS8 : ASCII code (however, 00, 0A, 0D, 1A or 1F cannot be used)
- Japanese : 0x8140 to 0x9FFC, 0xE040 to 0xEBBF in Shift-JIS characters

\*12: Use the following guideline to calculate the maximum number of output characters (or the maximum number of output characters for one split barcode when split) for the QR code. However, for output data consisting of several different character types, the actual number of output characters will be less than the calculated number, because of the extra data added for each change in the character type.

| Barcode type       | Error correction ratio (%) | Maximum number of characters that can be output for each character type |                        |           |               |
|--------------------|----------------------------|---|------------------------|-----------|---------------|
|                    |                            | Numeric only  | Alphanumeric + symbols | JIS8 only | Japanese only |
| QR Code (Model 1)  | 7                          | 1167  | 707                    | 486       | 299           |
|                    | 15                         | 877   | 531                    | 365       | 225           |
|                    | 25                         | 738   | 447                    | 307       | 189           |
|                    | 30                         | 498   | 302                    | 207       | 127           |
| QR Code (Model 2)  | 7                          | 7089  | 4296                   | 2953      | 1817          |
|                    | 15                         | 5596  | 3391                   | 2331      | 1435          |
|                    | 25                         | 3993  | 2420                   | 1663      | 1024          |
|                    | 30                         | 3057  | 1852                   | 1273      | 784           |
| QR Code (Micro QR) | Only fault sensing         | 5   | -                      | -         | -             |
|                    | 7                          | 35  | 21                     | 15        | 9             |
|                    | 15                         | 30  | 18                     | 13        | 8             |
|                    | 25                         | 21  | 13                     | 9         | 5             |

\*13: The following character codes of ASCII code cannot be used:

00 0A 0D 1A 1F

For Japanese, 0x8140 to 0x9FFC and 0xE040 to 0xEBBF in Shift-JIS characters can be used.

\*14: If the field format (record format) is "Numeric", specify numerals. If it is "Text", specify strings.

\*15: Follow the field type of the configuration items in the group item.

\*16: Specify as follows:

XXXXXX999999XX9999

\*17: Specify as follows, if the separator character is a "comma (,)":

XXXXXX,999999,XX,9999

\*18: Use the following guideline to calculate the maximum number of characters that can be output in PDF417.

However, for output data consisting of several different character types, the actual number of output characters will be less than the calculated number, because of the extra data added for each change in the character type.

| Barcode type | Maximum number of characters that can be output for each character type |                        |           |               |
|--------------|---|------------------------|-----------|---------------|
|              | Numeric only  | Alphanumeric + symbols | JIS8 only | Japanese only |
| PDF417       | 1215 to 2710(*)   | 830 to 1850(*)         | -         | -             |

- \*: In List Creator, the maximum number of characters varies because error correction is automatically set based on the conditions such as the data specified for the barcode.
- \*19: When MaxiCode is selected in the barcode type, pay attention to the resolution of the printer for output and the maximum number of characters that can be output. For details, refer to the "Cautions regarding MaxiCode" in the Designer Help guide.
- \*20: The first 1 byte of the data must be "0" or "1". Specifying the data with the first byte other than "0" or "1" causes an error.
- \*21: The data must include an application identifier (AI).  
 With an AI, the data length is expected to be at least 16 digits because it is assumed that GTIN is specified for an AI.  
 However, numbers other than GTIN can also be specified for an AI, so the data with less than 16 digits can be output. In this case, perform the following procedure to define the form style and create data.
  - 1: In the [Form Style Descriptor], select [File] - [Form Properties] - [Barcodes] tab, and then uncheck the [Fill with a blank if data length does not reach field length] checkbox.
  - 2: Open the data file and specify the data of less than 16 digits in the barcode data.
- \*22: Follow the guidelines below to calculate the maximum number of characters that can be output in GS1 DataBar Expanded/GS1 DataBar Expanded Stacked.  
 However, for output data consisting of several different character types, the actual number of output characters will be less than the calculated number, because of the extra data added for each change in the character type.


| Barcode type  | Maximum number of characters that can be output for each character type |                        |           |               |
|---|---|------------------------|-----------|---------------|
|   | Numeric only  | Alphanumeric + symbols | JIS8 only | Japanese only |
| GS1 DataBar Expanded/<br>GS1 DataBar Expanded Stacked | 74  | 41                     | -         | -             |

It is recommended that specify "01" for 2 bytes at the beginning of data. When "01" is specified, data up to 74 characters can be input.

### 8.2.3 Understanding the Data Length of each Field

Data must be created in accordance with the length of the field.

The data length varies depending on the input data format. If the field format (record format) is "Numeric", symbols, currency signs, "commas (,)", and decimal points can be specified in the data. These are edited to the specified format when outputting the form.

|   |   |
|---|---|
| <br><b>Caution</b> | <p>- When input data is in Unicode, the following character codes are handled as "yen mark (¥)":<br/>           For UNICODE (UTF16): 0x005C, 0x00A5<br/>           For UNICODE (UTF8): 0x5C, 0xC2A5</p> |
|---|---|

- When the input data format is "Variable Length", "CSV Format", or "XML Format"

#### (10) Data length / (12) Decimal places

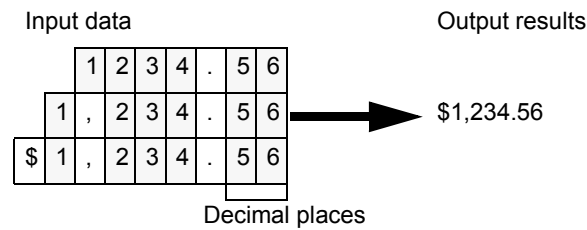
Data is created within the "Data length". The data can be shorter than the data length. Data exceeding the data length is trimmed to fit in the data length.

If the field is numeric or if the field format (record format) of a hidden field (record) is "Numeric", the data length does not include symbols, currency signs, "commas (,)", and decimal points.

The decimal places indicate the number of digits used for decimals in the data length.

For example, no matter which of the following formats is used for the data, it is edited to the same format and output.

When the data length is "6" and the formatting is "\$\$, \$\$9.99":



(\*) The shaded areas are counted for data length.

• When the Input Data Format is "Fixed Length"

(11) Byte size / (12) Decimal places

Data is created in the length of "Byte size". If the actual data is less than the byte size, be sure to add half-width spaces to create data of the byte size.

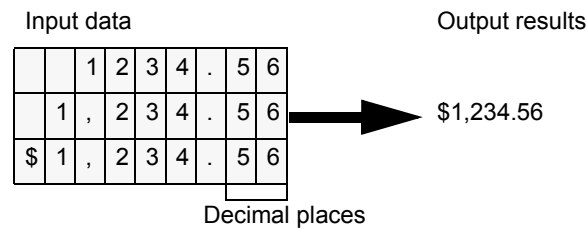
If the field is numeric or if the field format (record format) of a hidden field (record) is "Numeric", the byte size includes symbols, currency signs, "commas (,)", and decimal points.

The decimal places indicate the number of digits used for decimals in the byte size.

The initial byte size displayed in the [Data Definitions (Report Header)] tab and the [Data Definitions] tab of the [Data Definitions] window is "Data length". If the data is to include symbols, currency signs, "commas (,)", or decimal points, the byte size must first be increased by the number of additional digits.

For example, no matter which of the following formats is used for the data, it is edited to the same format and output.

When the byte size is "9" and the formatting is "\$\$, \$\$9.99":



(\*) The shaded areas are counted for byte size.

**Caution** - When a field is printed as a configuration item of a group item in the input data report (i.e., when "Use Group Item Name" is specified for a field belonging to a group item in [Group Item Handling] in the [General] tab of the [Data Definitions] window), specify the configuration item data using the fixed length format without separator characters, regardless of the input data format.

### 8.2.4 Sorting the Data

Sort the records (lines) of the input data in the order you want to output. (List Creator does not sort input data.) Data must be sorted in advance in units of calculations such as subtotals and totals.

(14)Control level

This displays the control level when a BREAK is specified.  
Data is sorted in order from the fields with the lowest control level (largest number).

## 8.3 Designing Input Data for Multi-form Output

The method for creating data files when outputting multi-forms is explained below.

When outputting multi-forms, it is necessary to enter form names in the data file. It is not necessary to define fields that receive the form names in the [Form Style Descriptor] window.

### 8.3.1 When the Input Data Format is "Variable Length" or "CSV Format"

The form name is placed at the beginning of each record (line). Next is the separator character specified in the [General] tab in the [Data Definitions] window. Then, the data, in the same format as when a single form, is output.



**Caution**

- For all forms in the multi-form output, the input data format and separator characters should be the same.
- Specify the separator character when the form is output.



**Note**

- A data file with the field name label can be created by adding a row of field names defined in the form to the top of the input data file. For details on a data file with the field name label, refer to the following:  
=> ["8.4 Designing a Data File with the Field Name Label"](#)

The following shows an example of data files to be output for 4 separate forms, and an example of the data file for multi-form output.

- **When outputting 4 forms separately**

Create 4 data files, and output 4 separate forms.

The input data format of the data files is "Variable Length", and the separator character is "Vertical Bar (|)".

Data for "Bill" form

```
0000000 | Madison Avenue XXX | Des Moines IA | ABC Electronics | 20031221
```

Data for "Statement" form

```
ABC Electronics | 20031221 | 03101 | 005 | 0020 | Color television | 3 | 120000
ABC Electronics | 20031221 | 03101 | 005 | 0022 | Karaoke set | 1 | 82000
ABC Electronics | 20031221 | 03101 | 005 | 0100 | Electric cleaner | 2 | 23000
```

Data for "BillState" form

```
0000000 | Jefferson Avenue XXX | New York NY | XYZ Electronics | 20031221
```

Data for "DetailBill" form

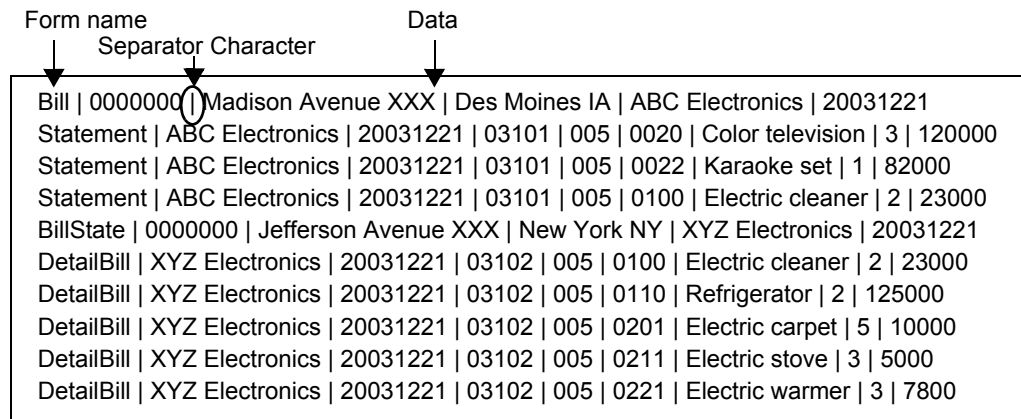
```
XYZ Electronics | 20031221 | 03102 | 005 | 0100 | Electric cleaner | 2 | 23000
XYZ Electronics | 20031221 | 03102 | 005 | 0110 | Refrigerator | 2 | 125000
XYZ Electronics | 20031221 | 03102 | 005 | 0201 | Electric carpet | 5 | 10000
XYZ Electronics | 20031221 | 03102 | 005 | 0211 | Electric stove | 3 | 5000
XYZ Electronics | 20031221 | 03102 | 005 | 0221 | Electric warmer | 3 | 7800
```

- **When outputting 4 forms as multi-forms**

Create a single data file that includes the data for four forms, and output it as a single form.

The input data format of the data files is "Variable Length", and the separator character is "Vertical Bar (|)".

The form name is placed at the beginning of each record (line). Next is the "Vertical Bar (|)" separator character. Then, the data, in the same format as when a single form, is output.



**Figure 8.2** Data file description format when outputting multi-forms (variable length)

### 8.3.2 When the Input Data Format is "Fixed Length"

Of the forms used when outputting multi-forms, the length of the longest form name is set as the "Form Name Data Field Length". (It is OK for the "Form Name Data Field Length" to be longer than the length of the longest form name.)

The form name is placed at the beginning of each record (line). If the form name is shorter than the form name data field length, be sure to add half-width spaces to create data of the form name data field length.

If "Yes" is specified on the [General] tab of the [Data Definitions] window, select any single-byte character to use as the separator character. If "No" is specified, no separator character is used.

Then, the data, in the same format as when a single form, is output.



#### Caution

- For all forms in the multi-form output, the input data format and separator characters should be the same.
- Use the same form name data field length throughout the data file.
- The form name data field length does not include separator characters. The separator character is 1 byte (for the character code other than Shift-JIS, a character with a code 0x007F/0x7F or less).
- Specify the separator character and form name data field length when the form is output.
- When the character code of input data is anything other than Shift-JIS, the input data format is "Fixed Length", and "Yes" is specified for the separator character, specify a character with a code less than 0x007F/0x7F for the separator character in the input data.

The following shows an example of data files to be output for 4 separate forms, and an example of the data file for multi-form output.

- When outputting 4 forms separately

Create 4 data files, and output 4 separate forms.

The input data format of the data file is "Fixed Length", and the separator character is "Yes" (space).

Data for "Bill" form

0000000 Madison Avenue XXX Des Moines IA ABC Electronics 20031221

Data for "Statement" form

ABC Electronics 20031221 03101 005 0020 Color television 3 120000  
 ABC Electronics 20031221 03101 005 0022 Karaoke set 1 82000  
 ABC Electronics 20031221 03101 005 0100 Electric cleaner 2 23000

Data for "BillState" form

0000000 Jefferson Avenue XXX New York NY XYZ Electronics 20031221

Data for "DetailBill" form

XYZ Electronics 20031221 03102 005 0100 Electric cleaner 2 23000  
 XYZ Electronics 20031221 03102 005 0110 Refrigerator 2 125000  
 XYZ Electronics 20031221 03102 005 0201 Electric carpet 5 10000  
 XYZ Electronics 20031221 03102 005 0211 Electric stove 3 5000  
 XYZ Electronics 20031221 03102 005 0221 Electric warmer 3 7800

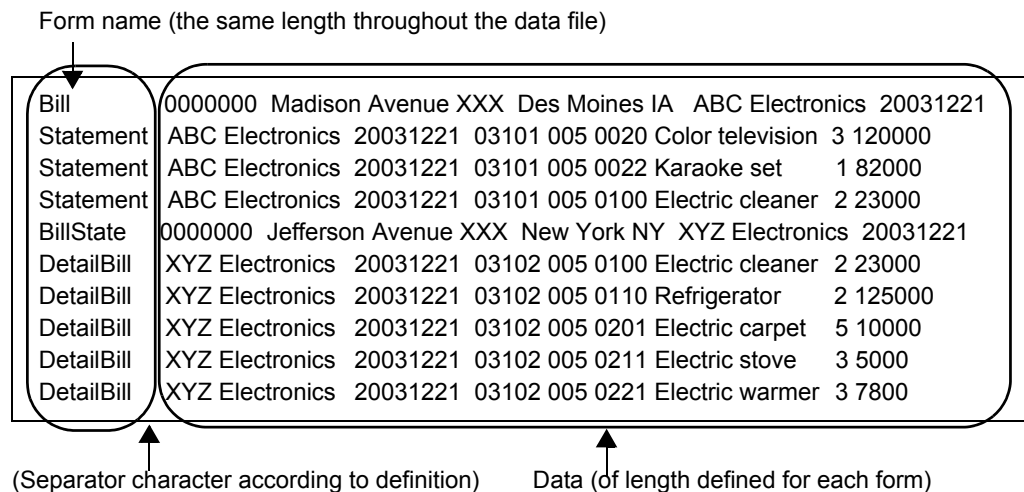
- When outputting 4 forms as multi-forms

Create a single data file that includes the data for four forms, and output it as a single form.

The input data format of the data file is "Fixed Length", and the separator character is "Yes" (space).

Of the forms to be used, set the longest form name of 10 bytes as the "Form Name Data Field Length".

The form name appears at the beginning of each record (line), and half-width spaces are inserted if the 10 byte "Form Name Data Field Length" is not met. Next is a "space" separator character. Then, the data, in the same format as when a single form, is output.



**Figure 8.3** Data file description format when outputting multi-forms (fixed length)



### 8.3.3 When the Input Data Format is "XML Format"

When the input data in "XML Format" is output in multi-forms, enclose the XML data in multi-form tags <FjL-CForm> and specify the form names to be output with the attribute "name".



- Specify the multi-form tag <FjLForm> in XML file. You should not use the tag <FjLForm> when defining form style descriptor.

The following shows an example of XML data to be output for 2 separate forms, and an example of the same XML data formatted for multi-form output.

- When outputting 2 forms separately

Create 2 sets of XML data, and output 2 separate forms.

Data for "Bill" form

```
<Bill>
  <PostCode>000000</PostCode>
  <Address>Madison Avenue XXX Des Moines IA</Address>
  <Customer>ABC Electronics</Customer>
  <BillingDate>20031221</BillingDate>
</Bill>
```

Data for "Statement" form

```
<Statement>
  <Customer>ABC Electronics</Customer>
  <BillingDate>20031221</BillingDate>
  <RecordNo.>301</RecordNo.>
  <ContactCode>005</ContactCode>
  <Detail>
    <ProductCode>0020</ProductCode>
    <ProductName>Color television</ProductName>
    <Quantity>3</Quantity>
    <UnitPrice>12000</UnitPrice>
  </Detail>
  <Detail>
    <ProductCode>0022</ProductCode>
    <ProductName>Karaoke set</ProductName>
    <Quantity>1</Quantity>
    <UnitPrice>82000</UnitPrice>
  </Detail>
  <Detail>
    <ProductCode>0100</ProductCode>
    <ProductName>Electric cleaner</ProductName>
    <Quantity>2</Quantity>
    <UnitPrice>23000</UnitPrice>
  </Detail>
</Statement>
```

- When outputting 2 forms as multi-forms

Create a set of XML data that includes the data for 2 forms, and output it as a single form.

Enclose the XML data in the multi-form tags. Also, specify the form name with the multi-form tag attribute "name". Enter the data root defined in the Form Style Descriptor for child element of multi-form tag.

Multi-form tags are specified under the XML root.

```

<XMLRoot>
  <FjLCForm name="Bill">
    <Bill>
      <PostCode>0000000</PostCode>
      <Address>Madison Avenue XXX Des Moines IA</Address>
      <Customer>ABC Electronics</Customer>
      <BillingDate>20031221</BillingDate>
    </Bill>
  </FjLCForm>
  <FjLCForm name="Statement">
    <Statement>
      <Customer>ABC Electronics</Customer>
      <BillingDate>20031221</BillingDate>
      <RecordNo.>301</RecordNo.>
      <ContactCode>005</ContactCode>
      <Detail>
        <ProductCode>0020</ProductCode>
        <ProductName>Color television</ProductName>
        <Quantity>3</Quantity>
        <UnitPrice>12000</UnitPrice>
      </Detail>
      <Detail>
        <ProductCode>0022</ProductCode>
        <ProductName>Karaoke set</ProductName>
        <Quantity>1</Quantity>
        <UnitPrice>82000</UnitPrice>
      </Detail>
      <Detail>
        <ProductCode>0100</ProductCode>
        <ProductName>Electric cleaner</ProductName>
        <Quantity>2</Quantity>
        <UnitPrice>23000</UnitPrice>
      </Detail>
    </Statement>
  </FjLCForm>
</XMLRoot>

```

### 8.3.4 Points

Note the following points when designing input data for multi-form output:

- When performing multi-form output, an error may occur if there are two or more consecutive pieces of blank data in the same form (when linefeed processing is specified to perform during form output by linefeed code control, an error will not occur).

The state when there are two or more consecutive pieces of blank data is defined as the state when all of the following conditions are met:

- A form name and a form name data separator character for multi-form are described
- There is no other data after the form name in the record
- There is no succession of records for the same form name

Therefore, to perform multi-form output, it is recommended to design an input data such that two or more consecutive blank lines will not be in the data.

The following shows an example of two or more consecutive blank lines in input data.

Example) When the separator character for form is a vertical bar (|) and input data is in CSV format

```
Bill|0000000 Madison Avenue XXX Des Moines IA ABC Electronics 20031221
Statement|ABC Electronics 20031221 03101 005 0020 Color television 3 120000
Statement|ABC Electronics 20031221 03101 005 0022 Karaoke set 1 82000
Statement|ABC Electronics 20031221 03101 005 0100 Electric cleaner 2 23000
BillState|0000000 Jefferson Avenue XXX New York NY XYZ Electronics 20031221
DetailBill|XYZ Electronics 20031221 03102 005 0100 Electric cleaner 2 23000
DetailBill| <-Blank line
DetailBill| <-Blank line
DetailBill|XYZ Electronics 20031221 03102 005 0211 Electric stove 3 5000
DetailBill|XYZ Electronics 20031221 03102 005 0221 Electric warmer 3 7800
```

## 8.4 Designing a Data File with the Field Name Label

The following describes the method for creating a data file to output forms by handling the input data as a data file with the field name label.

### 8.4.1 Definition of a Data File with Field Name Label

A data file with the field name label is a data file in which a label line containing the field name of the form descriptor information for each data (a field name label line) is added.

- Example of data file

```
"photo.bmp", "Nihon Hanako", "123456", "19960401"<- Data line
```

- Example of a data file with the field name label

```
IDPhoto, Name, EmployeeNumber, EmploymentStartDate<- Field name label line  
"photo.bmp", "Nihon Hanako", "123456", "19960401"<- Data line
```

A field name label line can be added to the following data files.

- Variable length
- CSV format

When outputting forms using a data file with the field name label, there is no setting to define in the form descriptor information. In the form output interface, specify whether to handle the input data as a data with the field name label. When using a data file with the field name label as input data, it is not required to match the order of field names and input data in the [Data Definitions] tab of the [Data Definitions] window. Therefore, it is not required to edit the input data again when the order of the fields in form descriptor information is changed.



- You cannot use a data file with the field name label in the following cases:
  - The input data format is "Fixed Length" or "XML Format"
  - Using SAP R/3 RDI integration



**Note**

- You can use the following functions to create a data file with the field name label (a data file in CSV format separated by comma).
  - XBRL data support function
  - Navigator links function

### 8.4.2 Description Format of a Data File with the Field Name Label

Enter a field name label line in a data file when handling the input data as data file with the field name label.

The following explains the description format of the field name label line.



**Note**

- This section explains the description format of a field name label line for outputting a single form. For other description formats, refer to the following.
  - When performing multi-form output  
=> ["8.4.4 When Performing Multi-form Output"](#)
  - When using the single page format in which repeats are specified for fields  
=> ["8.4.5 When Using the Single Page Format in which Repeats are Specified for Fields"](#)
  - When using the multipart table format in which a report header (RH) partition is defined  
=> ["8.4.6 When Using the Multipart Table Format in which a Report Header \(RH\) Partition is Defined"](#)

- **Description format**

- Enter a field name label line at the beginning of the data file.
- Each field name is separated by a separator character.  
As the separator character, use a separator character used in the input data (data line).



- (2) : Data line of "Bill" form
- (3): Field name label line of "Statement" form
- (4): Data line of "Statement" form
- (5): Data line of "Statement" form
- (6): Field name label line of "BillState" form
- (7): Data line of "BillState" form
- (8): Field name label line of "DetailBill" form
- (9): Data line of "DetailBill" form
- (10): Data line of "DetailBill" form
- (11): Data line of "DetailBill" form

### 8.4.5 When Using the Single Page Format in which Repeats are Specified for Fields

When the form is in the single page format, repeats can be defined in the fields.

The following shows an example when using single page format in which repeats are specified for fields. The input data format is "Variable Length", and the separator character is "Vertical Bar (|)".

In this instance, the number of repeats are defined in the [Data Definitions] tab of the [Data Definitions] window as follows:

| Data Field Name | Number of Repeats |
|-----------------|-------------------|
| Name            | 0                 |
| Score           | 5                 |

```
Name|Score|Score|Score|Score|Score<- Field name label line
Nihon Hanako|60|70|80|90|100<- Data line
```

- First line  
Enter a field name label line.
  - Enter the field name "Name" once.
  - Enter the field name "Score" 5 times repeatedly.
- Second line  
Enter a data line.

### 8.4.6 When Using the Multipart Table Format in which a Report Header (RH) Partition is Defined

The following shows an example when using the multipart table form in which a report header (RH) partition is defined. The input data format is "Variable Length", and the separator character is "Vertical Bar (|)".

In this instance, the operators are defined in the [Data Definitions (Report Header)] tab of the [Data Definitions] window as follows:

| Data Field Name | Operand |
|-----------------|---------|
| DepartmentName  | (*1)    |
| EmployeeNumber  | SHARED  |
| Name            | SHARED  |

\*1: No operator specified

|  |   |
|--|---|
| DepartmentName   | <- Field name label line for the fields defined in the RH (Report Header) partition   |
| Sales department) 1st Development Division                 | <- Data line for the fields defined in the RH (Report Header) partition               |
| EmployeeNumber Name Month Date ...(omitted)...             | <- Field name label for the fields defined in other than RH (Report Header) partition |
| 385945 Nihon Hanako 1 6 830 1715 8 0 Distant Business Trip | <- Data line for the fields defined in other than RH (Report Header) partition        |
| 385945 Nihon Hanako 1 11 830 2130 8 3.5                    | <- Data line for the fields defined in other than RH (Report Header) partition        |

- First line  
Enter a field name label line for the fields defined in the RH (Report Header) partition.
  - Enter the field name "DepartmentName".
- Second line  
Enter a data line for the fields defined in the RH (Report Header) partition.
- Third line  
Enter a field name label line for the fields defined in other than RH (Report Header) partition.
  - Enter the field name "EmployeeNumber" and "Name".
- Subsequent lines  
Enter data lines for the fields defined in other than RH (Report Header) partition.

#### 8.4.7 Specifying Operation when Excess or Shortage of Sata is Detected in a Data File with the Field Name Label

When using a data file with the field name label, you can use the setup command for List Creator shown below to specify operation when excess or shortage of data is detected in the data file. For details, refer to the section about the setup command for List Creator in the Online Manual "Setup & Operation".

There are two kinds of operation when excess or shortage of data is detected in a data file:

- An error occurs when excess or shortage of data is detected in a data file
- An error does not occur when excess or shortage of data is detected in a data file

According to the setting above, a data file with the field name label must be created as follows:

- When specifying "an error occurs when excess or shortage of data is detected in a data file"
  - Of the fields displayed in the [Data Definitions (Report Header)] tab and the [Data Definitions] tab of the [Data Definitions] window, enter field names of all fields that require input data in the field name label line.
- When specifying "an error does not occur when excess or shortage of data is detected in a data file"
  - Of the fields displayed in the [Data Definitions (Report Header)] tab and the [Data Definitions] tab of the [Data Definitions] window, you do not have to enter field names of all fields that require input data in the field name label line. If you do not enter the name of a field that requires input data, the input data for the field name is regarded as being omitted.

- You can enter the name of a field that is not displayed in the [Data Definitions (Report Header)] tab and the [Data Definitions] tab of the [Data Definitions] window. In this case, the input data for the entered field name will be skipped.



- An error occurs if the number of repeats does not match the input data (rows) entered. For details on when repeats are specified for fields, refer to the following:  
=> ["8.4.5 When Using the Single Page Format in which Repeats are Specified for Fields"](#)



## 8.5 New Page/New Frame of Form

When a form is output, the timings of when a new page/frame starts differ for different form formats. The new frame function is valid only when the form is in the free frame format.

### 8.5.1 Page Feed Timing

A "new page" means finishing the output of a form on a page and continuing to the next page. The following explains the timings of starting a new page for each form format.

- For a single page form

A new page starts for each record (line) of input data.



**Caution**

- When the input data format is "XML Format", a new page starts for each data root.

- For a simple table, block repeat, or sticky label form

A new page starts when the number of repeats for the details exceeds the number of the details for one page, or when a BREAK occurs.

A new page does not start simultaneously with a new record (line).



**Caution**

- When the input data format is "XML Format", the end tag of a data root does not start a new page.

- For a multipart table form

A new page starts when:

- The number of repeats for the details exceeds the number of the details for one page (You can specify when to start a new page in the [Page Feed Timing] of the [General] tab on the [Data Definitions] window.)
- A BREAK occurs
  - A BREAK occurs, as follows:
    - It occurs when a control footer partition (CF) is output which specifies a new page (in the [Control Type] of the [Post-Print] in the properties window of the partition)
    - If a new page is specified in the [Extended] tab of the [Data Definitions] window, only a new page starts when a BREAK occurs

A new page does not start simultaneously with a new record (line).



**Caution**

- When the input data format is "XML Format", the end tag of a data root does not start a new page.

- For a free frame form

For a form in the free frame format, the timing when a new page starts differs depending on whether a frame is defined and whether the defined frame is linked.

- When a frame is not defined

A new page starts for each record (line) of input data.



**Caution**

- When the input data format is "XML Format", a new page starts for each data root.

- When a frame is defined

The timing when a new page starts differs depending on whether the defined frame is linked.

- If the defined frame is not linked

A new page starts when:

- The number of repeats for the details exceeds the number of the details for one frame

- A BREAK occurs (when a control footer partition (CF) is output which specifies a new page in the [Control Type] of the [Post-Print] in the properties window of the partition)
- A BREAK occurs (If a new page is specified in the [Extended] tab of the [Data Definitions] window, only a new page starts when a BREAK occurs)

A new page does not start simultaneously with a new record (line).

**Caution**

- When the input data format is "XML Format", the end tag of a data root does not start a new page.
- When a control footer partition (CF) of a fixed partition is defined for a form, the details are not output after the print start position of the control footer partition (CF). A new page starts and output continues.

- If the defined frame is linked  
If the defined frame is linked, the timing when a new page starts and the timing when a new frame starts differ.

A new page starts when:

- A new frame starts (For the last frame linked, a new frame starts a new page)
- A BREAK occurs (when a control footer partition (CF) is output which specifies a new page in the [Control Type] of the [Post-Print] in the properties window of the partition)
- A BREAK occurs (If a new page is specified in the [Extended] tab of the [Data Definitions] window, only a new page starts when a BREAK occurs)

A new page does not start simultaneously with a new record (line).

**Caution**

- When the input data format is "XML Format", the end tag of a data root does not start a new page.
- When a control footer partition (CF) of a fixed partition is defined for a form, the details are not output after the print start position of the control footer partition (CF). A new page or frame starts and output continues.

### 8.5.2 New Frame Timing

A "new frame" means finishing the output of the details in a frame and continuing to the next frame.

**Sales Invoice**

**Multi Electronics**  
1-6-2, Marunouchi,  
Chiyoda-ku, Tokyo  
100-8211

**FUJITSU Co. Ltd.**  
1-5-2 Higashi Shinbashi, Minato-ku, Tokyo  
(105-7123)  
Tel: +81-03-6252-2220  
Fax: +81-03-6252-2221

A "new frame" means finishing the output of the details in a frame and continuing to the next frame.

| Sales Date  |                           |     |            |           | Invoice No. |                     |     |            |              | PIC Code    |  |  |  |  |
|-------------|---------------------------|-----|------------|-----------|-------------|---------------------|-----|------------|--------------|-------------|--|--|--|--|
| 14-Sep-2005 |                           |     |            |           | 05102       |                     |     |            |              | 007         |  |  |  |  |
| #           | Product                   | Qty | Unit Price | Amount    | #           | Product             | Qty | Unit Price | Amount       |             |  |  |  |  |
| 01          | 0001 Desk Lamp            | 7   | \$10,000   | \$70,000  | 11          | 0122 Tumble Dryer   | 2   | \$15,000   | \$30,000     |             |  |  |  |  |
| 02          | 0020 Color TV             | 3   | \$120,000  | \$360,000 | 12          | 0130 Mattress Dryer | 3   | \$15,000   | \$45,000     |             |  |  |  |  |
| 03          | 0023 Video Deck           | 4   | \$50,000   | \$200,000 | 13          | 0140 Air Dryer      | 1   | \$16,000   | \$16,000     |             |  |  |  |  |
| 04          | 0031 Boom Box             | 4   | \$40,000   | \$160,000 | 14          | 0201 Hot Carpet     | 5   | \$10,000   | \$50,000     |             |  |  |  |  |
| 05          | 0041 Component System     | 2   | \$80,000   | \$160,000 | 15          | 0211 Bar Radiator   | 3   | \$5,000    | \$15,000     |             |  |  |  |  |
| 06          | 0042 Mini-Component Syste | 6   | \$40,000   | \$240,000 | 16          | 0221 Table Heater   | 3   | \$7,800    | \$23,400     |             |  |  |  |  |
| 07          | 0040 Karaoke System       | 1   | \$82,000   | \$82,000  |             |                     |     |            |              |             |  |  |  |  |
| 08          | 0100 Vacuum Cleaner       | 2   | \$23,000   | \$46,000  |             |                     |     |            |              |             |  |  |  |  |
| 09          | 0110 Refrigerator         | 2   | \$125,000  | \$250,000 |             |                     |     |            |              |             |  |  |  |  |
| 10          | 0121 Washing Machine      | 2   | \$70,000   | \$140,000 |             |                     |     |            |              |             |  |  |  |  |
|             |                           |     |            |           |             |                     |     |            | <b>Total</b> | \$1,887,400 |  |  |  |  |

The following explains the timings of starting a new frame in the free frame format.

A new frame starts when:

- The number of repeats for the details exceeds the number of the details for one frame
- A control footer partition (CF) is output which specifies a new frame (in the [Control Type] of the [Post-Print] in the properties window of the partition)
- A BREAK occurs (if a new frame is specified)

# Chapter 9

## Defining the Form Output Methods

This chapter explains the method for defining form output methods regarding the output method of the form, operation during output, and operation after output.

Form output methods are set in the [Form Output Methods] window of the form descriptor information, according to the output method you want to use. If you want to simultaneously output forms to multiple destinations, set information for every output method you want to use.

Tabs that require setting per output method are shown below. Refer to the descriptions of each tab in the Designer Help guide according to the output method.

**Table 9.1** Output methods and tab settings

| Tabs to be set / Output method to be used | General | Print | Make e-Record | Fax   | File  | File (Excel) |
|---|---------|-------|---------------|-------|-------|--------------|
| Print                                     | O       | R     | O(*1)         | O(*1) | O(*2) | O(*1)        |
| Make e-Record                             | O       | -     | R             | -     | -     | -            |
| Send FAX                                  | O       | -     | -             | R     | -     | -            |
| Save as PDF file                          | O       | -     | -             | -     | R     | -            |
| Output to Excel file                      | O       | -     | -             | -     | -     | R            |
| Create OWF file                           | O       | -     | -             | -     | -     | -            |


R: Required, Must be specified in form output method or user application

O: Optional

-: Unnecessary

\*1: Only substitute font settings are valid.

\*2: If font embedding is specified, only substitute font settings are valid.

|  |   |
|--|---|
| <br><b>Note</b> | <ul style="list-style-type: none"> <li>- For form style information that has been newly created, the output method is specified as "Print the form". When you do not want to print the form, remove the checkmark from the "Print the form" checkbox on the [Print] tab, or specify another output method when outputting forms.</li> <li>- Settings in the [Print] tab are also enabled when creating OWF file.</li> <li>- OWF file creation can be specified only in a user application. For details, refer to the Online Manual "Application Design".</li> </ul> |
|--|---|

The available output methods and form output methods may vary depending on the operating system of the List Creator print server. For details of the product support, refer to the Online Manual "About the List Creator" or "Setup & Operation".

Some form output methods can be specified when outputting forms. For the settings that can be specified at form output, refer to the "Form Output Interface Function Table" in the Online Manual "Application Design".

For the linkage with the related software products, such as how and to which window of related software the form output methods defined in List Creator are passed, refer to the following:

- Online Manual "Setup & Operation" attached with List Creator on the List Creator print server
- Manuals of each related software product

The procedure for setting form output methods is shown below.

**= Procedure =**

- 1) Select [Start] - [Programs] - [List Creator] - [Form Descriptor List].  
=>The [Form Descriptor List] window is displayed.
- 2) Select the form of which form output methods you want to define in the Form Descriptor List window, and select [File] - [Form Output Methods].  
=>The properties window of the form output method is displayed.
- 3) Define the form output methods. For the definitions, refer to the descriptions of each tab in the Designer Help guide according to the output method you want to use.



**Note**


- For forms that have been newly created, the output method is specified as "Print the form". When you do not want to print the form, remove the checkmark from the "Print the form" checkbox on the [Print] tab, or specify another output method when outputting forms.

- 4) When the definition is complete, save the defined settings. Click the [OK] button on the [Form Output Methods] window.  
=>The definitions are saved, and the display returns to the [Form Descriptor List] window.



**Note**

- Form output methods can also be defined in the [Form Style Descriptor] window. In the [Form Style Descriptor] window, the [Form Output Methods] window can be displayed by performing one of the following operations:

- Click the [Form Output Methods Settings] button  on the toolbar.
- Select [File] - [Form Output Methods Settings].

# **Chapter 10**

## *Designing the Combination-Form*

---

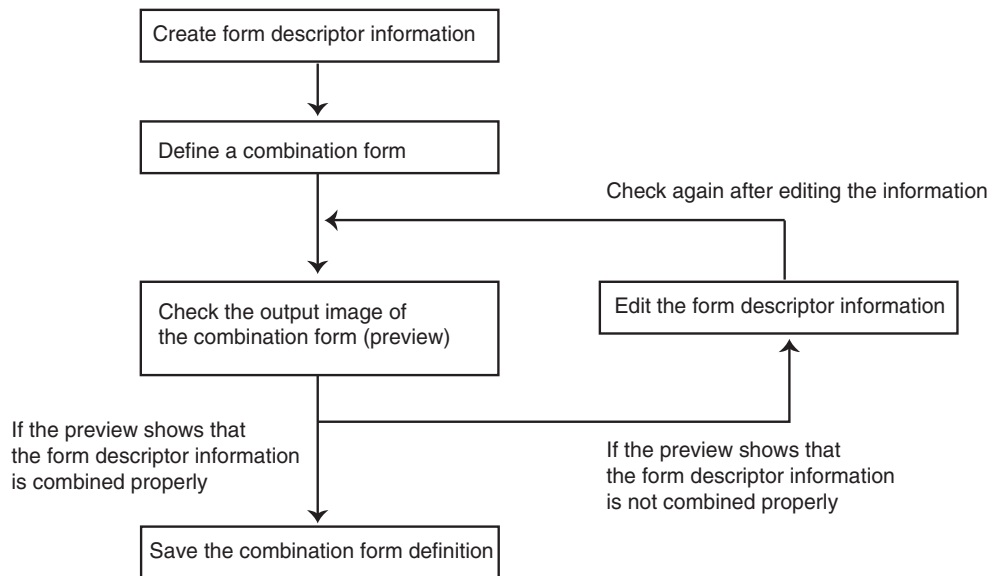
This chapter explains the Combination-Form descriptor, the input data designing, and some important points etc. for creating "Combination-Form Output" that combines multiple pieces of form descriptor information in one page.

## 10.1 Designing the Combination-Form

This section explains how to design the Combination-Form Descriptor file used in Combination-Form output. Combination-Form Descriptor is a definition specifying the combination of the form descriptor information when creating the "Combination-Form Output" by combining multiple pieces of form descriptor information in one page.

### 10.1.1 Combination-Form Defining Flow

Combination-Form Descriptor is created in the following workflow.



- **Creating Form Descriptor Information**

Create a form descriptor information for each form making up a Combination-Form.

For details on creating the form descriptor information, refer to the following:

=> ["Part 2 Form Designing"](#)



**Caution**

- There are some points to note while creating the form descriptor information making up the Combination-Form. For details, refer to the following:

=> ["10.1.5 Cautions for Designing a Combination-Form"](#)

- **Combination-Form Descriptor**

Multiple pieces of form descriptor information which are combined and output is defined as a Combination-Form.

For details on defining a Combination-Form, refer to the following:

=> ["10.1.2 Designing the Combination-Form"](#)

- **Checking the output image of Combination-Form (preview)**

Check the preview of the output image to confirm that the each piece of the form descriptor information is combined properly.

For details on checking the Combination-Form output image, refer to the following:

=> ["10.1.3 Checking the Output Image of Combination-Form"](#)

If the preview shows that the form descriptor information is not combined properly, edit the form descriptor information and check the output image again.

- **Saving the Combination-Form Descriptor**

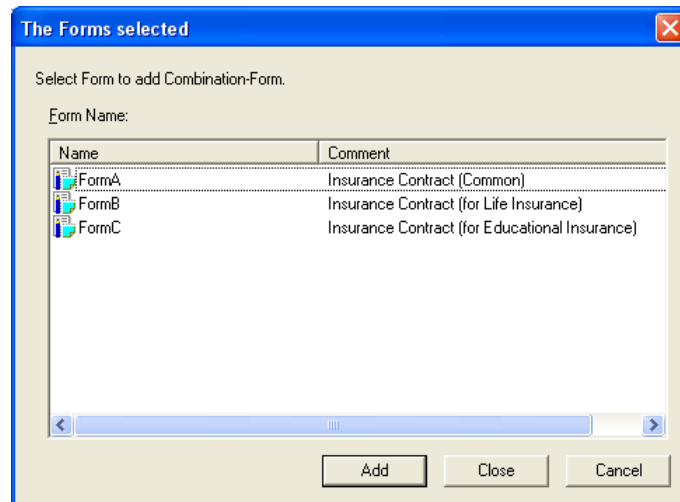
If the preview shows that the form descriptor information is combined properly, save the Combination-Form Descriptor.

For details on saving a Combination-Form, refer to the following:

=> ["10.1.4 Save the Combination-Form Descriptor"](#)







=>The new Combination-Form Descriptor is created.



**Note**

- Overlays of the form descriptor information are output in the order of each form descriptor information in the Combination-Form Descriptor list.  
To change the order, select a form descriptor information in the combination-form descriptor and drop it to a form descriptor information above or below it.  
For details on the form descriptor information overlapping order (output order), refer to the following:  
[=> "10.1.5 Cautions for Designing a Combination-Form"](#)

### 10.1.3 Checking the Output Image of Combination-Form

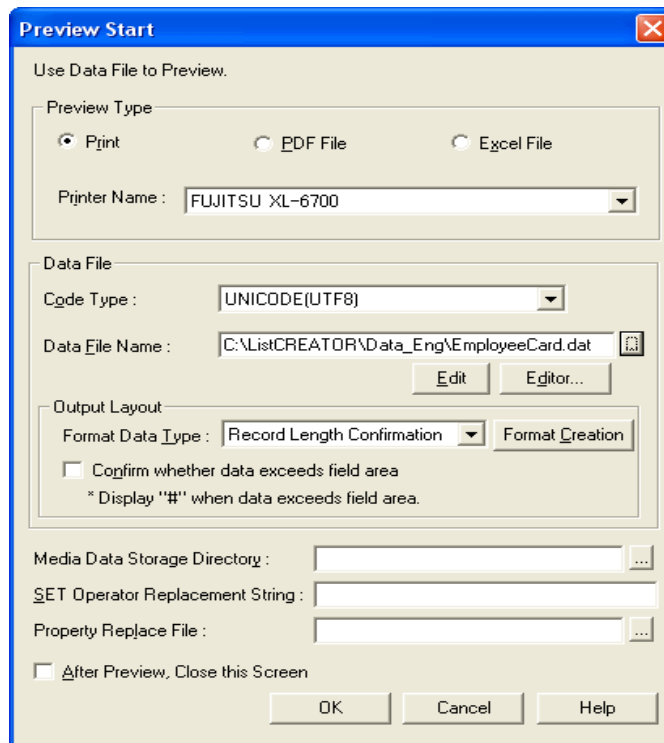
Check the output image of Combination-Form by using the Combination-Form preview function.  
If the preview shows that any form descriptor information is not combined properly, edit the form style descriptor.  
Repeat "previewing" and "editing form style descriptor information" until all forms are combined properly.

#### 10.1.3.1 Previewing the Combination-Form

Check the output image of the Combination-Form (preview) by the following operation.

##### = Procedure =

- 1) Select the Combination-Form to be previewed from the Combination-Form Descriptor list, and select [File]-[Preview Combination-Form].  
=>The [Preview Start] window for the Combination-Form is displayed.



- 2) Select "Preview Type".
- 3) Specify a data file (input data).
- 4) If an image field is defined in the form style descriptor, specify the "Media Data Storage Directory".  
Click the button to the right of the input area to browse the directories under [My Computer].
- 5) Specify the replacement string of "%" in the "SET Operator Replacement String" when the variable "%" is specified for the SET operator in the operand of the data field in the [Data Definitions] tab of the [Data Definitions] window.
- 6) To close this window after you are finished previewing, check the "After Preview, Close this Screen" checkbox.
- 7) Click the [OK] button.  
=>The [Preview] window is displayed.

**Reserve Woman Insurance Contract Application Form**

Insurance Type: Reserve Woman Insurance Contract

Contract No.: \*\*\*\*\*Co.Ltd.

Contract Date: 2002/05/31

| Contract Type                  | A          | B         | C         | D         | E         |
|--------------------------------|------------|-----------|-----------|-----------|-----------|
| Expiration Return Amount       | 1,000,000  | 700,000   | 500,000   | 300,000   | 200,000   |
| Insured Benefit                | 10,000,000 | 7,000,000 | 5,000,000 | 3,000,000 | 2,000,000 |
| Hospitalization Insurance Rate | 10,000     | 7,000     | 5,000     | 3,000     | 2,000     |
| Disability Insurance Rate      | 5,000      | 3,500     | 2,500     | 1,500     | 1,000     |
| Competition Responsibility     | 10,000,000 | 7,000,000 | 5,000,000 | 3,000,000 | 2,000,000 |
| Forfeiture Good Will           | 300,000    | 210,000   | 150,000   | 90,000    | 60,000    |
| Insurance period 3 years       | 28,370     | 19,880    | 14,215    | 8,580     | 5,715     |
| Insurance period 5 years       | 17,440     | 12,295    | 8,795     | 5,270     | 3,500     |

Premium Amount: \$16,000

### 10.1.3.2 Edit the Form Style Descriptor

When checking the output image of Combination-Form, if the preview shows that the forms are not combined properly, correct the form style descriptor.

Edit the form style descriptor by performing the following operation.

**= Procedure =**

- 1) Select the form style descriptor to be edited from the Combination-Form Descriptor list, and select [File]-[Open (Form Style Descriptor)].  
=>The [Form Style Descriptor] window of the selected form descriptor information is displayed.
- 2) Edit the form style descriptor and save the form descriptor information.

Preview the Combination-Form again, and confirm that the forms are combined properly.


### 10.1.4 Save the Combination-Form Descriptor

Save the created Combination-Form Descriptor in the Combination-Form Descriptor file.

Save the Combination-Form Descriptor by performing the following operation.

**= Procedure =**

- 1) Select [File] - [Update Combination-Form Descriptor] in the Combination-Form Descriptor list.  
=>The Combination-Form Descriptor file "Lccbform.ini" is saved in the form storage directory.

|  |   |
|--|---|
| <br><b>Note</b> | <ul style="list-style-type: none"> <li>- Lccbform.ini file is created with Shift-JIS character code.</li> <li>- For details on Combination-Form Descriptor file, refer to the following:<br/>=&gt; "Appendix E Combination-Form Descriptor File"</li> </ul> |
|--|---|

## 10.1.5 Cautions for Designing a Combination-Form

To performing Combination-Form output, note the following points when design a Combination-Form Descriptor.

### • Cautions for designing a Combination-Form Descriptor

Note the following points when designing a Combination-Form descriptor.

- The form descriptor information for "multipart table form" cannot be used in a Combination-Form Descriptor.
  - Only "single page forms" can be used to combine with other formats in a Combination-Form. Additionally, the form formats other than single page cannot be used more than once in a Combination-Form. For example, two simple table forms cannot be combined.
  - Less than or equal to ten form style descriptors can be combined in a Combination-Form.
  - For the form output methods, including the output method, the specification defined in the first form descriptor to be output will become valid.
  - The Combination-Form name is set to the following fields when the Combination-Form is output.
    - Print name
    - Title in PDF file properties
    - Title in Excel file properties
    - Title when making e-Record
    - Form name output in PDF file bookmark.
    - Form name output in Excel file sheet name
    - Form name output in audit trail log
    - Form name output in event/system log and message box (\*1)
- \*1: Instead of the form name, the "Combination-Form name: form name" is displayed in the event log/message box.
- Comments are set to the following fields when the Combination-Form is output.
    - Comment when linking to Print Manager
    - Comment when making e-Record
    - Comment output in audit trail log
- Comments are not set unless specified when performing Combination-Form outputting or in the Combination-Form Descriptor.
- Data definition of each form style descriptors in the Combination-Form Descriptor cannot be linked.

### • Cautions for designing a form descriptor information

Note the following points when creating the form descriptor information making up a Combination-Form.

- The following fields must use the same values in all forms to be used in the Combination-Form Descriptor.
  - Paper Type
  - Paper Size
  - Orientation
  - Print Area
  - Print Size
  - Multi-Up Printing
  - Input Data Format
  - Separator Character
  - Block Text Field New Line Codes
  - "Alternative text of Control Code" in the [Barcode] tab of the [Form Properties] window.
  - "Fill with a blank if data length does not reach field length" in the [Barcode] tab of the [Form Properties] window.
- When specifying the variable "%" to the SET operator, the same replacement strings specified during the form output are applied to all forms used in Combination-Form output.
- When outputting a form in multiple pages, PAGE and TPAGE variables in data definition will not work properly. In this case, use PAGE-ALL and TPAGE-ALL variables instead of PAGE and TPAGE variables.
 

Example: The case of outputting 2 forms with total 3 pages as multi-forms is used here as an example. Page number (PAGE variable) and total number of pages (TPAGE variable) in individual forms cannot be output as shown below.

1st page: 1/3

2nd page: 2/3  
 3rd page: 3/3  
 4th page: 1/3  
 5th page: 2/3  
 6th page: 3/3

Page number (PAGE-ALL variable) and total number of pages (TPAGE-ALL variable) through all forms can be output as shown below.

1st page: 1/6  
 2nd page: 2/6  
 3rd page: 3/6  
 4th page: 4/6  
 5th page: 5/6  
 6th page: 6/6

- When outputting a form in multiple pages, TSUM function in data definition will not work properly. There is no alternative method.
- When outputting a free frame format form in multiple pages, if any of the following operators is specified in data definition to be used across pages (for example, a field data in the previous page is used in the next page), the operator will not work properly.

|                    |   |
|--------------------|---|
| Media data         | : XMEDIA-x(y)   |
| String operation   | : HEADSTR(x), TAILSTR(x), CONNECTSTR(x1,x2[,x3...])   |
| Condition judgment | : CASE(x1,y1)[CASE(x2,y2) ..]DEFAULT(z)   |
| Function           | : ADD(x,y), SUB(x,y), MUL(x,y), DIV(x,y), CAL(Equation), SET(x),<br>SUM(x), TSUM(x), BSUM-nn(x), MAX(x), MIN(x), MEAN(x), CNT |
| Internal variable  | : PAGE, TPAGE, BPAGE-nn, BTPAGE-nn  |

- Depending on the output method, there are some points to note when creating a form descriptor information making up a Combination-Form.  
 For details, refer to the Online Manual "Setup & Operation".

- **Cautions regarding output order of form descriptor information**

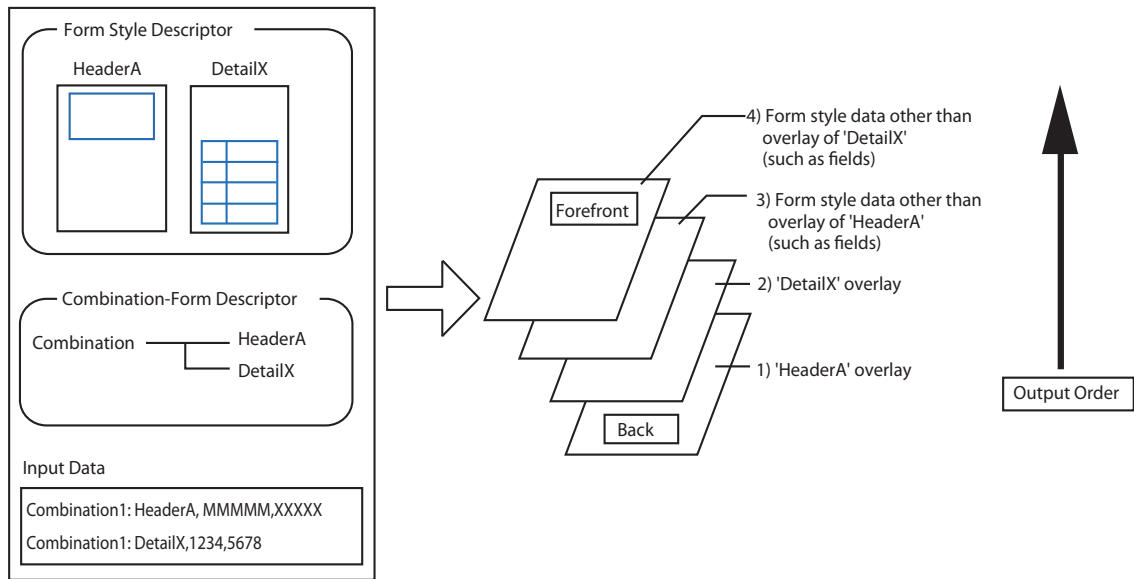
The following explains the output order of form descriptor information used for Combination-Form output.

- Objects in form style information are output in the following order (1 to 2).
  - 1: Overlays
  - 2: Objects other than overlays in form style information (fields etc.)
- To output combined multiple pieces of form descriptor information, the objects are output in the following order:
  - Overlays are output according to the order of the form descriptor information in the Combination-Form descriptor (output from the form descriptor information displayed top of the Combination-Form Descriptor list).
  - Objects other than overlays in the form style information (fields etc.) are output according to the order in which they appear in the input data for the Combination-Form.

In the following example, objects are output in the following order (1 to 2 to 3 to 4).

- 1: Overlays in "HeaderA"
- 2: Overlays in "DetailX"
- 3: Objects other than overlays in form style information in "HeaderA" (fields etc.)
- 4: Objects other than overlays in form style information in "DetailX" (fields etc.)

When overlapped, the object of (4) is output in foreground.




**Caution** - Overlays of all form descriptor information specified in the Combination-Form descriptor are output. Therefore, note that the overlays in the form descriptor information not specified in the form name of the input data are also output.

- Whether to output the overlays is decided according to the specification in the first form in the input data. If "Print overlay" in the [Overlay] tab of the "Form properties" is not checked in the first form in the input data, even if it is checked in the subsequent forms to be combined, the overlays are not output on the pages.

## 10.2 Designing Input Data for Combination-Form


This section explains how to design the input data used in Combination-Form output.

When outputting combination-forms, it is necessary to enter combination-form names and form names in the data file. It is not necessary to define fields that receive the combination-form names and form names in the [Form Style Descriptor] window.

 **Caution** - "Fixed length" input data cannot be used in Combination-Form output.  
 - Data file with the field name label cannot be used in Combination-Form output.

### 10.2.1 When the Input Data Format is "Variable Length" or "CSV Format"

The Combination-Form name and form name are placed at the beginning of each record (line). Next is a separator character to be specified in "Form Name Data Separator Character for Combination-Form" when outputting the form (the same separator character as specified in the [General] tab of the [Data Definitions] window). Then, the data, in the same format as when each form in the Combination-Form is output.

 **Caution** - For all forms in the Combination-Form output, the input data format and separator characters should be the same.  
 - Specify the separator character when the form is output.

The following shows an example of data files to be output for two separate forms, and an example of the data file for Combination-Form output.

- When outputting 2 forms separately

Create 2 data files, and output 2 separate forms.

The input data format is "Variable Length", and the separator character is "Vertical Bar (|)".

Data for "HeaderA" form

Madison Avenue XXX | Des Moines IA | ABC Electronics | 20110810

Data for "DetailX" form

ABC Electronics | 20110810 | 03101 | 005 | 0020 | TV | 3 | 120000  
 ABC Electronics | 20110810 | 03101 | 005 | 0022 | Microwave oven | 1 | 82000  
 ABC Electronics | 20110810 | 03101 | 005 | 0100 | Vacuum Cleaner | 2 | 23000

- To output a Combination-Form of 2 forms

Create a single data file that includes the data for two forms, and output it as a single form.

The input data format is "Variable Length", and the separator character is "Vertical Bar (|)".

Enter the combination-form name and form name separated by a colon (:) at the beginning of each record (line).

Next is the "Vertical Bar (|)" separator character. Then, the data, in the same format as when a single form, is output.

Combination-  
 Form name    Form name    Data

Combination1:HeaderA | Madison Avenue XXX | Des Moines IA | ABC Electronics | 20110810  
 Combination1:DetailX | ABC Electronics | 20110810 | 03101 | 005 | 0020 | TV | 3 | 120000  
 Combination1:DetailX | ABC Electronics | 20110810 | 03101 | 005 | 0022 | Microwave oven | 1 | 82000  
 Combination1:DetailX | ABC Electronics | 20110810 | 03101 | 005 | 0100 | Vacuum cleaner | 2 | 23000

↑  
 Separator Character

## 10.2.2 When the Input Data Format is "XML Format"

When the input data in "XML Format" is output in combination-forms, enclose the XML data in multi-form tags <FjLForm> and specify the Combination-Form names and form names to be output separated by a colon (:) with the attribute "name".



**Caution**

- Specify the multi-form tag <FjLForm> in XML file. You should not use the tag <FjLForm> when defining form style descriptor.
- For all forms in the Combination-Form output, the input data format should be the same.

- When outputting 2 forms separately

Create 2 sets of XML data, and output 2 separate forms.

Data for "HeaderA" form

```
<HeaderA>
  <Address>Madison Avenue XXX Des Moines IA</Address>
  <Customer>ABC Electronics</Customer>
  <BillingDate>20110810</BillingDate>
</HeaderA>
```

Data for "DetailX" form

```
<DetailX>
  <Customer>ABC Electronics</Customer>
  <BillingDate>20110810</BillingDate>
  <SalesRecordNumber>03101</SalesRecordNumber>
  <ContactCode>005</ContactCode>
  <Detail>
    <ProductCode>0020</ProductCode>
    <ProductName>TV</ProductName>
    <Quantity>3</Quantity>
    <UnitPrice>120000</UnitPrice>
  </Detail>
  <Detail>
    <ProductCode>0022</ProductCode>
    <ProductName>Microwave oven</ProductName>
    <Quantity>1</Quantity>
    <UnitPrice>82000</UnitPrice>
  </Detail>
  <Detail>
    <ProductCode>0100</ProductCode>
    <ProductName>Vacuum cleaner</ProductName>
    <Quantity>2</Quantity>
    <UnitPrice>23000</UnitPrice>
  </Detail>
</DetailX>
```



- To output a Combination-Form of 2 forms

Create a single XML data that includes the data for the 2 forms, and output it as a single form. Enclose the XML data in the multi-form tags. Specify the combination-form name and form name separated by a colon (:) with the attribute "name". Enter the data root defined in the Form Style Descriptor for child element of multi-form tag.

Multi-form tags are specified under the XML root.

```
<XMLRoot>
  <FjLCForm name="Combination1:HeaderA">
    <HeaderA>
      <Address>Madison Avenue XXX Des Moines IA</Address>
      <Customer>ABC Electronics</Customer>
      <BillingDate>20110810</BillingDate>
    </HeaderA>
  </FjLCForm>
  <FjLCForm name="Combination1:DetailX">
    <DetailX>
      <Customer>ABC Electronics</Customer>
      <BillingDate>20110810</BillingDate>
      <SalesRecordNumber>03101</SalesRecordNumber>
      <ContactCode>005</ContactCode>
      <Detail>
        <ProductCode>0020</ProductCode>
        <ProductName>TV</ProductName>
        <Quantity>3</Quantity>
        <UnitPrice>120000</UnitPrice>
      </Detail>
      <Detail>
        <ProductCode>0022</ProductCode>
        <ProductName>Microwave oven</ProductName>
        <Quantity>1</Quantity>
        <UnitPrice>82000</UnitPrice>
      </Detail>
      <Detail>
        <ProductCode>0100</ProductCode>
        <ProductName>Vacuum cleaner</ProductName>
        <Quantity>2</Quantity>
        <UnitPrice>23000</UnitPrice>
      </Detail>
    </DetailX>
  </FjLCForm>
</XMLRoot>
```

### 10.2.3 Points

Note the following points when designing input data for combination-form outputs:

- When performing combination-form output, an error may occur if there are two or more consecutive pieces of blank data in the same form (when linefeed processing is specified to perform during form output by linefeed code control, an error will not occur).

The state when there are two or more consecutive pieces of blank data is defined as the state when all of the following conditions are met:

- A form name and a form name data separator character for multi-form are described
- There is no other data after the form name in the record
- There is no succession of records for the same form name

Therefore, to perform combination-form output, it is recommended to design an input data such that two or more consecutive blank lines will not be in the data.

The following shows an example of two or more consecutive blank lines in input data.

Example) When the separator character for form is a vertical bar (|) and input data is in CSV format

```

Combination1:Header1 | 0000000 Madison Avenue Des Moines IA ABC Electronics 20110810
Combination1:Statement | ABC Electronics 20110810 03101 005 0020 TV 3 120000
Combination1:Statement | ABC Electronics 20110810 03101 005 0022 Microwave oven 1 82000
Combination1:Statement | ABC Electronics 20110810 03101 005 0100 Vacuum cleaner 2 23000
Combination2:DetailBill | 0000000 Jefferson Avenue XXX New York NY XYZ Electronics 20110810
Combination2:DetailBill | XYZ Electronics 20110810 03102 005 0100 Vacuum cleaner 2 23000
Combination2:DetailBill |
Combination2:DetailBill |
Combination2:DetailBill | XYZ Electronics 20110810 03102 005 0211 Stove 3 5000
Combination2:DetailBill | XYZ Electronics 20110810 03102 005 0221 Warmer 3 7800
    
```

<- Blank line  
<- Blank line

## 10.2.4 Page Feed Timing

During combination-form output, a new page starts as follows:

- When a record has the same combination-form name and the same form name as a previous record in the input data

New Page →

|                      |  |                    |  |               |  |                 |  |   |
|----------------------|--|--------------------|--|---------------|--|-----------------|--|---|
| Combination1:HeaderA |  | Madison Avenue XXX |  | Des Moines IA |  | ABC Electronics |  | 20110810                                |
| Combination1:DetailX |  | ABC Electronics    |  | 20110810      |  | 03101           |  | 005   0020   TV   3   120000            |
| Combination1:DetailX |  | ABC Electronics    |  | 20110810      |  | 03101           |  | 005   0022   Microwave oven   1   82000 |
| Combination1:DetailX |  | ABC Electronics    |  | 20110810      |  | 03101           |  | 005   0100   Vacuum cleaner   2   23000 |
| Combination1:HeaderA |  | Madison Avenue XXX |  | Des Moines IA |  | ABC Electronics |  | 20110910                                |
| Combination1:DetailX |  | ABC Electronics    |  | 20110910      |  | 03102           |  | 005   0020   TV   5   120000            |
| Combination1:DetailX |  | ABC Electronics    |  | 20110910      |  | 03102           |  | 005   0200   Dishwasher   1   82000     |
| Combination1:DetailX |  | ABC Electronics    |  | 20110910      |  | 03102           |  | 005   0300   Refrigerator   3   23000   |



**Note**

- If the form format is free frame or simple table format, new page is not started even if the detail rows are repeated consecutively.

- In the input data, when the Combination-Form name is changed

New Page →

|                      |  |                      |  |               |  |                 |  |   |
|----------------------|--|----------------------|--|---------------|--|-----------------|--|---|
| Combination1:HeaderA |  | Madison Avenue XXX   |  | Des Moines IA |  | ABC Electronics |  | 20110810                                |
| Combination1:DetailX |  | ABC Electronics      |  | 20110810      |  | 03101           |  | 005   0020   TV   3   120000            |
| Combination1:DetailX |  | ABC Electronics      |  | 20110810      |  | 03101           |  | 005   0022   Microwave oven   1   82000 |
| Combination1:DetailX |  | ABC Electronics      |  | 20110810      |  | 03101           |  | 005   0100   Vacuum cleaner   2   23000 |
| Combination2:HeaderB |  | Jefferson Avenue XXX |  | New York NY   |  | XYZ Electronics |  | 20110810                                |
| Combination2:DetailY |  | XYZ Electronics      |  | 20110810      |  | 03102           |  | 005   0100   Vacuum cleaner   2   23000 |
| Combination2:DetailY |  | XYZ Electronics      |  | 20110810      |  | 03102           |  | 005   0110   Refrigerator   2   125000  |
| Combination2:DetailY |  | XYZ Electronics      |  | 20110810      |  | 03102           |  | 005   0201   Carpet   5   10000         |



**Caution**

- If the form format is free frame or simple table format, when the number of repeats for the details exceeds the number of the details for one page, or when a BREAK occurs, a new page starts but not output in the combination-form correctly. In order to make sure that new page is not started in the above mentioned timing, specify the detail rows according to the page start position, and insert the header line in the appropriate position if required.

# **Chapter 11**

## *Replacing Properties when Outputting Forms*

---

This chapter explains the overview of property replacement, how to use this function, and details of the property replace file.

## 11.1 What is "Replacing Properties"?

Replacing properties is to replace property information (fonts, strings, etc.) of fields defined in the form descriptor information with other information and output it when a form is output.

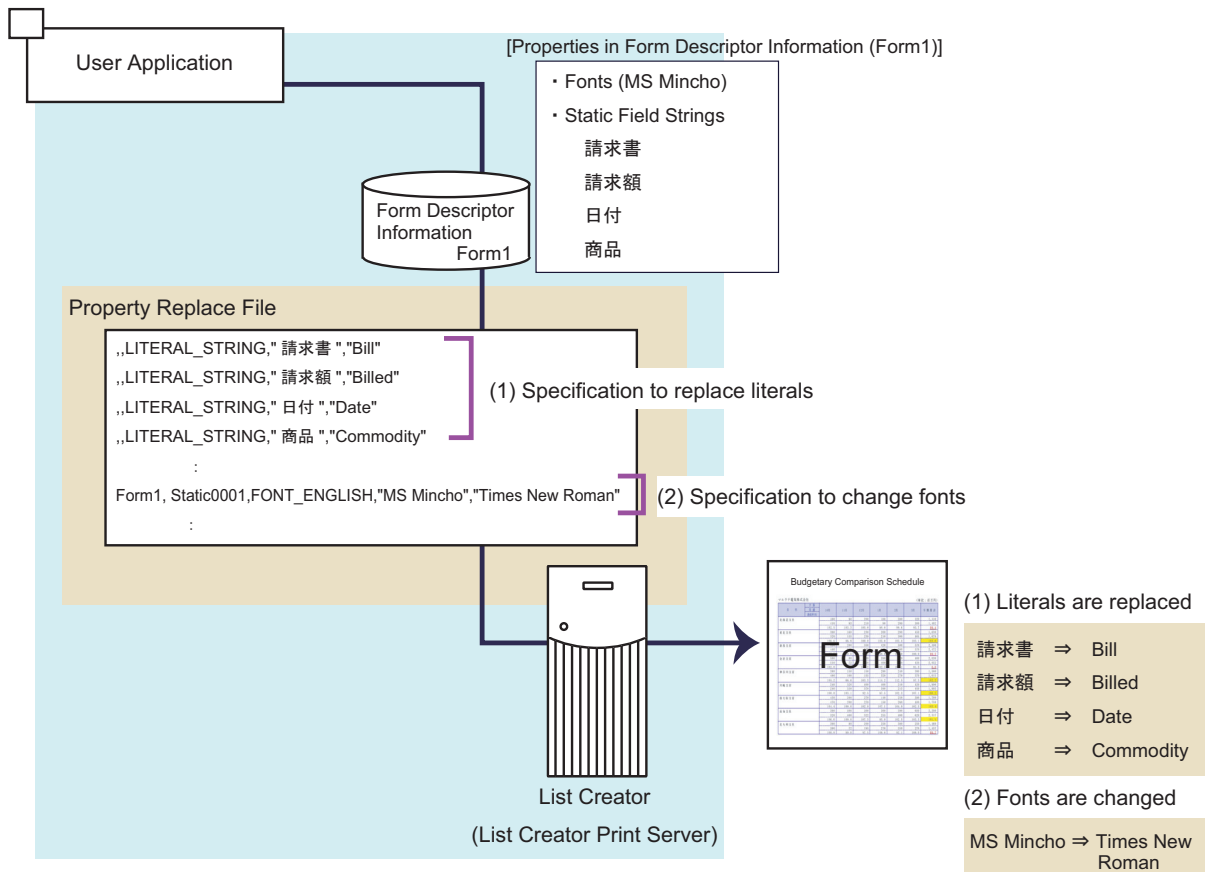
For example, a Japanese font defined as the field's property can be changed to another type of font when the form is output.

This is called the "property replacement function".

With the property replacement function, you can change the output result of a form without having to modify the form descriptor information.

To replace properties, the field property information you want to replace should be defined in a "property replace file".

Specify the file when outputting the form to replace the relevant properties.



The following explains the properties that can be replaced during form output, and the points to note when replacing properties.

When using the property replacement function, make sure that you note the following information.

### 11.1.1 Properties that Can Be Replaced

The following shows the field properties that can be replaced when a form is output.

| Property Type | Corresponding Property |  | Corresponding Fields   |  |
|---------------|------------------------|--|--|--|
| Font          | Japanese Font          | [Japanese Font] of the [Font] tab                                      | - Text field<br>- Block text field<br>- Radio button<br>- Check box<br>- Static field<br>- Block static field<br>- Date field<br>- Time field                    |  |
|               | Font                   | [Font] of the [Font] tab   | - Numeric field<br>- Text field<br>- Block text field<br>- Radio button<br>- Check box<br>- Static field<br>- Block static field<br>- Date field<br>- Time field |  |
| Literal       | String                 | [String] of the [Properties] tab                                       | - Static field<br>- Date field<br>- Time field   |  |
|               | Strings (Multi line)   | [String] of the [Properties] tab                                       | - Block static field   |  |
|               | Label                  | [Label] of the [Properties] tab  | - Radio button<br>- Check box  |  |
| Numeric       | Decimal Format         | [Decimal Format] in the [Form Descriptor] tab of [Form Properties>(*1) | - Numeric field (Only when "Value" or "Currency" is selected for the formatting type)  |  |
|               | Decimal Places         | [Decimal Places] of the [Properties] tab                               |  |  |
|               | Decimal Numbers        | Output Decimal Point: On / Off   |  | [Suppress any Decimal Places] of the [Formatting] tab(*2)    |
|               |                        | Format the Decimal Places  |  | [Format the Decimal Places] of the [Formatting] tab(*2) (*3) |
| Currency      | Currency Symbol        | [Currency Symbol] of the [Formatting] tab                              | - Numeric field (Only when "Currency" is selected for the formatting type)   |  |
| Date          | Date Edit Strings      | [String] of the [Formatting] tab                                       | - Numeric field (Only when "Date" is selected for the formatting type)   |  |

\*1: [Decimal Format] is a property applied globally to the entire form descriptor information and cannot be specified on a per-field basis.

\*2: This applies only when at least 1 is specified in [Decimal Places] of the [Properties] tab.

\*3: This applies only to fields to which [Format the Decimal Places] is specified on the [Formatting] tab.

## 11.1.2 Points to Note when Replacing Properties

When using the property replacement function, you must consider the target where properties are to be replaced. Determine which properties should be replaced in accordance with your specific way of using the forms. Also, you must take account of any changes to the target fields, such as the field size, caused by replacing the fields during form outputting.

This section explains the following two points:

- [Points for creating a property replace file](#)
- [Points for creating form descriptor information](#)

- **Points for creating a property replace file**

The following explains the target of property replacement and how to determine which type of property replace file to create.

### Target of property replacement

In the property replace file, specify either of the following as the target of property replacement:

- Target all forms  
If you do not specify any form name, the definitions will be "common" to all the forms.
- Target specific form(s) only  
If you specify a specific form name (and a field name), the definitions apply only to the "specified form".

You can also use both ways shown above at the same time.

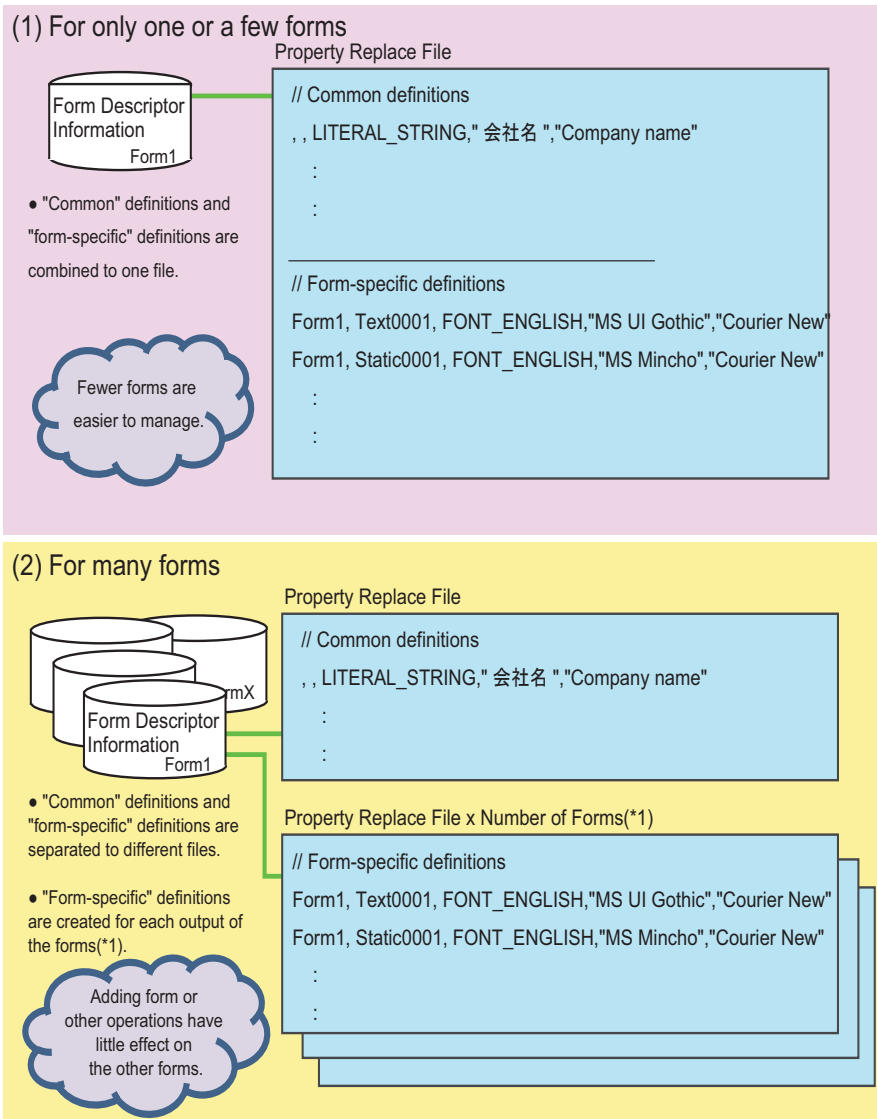
For the priority of overlapping definitions resulting from specifying in both ways above at the same time, refer to the following:

"•Priority" in => ["11.3 Property Replace File"](#)

Determining which type of property replace file to create

Create property replace files in accordance with your specific way of using the forms.

Either of the following cases applies when determining which type of property replace file to create:



\*1: For multi-form/Combination-Form output, create form-specific definitions separately for each output unit of the forms. (The definitions specific to the forms to combine must be specified in a single file.)

• Points for creating form descriptor information

The following explains the points to note for creating form descriptor information when replacing properties and outputting a form.

- When you define form descriptor information as shown below, character alignment is adjusted so that the characters fit into the field size. This way, you can output forms without having to adjust the string length and the field length after the changes are applied.

For example, you can prevent a string from being truncated even if the applied change makes it longer. Otherwise, part of the string does not fit into the field length and may be truncated.

- [Alignment] tab in the field's properties window
  - Specify [Write closely, Output all data] for [Do if the data does not fit]
- [Block Layout] tab in the field's properties window
  - Specify [Decrease spacing to fit] for [Ensure all contents are shown]





**Note**

- If the above settings are specified, combining characters will not be output correctly. When using combining characters, do not specify these settings. Instead, adjust the string length and field length to fit the combined characters.

- When outputting multilingual replace information, it may contain Unicode characters. Therefore, check [Use Unicode] in the [Form Descriptor] tab of the [Form Properties] window for the form descriptor information.

## 11.2 Procedure for Using "Property Replacement"

This section explains the procedure for creating a form in which properties are replaced during form output.

- 1) Preparing for Form Designing  
Design a specification document for your desired form descriptor information and create the layout for form output.
- 2) Creating Form Descriptor Information  
Create the base form descriptor information in accordance with the design.  
=> "[Chapter 2 Form Designing Flow](#)"
- 3) Creating a Format Property Replace File  
Create a format property replace file.  
=> "[11.2.1 Creating Format Property Replace File](#)"
- 4) Creating a Property Replace File  
Edit the format file created in Step 3) and create a property replace file.  
=> "[11.2.2 Creating Property Replace File](#)"
- 5) Checking Output Image of Form  
Check the resulting output image that contains the properties replaced by using the property replace file.  
=> "[11.2.3 Checking Output Image of Form](#)"
- 6) Adjusting Form Descriptor Information (Form Layout Data) and Correcting Property Replace File  
If the form output image is not what you intend, correct the form descriptor information and the property replace file, and check the form output image again. Repeat this adjustment step until you are satisfied with the result.



**Note**

- To design and use forms in a non-Japanese environment, refer to the descriptions on multi-language form operation in the Online Manual "Setup & Operation".

- Specifying a property replace file

You can replace and output properties in a form by specifying a property replace file when outputting the form. For details on specifications when outputting a form, refer to the Online Manual "Application Design".



**Note**

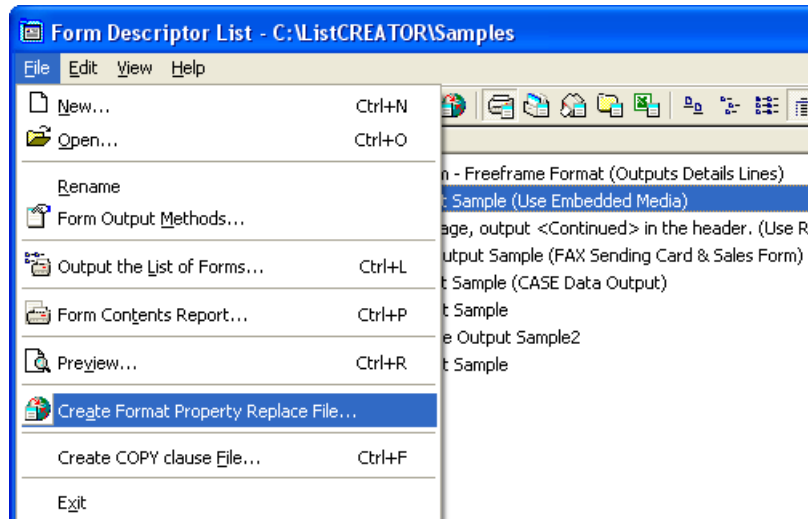
- When using property replacement function during multi-form output, it is recommended to specify "Performance mode for multi-form output" with the setup command for List Creator. For details, refer to the description about the List Creator setup command used for specifying "Performance mode for multi-form output" in the Online Manual "Setup & Operation".

## 11.2.1 Creating Format Property Replace File

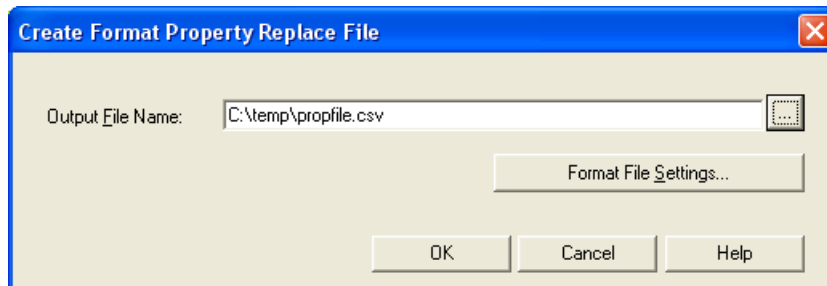
The following shows the procedure for creating a format property replace file.

### = Procedure =

- 1) Select [Start] - [Programs] - [List Creator] - [Form Descriptor List].  
=>The [Form Descriptor List] window is displayed.
- 2) In the [Form Descriptor List] window, select a form for which you want to create a format file. Then select [File] - [Create Format Property Replace File].



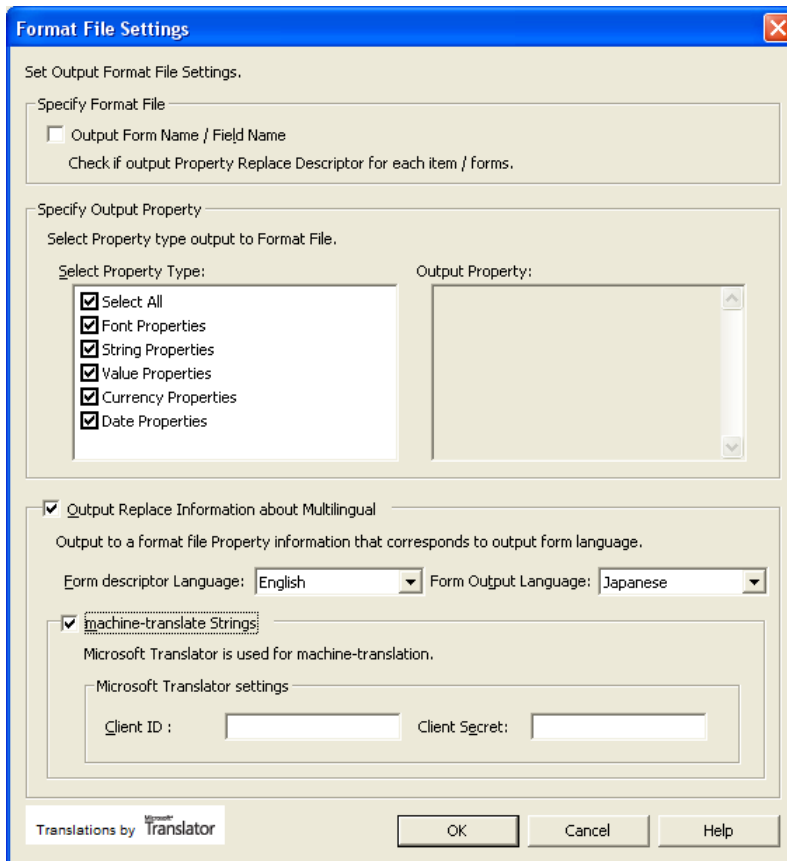
=>The [Create Format Property Replace File] window is displayed.



### Note

- For details of the [Create Format Property Replace File] window, refer to the Designer Help guide.
- You can select multiple forms in the [Form Descriptor List] window and create a format file.  
If multiple forms are selected, the property information from all the selected forms is combined to create a single format file.

- 3) To specify details of the format file, click the [Format File Settings] button.  
If you do not specify the detailed settings of the format file, proceed to Step 4).  
=>The [Format File Settings] window is displayed.



Specify any properties that will be replaced for different forms or fields, any language-specific property information (such as machine-translation of strings), or other details.



**Note**

- For details of the [Format File Settings] window, refer to the Designer Help guide.
- Microsoft Translator is used for machine-translation of strings. You should have completed necessary procedures for using Microsoft Translator. For details, refer to the following:  
=> "[Procedure for using Microsoft Translator](#)"
- To perform machine-translation, the system will connect to the Microsoft Translator service using the Internet Explorer settings of the logged on user. Before performing this function, set up the Internet Explorer and make sure it can access the Internet.

4) Specify the output destination of the format file and click the [OK] button.  
=>A format file is created.



**Note**

- For the description format of the format file, refer to the following:  
=> "[11.3.2 Format of Format Property Replace File](#)"

- Procedure for using Microsoft Translator


Microsoft Translator is used for machine-translation of strings.

Microsoft Translator is a machine-translation service published on the Windows Azure Marketplace.

To use Microsoft Translator, "Client ID" and "Client Secret" are required.


The client ID and the client secret are obtained when you register your application on Windows Azure Marketplace.

The following is the procedure to register the application with Windows Azure Marketplace.


|   |   |
|---|---|
| <br><b>Caution</b> | <ul style="list-style-type: none"><li>- When using Microsoft Translator for machine-translation, strings to be translated (text strings defined in the form descriptor information) will be sent to Microsoft Translator as is. Check the strings before translation.<br/>Strings can be checked by creating a format file to disable machine-translation.<br/>To create a format file to disable machine-translation, set the followings in the "Format File Settings" Window.<ul style="list-style-type: none"><li>- Check "String" for [Select Property Type].</li><li>- Uncheck "machine-translate Strings".</li></ul></li><li>- For more information on the terms of use of Windows Azure Marketplace, refer to the [Terms of Use] on the Windows Azure Marketplace web site.</li><li>- For more information on the terms of use of Microsoft Translator, refer to the [Publisher Offer Terms] on the Microsoft Translator web site.</li></ul> |
|---|---|

**= Procedure =**


- 1) Sign up for the Windows Azure Marketplace with your Windows Live ID, and register a Windows Azure Marketplace account.

|  |   |
|--|---|
| <br><b>Note</b> | <ul style="list-style-type: none"><li>- You must register an account even when you use the free plan of Microsoft Translator.</li></ul> |
|--|---|

- 2) Sign up for Microsoft Translator.  
Select a subscription plan of Microsoft Translator and enter your name and e-mail address.  
=>You are signed up for Microsoft Translator.

|  |  |
|--|--|
| <br><b>Note</b> | <ul style="list-style-type: none"><li>- Select a subscription plan suitable to the amount of translation. It is recommended to select a "free subscription".</li></ul> |
|--|--|

- 3) Register the application with the Windows Azure Marketplace.  
Specify the following registration data:
  - Client ID  
Specify any string that can be recognized (for example, "ListCreatorXXXXX").  
This string is the "Client ID" to be specified in the [Format File Settings] window.
  - Application name  
Specify any string (for example, "ListCreator").
  - Redirected URI  
Specify any URI.

|   |   |
|---|---|
| <br><b>Caution</b> | <ul style="list-style-type: none"><li>- "Client Secret", displayed automatically in the application registration window of the Windows Azure Marketplace, will be the "Client Secret" to be specified in the [Format File Settings] window. Make sure that you take a note of it.</li></ul> |
|---|---|

## 11.2.2 Creating Property Replace File

■ Create a property replace file from the format property replace file.  
For properties replaced when a form is output, specify the information to be replaced with.  
For properties not replaced when a form is output, delete the corresponding lines.  
For details, refer to the following:  
[=> "11.3 Property Replace File"](#)

## 11.2.3 Checking Output Image of Form

■ Use the preview function to check the output image that contains the properties replaced by using the property replace file.

The following methods are available for checking with the preview function:

- Checking in the [Preview Start] window
  - Previewing from the [Form Descriptor List] window
  - Previewing from the [File] menu of the [Form Style Descriptor] window
- Checking with Preview
  - Previewing with Utility (Preview)
- **Checking in the [Preview Start] window**

Specify a property replace file in [Property Replace File] in the [Preview Start] window.  
For details regarding the [Preview Start] window, refer to the Designer Help guide.
- **Checking with Preview**

Specify a property replace file in [Property Replace File] of the [Details] tab in the [Preview] window.  
For details regarding the [Preview] window, refer to the Online Help guide.

## 11.3 Property Replace File

This section explains the details of the property replace file.

- **Definition of property replace file**

A property replace file is a file that defines property information (fonts, strings, etc.) to be replaced and output when a form is output.

The target of property replacement varies depending on the definitions in the property replace file, as shown below.

- Target all forms  
If you do not specify any form name, the definitions will be "common" to all the forms.
- Target specific form(s) only  
If you specify a specific form name (and a field name), the definitions apply only to the "specified form".

- **Character code**

Use the following character codes to create a property replace file:

- When the List Creator print server runs on Windows  
UNICODE (UTF8) character code, with the linefeed code of CR+LF (0x0d+0x0a)
- When the List Creator print server runs on Solaris/Linux  
UNICODE (UTF8) character code, with the linefeed code of LF (0x0a)

- **Priority**

In the following cases, definitions in property replace files may overlap:

- When a property replace file "common" to all forms and another file that only applies to a specific form are specified at the same time
- When the definitions in a property replace file that only applies to a specific form are combined to a property replace file that is "common" to all forms

If definitions in property replace files overlap as shown above, the following priority applies:


- 1) Specification with a form/field name
- 2) Specification with a form name
- 3) Specification with a property identifier


### 11.3.1 Description Format of Property Replace File

The following explains the description format of the property replace file.

- Description format

[Form\_name],[Field\_name],Property\_identifier,"Information\_before\_replacement","Information\_after\_replacement"

 **Caution** - Do not place any unnecessary spaces before or after a comma (,) or between a comma (,) and specified information. As spaces are regarded as specified characters, it is possible that specified values do not match the information defined in the form style descriptor, causing a failure in replacing the target properties.


 **Note** - A "comma (,)" is the separator character. This specification cannot be omitted.  
- "/" at the start of a line indicates a comment line.

[Form\_name]/[Field\_name]

Specify form and field names for which properties are to be replaced.

The way this specification should be made differs depending on the target of the property replacement.


- When properties are replaced in all forms  
Omit both [Form\_name] and [Field\_name].
- When properties are replaced in a specific form  
Specify the target form name in [Form\_name].
- When properties are replaced in specific fields of a specific form  
Specify [Form\_name] and [Field\_name].

 **Caution**

- When specifying [Field\_name], make sure that you also specify [Form\_name]. Specifying [Field\_name] without [Form\_name] is invalid.
- When specifying [Field\_name], make sure that you specify the correct uppercase and lowercase letters as defined in the form descriptor information.
- [Decimal Format] (Property identifier: "NUM\_DECIMAL\_FORMAT") is a property applied globally to the entire form descriptor information and cannot be specified on a per-field basis. Therefore, do not specify [Field\_name].
- For more information on whether to specify items for each target of property replacement, refer to the following:  
=> "[Specification example](#)"

Property\_identifier

Specify the property identifier of a property to be replaced when outputting the form.

 **Caution** - This must be specified in uppercase letters. The property identifiers containing any non-uppercase letters is invalid.

The property identifiers that can be specified are shown below.

| Property Type | Property Identifier | Description                              |
|---------------|---------------------|--|
| Font          | FONT_JAPANESE       | Identifier for replacing Japanese fonts. |
|               | FONT_ENGLISH        | Identifier for replacing English fonts.  |




| Property Type | Property Identifier         | Description  |
|---------------|-----------------------------|--|
| Literal       | LITERAL_STRING              | Identifier for replacing strings in static fields, date fields, or time fields.                      |
|               | LITERAL_RECTSTRING          | Identifier for replacing strings in block static fields.   |
|               | LITERAL_LABEL               | Identifier for replacing label strings of radio buttons or check boxes.                              |
| Value         | NUM_DECIMAL_FORMAT          | Identifier for replacing the decimal format style.   |
|               | NUM_DECIMAL_PLACES          | Identifier for replacing the specification of the number of decimal places in numeric fields.        |
|               | NUM_DECIMAL_PLACES_SUPPRESS | Identifier for replacing the specification of whether to output the decimal point in numeric fields. |
|               | NUM_DECIMAL_PLACES_FORMAT   | Identifier for replacing the specification of the format of decimal places in numeric fields.        |
| Currency      | CURRENCY_SYMBOL             | Identifier for replacing the specification of the currency symbol in numeric fields.                 |
| Date          | DATE_FORMAT_STRING          | Identifier for replacing the date edit format in numeric fields.                                     |

Information\_before\_replacement/Information\_after\_replacement

- Information\_before\_replacement  
Specify the value of a property specified in the form descriptor information (information before replacement).
- Information\_after\_replacement  
Specify the substituting value of the property used when the form is output (information after replacement).  
If this is omitted, the property is output as specified in the form descriptor information (without being replaced).

For details about specification for each type, refer to the following:

| Property Type | For specification methods, refer to:   |
|---------------|--|
| Font          | =>Specification method when Property_identifier is "FONT_JAPANESE" or "FONT_ENGLISH"         |
| Literal       | =>When Property_identifier is "LITERAL_STRING" (String)                                      |
|               | =>When Property_identifier is "LITERAL_RECTSTRING" (Strings (Multi line))                    |
|               | =>When Property_identifier is "LITERAL_LABEL" (Label)  |
| Value         | =>When Property_identifier is "NUM_DECIMAL_FORMAT" (Decimal Format)                          |
|               | =>When Property_identifier is "NUM_DECIMAL_PLACES" (Decimal Places)                          |
|               | =>When Property_identifier is "NUM_DECIMAL_PLACES_SUPPRESS" (Output Decimal Point: On / Off) |
|               | =>When Property_identifier is "NUM_DECIMAL_PLACES_FORMAT" (Format the Decimal Places)        |
| Currency      | =>When Property_identifier is "CURRENCY_SYMBOL"  |
| Date          | =>When Property_identifier is "DATE_FORMAT_STRING"   |

|  |  |
|--|--|
| <br><b>Note</b> | <ul style="list-style-type: none"> <li>- If the same property identifier and the same information before replacement is specified more than once, the last specified value will be valid.</li> <li>- The specification in the line is invalid in the following cases:             <ul style="list-style-type: none"> <li>• If a form name that does not exist is specified</li> <li>• If [Field_name] is specified without [Form_name]</li> <li>• If a field name that is not defined is specified</li> <li>• When specifying a string property, if the field of the string (or label) set in "Information_before_replacement" does not exist in the form descriptor information</li> <li>• If a value is specified incorrectly (for example, if the Property_identifier is not specified in uppercase letters)</li> </ul> </li> </ul> |
|--|--|

- Specification method when Property\_identifier is "FONT\_JAPANESE" or "FONT\_ENGLISH"

Information\_before\_replacement

Specify a Japanese or English font name defined in the form descriptor information within 31 bytes (as counted in Shift-JIS).  
The value must be enclosed with double quotation marks (").

Information\_after\_replacement

Specify a font name that will replace the current one within 31 bytes (as counted in Shift-JIS).  
The value must be enclosed with double quotation marks (").

- Specification method when Property\_identifier is "LITERAL\_STRING", "LITERAL\_RECTSTRING" or "LITERAL\_LABEL"

When Property\_identifier is "LITERAL\_STRING" (String)


Information\_before\_replacement


Specify a string defined in the form descriptor information.  
The value must be enclosed with double quotation marks (").

Information\_after\_replacement

Specify a string that will replace the current one.  
The value must be enclosed with double quotation marks (").

- When [Use Unicode] is not checked in the [Form Descriptor] tab of the [Form Properties] window for the form descriptor information  
Specify in a range from 1 to 999 bytes.  
Only characters within the range of Shift-JIS coding can be specified.
- When [Use Unicode] is checked in the [Form Descriptor] tab of the [Form Properties] window for the form descriptor information  
Specify in a range from 1 to 499 characters.  
Characters within the range of Unicode coding can be specified.  
For characters represented by a surrogate pair, one character should be counted as two characters.

|   |  |
|---|--|
| <br><b>Caution</b> | <ul style="list-style-type: none"> <li>- When [Form_name] and [Field_name] are specified and repeats are specified for static fields, date fields, and time fields, only the repeat source field is replaced.<br/>If you want all the strings in static fields, date fields, and time fields for which repeats are specified to be replaced in the form descriptor information, do not specify [Field_name] and specify only [Form_name].</li> </ul> |
|---|--|

|  |  |
|--|--|
| <br><b>Note</b> | <ul style="list-style-type: none"> <li>- To delete a string of characters specified in the form descriptor information and output them in a form as empty characters, specify "".<br/>Example) Deleting "To" specified in the fields when outputting a form<br/>„LITERAL_STRING,"To",""</li> </ul> |
|--|--|

When Property\_identifier is "LITERAL\_RECTSTRING" (Strings (Multi line))

Information\_before\_replacement

Specify a string defined in the form descriptor information.  
The value must be enclosed with double quotation marks (").

Information\_after\_replacement

Specify a string that will replace the current one.  
The value must be enclosed with double quotation marks (").

- When [Use Unicode] is not checked in the [Form Descriptor] tab of the [Form Properties] window for the form descriptor information  
Specify in a range from 1 to 999 bytes.  
Only characters within the range of Shift-JIS coding can be specified.
- When [Use Unicode] is checked in the [Form Descriptor] tab of the [Form Properties] window for the form descriptor information  
Specify in a range from 1 to 499 characters.  
Characters within the range of Unicode coding can be specified.  
For characters represented by a surrogate pair, one character should be counted as two characters.



**Note**

- To insert a linefeed in the middle of a string, specify "¥n" at the position at which you want to start a new line.
- When using machine-translation, "¥n" in the strings may not be translated correctly, or the new lines may start at unintended positions. Adjust the positions of "¥n" in advance so that the new lines in the translated strings start at which you want.
- To include "¥" in a string, specify as "¥¥".  
Example) To output "¥n" as characters, specify as "¥¥n".
- To delete a string of characters specified in the form descriptor information and output them in a form as empty characters, specify "".

Example) Deleting "Thank you for your purchase. ¥nHere is the invoice for your order." specified in the fields when outputting a form

„LITERAL\_RECTSTRING,"Thank you for your purchase. ¥nHere is the invoice for your order.",""

When Property\_identifier is "LITERAL\_LABEL" (Label)

Information\_before\_replacement

Specify a label string defined in the form descriptor information.  
The value must be enclosed with double quotation marks (").

Information\_after\_replacement

Specify a label string that will replace the current one.  
The value must be enclosed with double quotation marks (").

- When [Use Unicode] is not checked in the [Form Descriptor] tab of the [Form Properties] window for the form descriptor information  
Specify in a range from 1 to 255 bytes.  
Only characters within the range of Shift-JIS coding can be specified.
- When [Use Unicode] is checked in the [Form Descriptor] tab of the [Form Properties] window for the form descriptor information  
Specify in a range from 1 to 127 characters.  
Characters within the range of Unicode coding can be specified.  
For characters represented by a surrogate pair, one character should be counted as two characters.



**Note**

- To delete a string of characters specified in the form descriptor information and output them in a form as empty characters, specify "".  
Example) Deleting "Other" specified in the fields when outputting a form  
„LITERAL\_LABEL,"Other",""

- Specification method when Property\_identifier is "NUM\_DECIMAL\_FORMAT", "NUM\_DECIMAL\_PLACES", "NUM\_DECIMAL\_PLACES\_SUPPRESS" or "NUM\_DECIMAL\_PLACES\_FORMAT"



**Note**

- For a property identifier whose type is "Value", "Information\_before\_replacement" and "Information\_after\_replacement" do not need to be enclosed with double quotation marks ("").

When Property\_identifier is "NUM\_DECIMAL\_FORMAT" (Decimal Format)

Information\_before\_replacement

Specify the decimal format style defined in the form descriptor information with either of the following:

- 0: Specify this when the separator character for every three digits is a ", (comma)" and the decimal point is a ". (period)".
- 1: Specify this when the separator character for every three digits is a ". (period)" and the decimal point is a ", (comma)".

Information\_after\_replacement

Specify the decimal format style that will replace the current one with either of the following:

- 0: Specify this when setting the separator character for every three digits to a ", (comma)" and the decimal point to a ". (period)".
- 1: Specify this when setting the separator character for every three digits to a ". (period)" and the decimal point to a ", (comma)".

When Property\_identifier is "NUM\_DECIMAL\_PLACES" (Decimal Places)

Information\_before\_replacement

Specify the number of decimal places defined in the form descriptor information.

Information\_after\_replacement

Specify the number of decimal places that will replace the current one in a range from 0 to the data length.

When Property\_identifier is "NUM\_DECIMAL\_PLACES\_SUPPRESS" (Output Decimal Point: On / Off)

Information\_before\_replacement

Specify the setting of whether to output the decimal point defined in the form descriptor information with either of the following:

- 0: Specify this when the decimal point is to be output.
- 1: Specify this when the decimal point is not to be output.

Information\_after\_replacement

Specify the setting of whether to output the decimal point that will replace the current one with either of the following:

- 0: Specify this to output the decimal point.
- 1: Specify this to not output the decimal point.

When Property\_identifier is "NUM\_DECIMAL\_PLACES\_FORMAT" (Format the Decimal Places)

Information\_before\_replacement

Specify the format of decimal places defined in the form descriptor information with either of the following:

- 0: Specify this when the format of decimal places is "999".
- 1: Specify this when the format of decimal places is "ZZZ".
- 2: Specify this when the format of decimal places is "9ZZ".

Information\_after\_replacement

Specify the format of decimal places that will replace the current one with either of the following:

- 0: Specify this to set the format of decimal places to "999".
- 1: Specify this to set the format of decimal places to "ZZZ".
- 2: Specify this to set the format of decimal places to "9ZZ".

• When Property\_identifier is "CURRENCY\_SYMBOL"


Information\_before\_replacement

Specify the currency symbol defined in the form descriptor information.  
The value must be enclosed with double quotation marks ("").

Information\_after\_replacement

Specify the currency symbol that will replace the current one.  
The value must be enclosed with double quotation marks ("").

- When [Use Unicode] is not checked in the [Form Descriptor] tab of the [Form Properties] window for the form descriptor information  
Specify in a range from 1 to 4 bytes.  
Only characters within the range of Shift-JIS coding can be specified.
- When [Use Unicode] is checked in the [Form Descriptor] tab of the [Form Properties] window for the form descriptor information  
Specify in a range from 1 to 4 characters.  
Characters within the range of Unicode coding can be specified.  
For characters represented by a surrogate pair, one character should be counted as two characters.

 **Caution** - The following characters cannot be used for a currency symbol:  
 Numeric: 0 to 9  
 Alphabet: A to D, N, P, R, S, V, X, Z  
 Symbol: Half-width space, "\*", "+", "-", ";", ":", ",", "(", ")", "''", "/", "=",

• When Property\_identifier is "DATE\_FORMAT\_STRING"

Information\_before\_replacement

Specify the date edit string defined in the form descriptor information.  
The value must be enclosed with double quotation marks ("").

Information\_after\_replacement

Specify a string containing a "special string representing dates" to set the date edit string that will replace the current one.  
The value must be enclosed with double quotation marks ("").

"Special strings representing dates" that can be specified are shown below.

- YYYY : Western calendar year (Example: "2012")
- YY : Last two digits of the Western calendar year (Example: "12")
- MMMM : English month (Example: "April")
- MMM : Abbreviated English month (Example: "Apr")
- MM : Month (Example: "04")
- DD : Day of the month (Example: "02")
- WWWW : English day of the week (Example: "Monday")
- WWW : Abbreviated English day of the week (Example: "Mon")

- When [Use Unicode] is not checked in the [Form Descriptor] tab of the [Form Properties] window for the form descriptor information  
Specify in a range from 1 to 128 bytes.  
Only characters within the range of Shift-JIS coding can be specified.
- When [Use Unicode] is checked in the [Form Descriptor] tab of the [Form Properties] window for the form descriptor information  
Specify in a range from 1 to 64 characters.  
Characters within the range of Unicode coding can be specified.

For characters represented by a surrogate pair, one character should be counted as two characters.

- Specification example

The following examples show specifications for when targeting all forms and when targeting a specific form.

[Specification Example for when Targeting All Forms]

```
//Common definitions
,,LITERAL_STRING," 請求書 ","Bill"
,,LITERAL_STRING," 請求額 ","Billed"
,,LITERAL_STRING," 日付 ","Date"
,,LITERAL_STRING," 商品 ","Commodity"
,,LITERAL_STRING," 数量 ","Amount"
,,LITERAL_STRING," 単価 ","Unit price"
,,LITERAL_STRING," 金額 ","Price"
,,LITERAL_STRING," 合計 ","Total"
```

[Specification Example for when Targeting a Specific Form]

```
//Definitions specific to a specific form
Form1,,LITERAL_STRING," ○○○株式会社 "," ○○○ Limited"
Form1,,LITERAL_STRING," 東京都港区東新橋 X-X-X","Tokyo Minato Ward east Shinbashi X-X-X"
Form1,,LITERAL_STRING," 続く ","Continues"
} (1)

Form1,UPrice,NUM_DECIMAL_PLACES,0,2
Form1,UPrice,NUM_DECIMAL_PLACES_FORMAT,1,2
} (2)

Form1,Price,NUM_DECIMAL_PLACES,2,2
Form1,Price,NUM_DECIMAL_PLACES_FORMAT,1,2
} (2)

Form1>Total,NUM_DECIMAL_PLACES,2,2
Form1>Total,NUM_DECIMAL_PLACES_FORMAT,1,2
} (2)
```


- (1): As [Form\_name] is specified and [Field\_name] is omitted, these specifications apply to the specified form descriptor information.
- (2): As [Form\_name] and [Field\_name] are specified, these specifications apply to the specific fields of the specified form descriptor information.

### 11.3.2 Format of Format Property Replace File

The following explains the format of the format property replace file.  
Here, the following cases are explained separately:

- [When targeting all forms](#)
- [When targeting specific form\(s\) only](#)

- When targeting all forms

 **Note** - Each of the property information defined in the form descriptor information will be output. For overlapping property information, only one piece of it will be output. However, the property identifiers shown below will be output only once, regardless of whether property information is overlapping.

- Property identifier "NUM\_DECIMAL\_FORMAT" for the decimal format for numerical data
- Property identifier "NUM\_DECIMAL\_PLACES\_SUPPRESS" for whether to output the decimal point
- Property identifier "NUM\_DECIMAL\_PLACES\_FORMAT" for the decimal places format

[Description Format of Format File]

```

,,FONT_JAPANESE,"Specified_Japanese_font_name",
,,FONT_ENGLISH,"Specified_English_font_name",
,,LITERAL_STRING,"Specified_character_string",
,,LITERAL_RECTSTRING,"Specified_character_string",
,,LITERAL_LABEL,"Specified_label",
,,CURRENCY_SYMBOL,"Specified_currency_symbol",
,,DATE_FORMAT_STRING,<Specified_date_edit_format>,
,,NUM_DECIMAL_FORMAT,0|1(Specified_decimal_format_style),
,,NUM_DECIMAL_PLACES,<Specified_number_of_decimal_places>,
,,NUM_DECIMAL_PLACES_SUPPRESS,0|1(Specified_setting_of_whether_to_output_the_decimal_point),
,,NUM_DECIMAL_PLACES_FORMAT,0|1|2(Specified_decimal_places_format),
-----
(*1)                (*2)                (*3)                (*4)
    
```

- \*1: [Form\_name] and [Field\_name] will be empty (not defined).
- \*2: Each property identifier is output.
- \*3: Information before replacement (information defined in the form descriptor information) is output.
- \*4: Information after replacement is output. What will be output varies depending on whether to output language information, as shown below.
  - When language information is not output  
The output will be empty (not defined).
  - When language information is output  
Information after replacement is output in accordance with the specified language.  
Strings that will be output vary depending on whether machine-translation is performed, as shown below.
    - When machine-translation is performed  
The result of machine-translation is output.
    - When machine-translation is not performed  
The output will be empty (not defined).

For the replaced information that will be output for each language, refer to the following:

=> ["• Information to be replaced for each language when outputting multilingual information"](#)

[Output Example of Format File]

The following is an output example for when the font for the field defined in the form descriptor information is MS Mincho and multilingual replace information is in English.

```

,,FONT_JAPANESE,"MS Mincho","Times New Roman"
,,FONT_ENGLISH,"MS Mincho","Times New Roman"
    
```

- When targeting specific form(s) only



**Note**

- Each field of the form defined in the form descriptor information will be output. Even if property information overlaps, it is output as long as the fields are not the same.
- To output by specifying a property type, specify the property type to be output in the [Format File Settings] window and create a format file.  
For details of the [Format File Settings] window, refer to the Designer Help guide.
- The property identifier "NUM\_DECIMAL\_FORMAT" for the decimal format for numerical data will be output only once, regardless of whether property information is overlapping. Also, [Field\_name] is not output.

[Description Format of Format File]

```
Form_name,Field_name,FONT_JAPANESE,"Specified_Japanese_font_name",
Form_name,Field_name,FONT_ENGLISH," Specified_English_font_name",
Form_name,Field_name,LITERAL_STRING,"Specified_character_string",
Form_name,Field_name,LITERAL_RECTSTRING,"Specified_character_string",
Form_name,Field_name,LITERAL_LABEL,"Specified_label",
Form_name,Field_name,CURRENCY_SYMBOL,"Specified_currency_symbol",
Form_name,Field_name,DATE_FORMAT_STRING,< Specified_date_edit_format>,
Form_name,NUM_DECIMAL_FORMAT,0|1(Specified_decimal_format_style),
Form_name,Field_name,NUM_DECIMAL_PLACES,<Specified_number_of_decimal_places>,
Form_name,Field_name,NUM_DECIMAL_PLACES_SUPPRESS,0|1(Specified_setting_of_whether_to_output_the_decimal_point),
Form_name,Field_name,NUM_DECIMAL_PLACES_FORMAT,0|1|2(Specified_decimal_places_format),
```

(\*1)                      (\*2)                      (\*3)                      (\*4)                      (\*5)

- \*1: The form names are output.
- \*2: The field names are output.
- \*3: Each property identifier is output.
- \*4: Information before replacement (information defined in the form descriptor information) is output.
- \*5: Information after replacement is output. What will be output varies depending on whether to output language information, as shown below.
  - When language information is not output  
The output will be empty (not defined).
  - When language information is output  
Information after replacement is output in accordance with the specified language.  
Strings that will be output vary depending on whether machine-translation is performed, as shown below.
    - When machine-translation is performed  
The result of machine-translation is output.
    - When machine-translation is not performed  
The output will be empty (not defined).

For the replaced information that will be output for each language, refer to the following:

=> "[• Information to be replaced for each language when outputting multilingual information](#)"

[Output Example of Format File]

The following is an output example for when all the conditions shown below are met and multilingual replace information is in English.

- Form descriptor information (Form1)
  - The separator character ", (comma)" and the decimal point ". (period)" are specified for the decimal format style in [Form Properties].
  - The Japanese font MS Mincho is defined for the static field (Static0001).
  - The English font MS Mincho is defined for the static field (Static0001).
  - The string " 会社名 " is defined for the static field (Static0001).
- Form descriptor information (Form2)
  - The separator character ". (period)" and the decimal point ", (comma)" are specified for the decimal format style.
  - The English font MS Mincho is defined for the numeric field (Numeric0001).
  - The number of decimal places "0" is defined for the numeric field (Numeric0001).
  - The currency symbol "¥" is defined for the numeric field (Numeric0001).



```
Form1,,NUM_DECIMAL_FORMAT,0,0
Form1,Static0001,FONT_JAPANESE,"MS Mincho","Times New Roman"
Form1,Static0001,FONT_ENGLISH,"MS Mincho","Times New Roman"
Form1,Static0001,LITERAL_STRING, " 会社名 ", "Company name"
Form2,,NUM_DECIMAL_FORMAT,1,0
Form2,Numeric0001,FONT_ENGLISH, "MS Mincho","Times New Roman"
Form2,Numeric0001,NUM_DECIMAL_PLACES,0,0
Form2,Numeric0001,CURRENCY_SYMBOL,"¥","$"
```

- Information to be replaced for each language when outputting multilingual information

When [Output Replace Information about Multilingual] is specified in the [Format File Settings] window, the replace information for each language that will be output to a format file is as follows:

| Property Type | Property Identifier               | Replacing Language      | Replace Information Output to Format File   |
|---------------|-----------------------------------|-------------------------|---|
| Font          | - FONT_JAPANESE<br>- FONT_ENGLISH | Japanese                | M S 明朝  |
|               |                                   | English                 | Times New Roman   |
|               |                                   | Chinese(Simplified)     | SimSun  |
|               |                                   | Chinese(Traditional)    | MingLiU   |
|               |                                   | Korean                  | BatangChe   |
|               |                                   | French                  | Times New Roman   |
|               |                                   | Spanish                 |   |
|               |                                   | German                  |   |
|               |                                   | Russian                 |   |
|               |                                   | Portuguese              |   |
|               |                                   | Czech                   |   |
|               |                                   | Thai                    | Tahoma  |
| Vietnamese    |                                   |                         |   |
| Literal       | LITERAL_STRING                    | Common to all languages | - When machine-translation is performed<br>The result of machine-translation (*1)<br>- When machine-translation is not performed<br>Empty (not defined) |
|               | LITERAL_RECTSTRING                |                         |   |
|               | LITERAL_LABEL                     |                         |   |

| Property Type  | Property Identifier     | Replacing Language  | Replace Information Output to Format File  |
|--|-------------------------|---|--|
| Value  | NUM_DECIMAL_FORMAT      | Japanese  | 0<br>* The separator character for every three digits is a ", (comma)" and the decimal point is a ". (period)" |
|  |                         | English   |  |
|  |                         | Chinese(Simplified)   |  |
|  |                         | Chinese(Traditional)  |  |
|  |                         | Korean  |  |
|  |                         | Thai  |  |
|  |                         | French  | 1<br>* The separator character for every three digits is a ". (period)" and the decimal point is a ", (comma)" |
|  |                         | Spanish   |  |
|  |                         | German  |  |
|  |                         | Russian   |  |
|  |                         | Portuguese  |  |
|  |                         | Czech   |  |
|  | Vietnamese              |   |  |
| - NUM_DECIMAL_PLACES<br>- NUM_DECIMAL_PLACES_SUPPRESS<br>- NUM_DECIMAL_PLACES_FORMAT | Common to all languages | Same property as specified in the form descriptor information |  |
| Currency   | CURRENCY_SYMBOL         | Japanese  | ¥  |
|  |                         | English   | \$   |
|  |                         | Chinese(Simplified)   | ¥  |
|  |                         | Chinese(Traditional)  | NT\$   |
|  |                         | Korean  | ₩  |
|  |                         | French  | €  |
|  |                         | Spanish   |  |
|  |                         | German  |  |
|  |                         | Russian   | ₽  |
|  |                         | Portuguese  | €  |
|  |                         | Czech   | Kč   |
|  |                         | Thai  | ฿  |
|  |                         | Vietnamese  | ₫  |
| Date   | DATE_FORMAT_STRING      | Japanese  | YYYY/MM/DD   |
|  |                         | English   | MM/DD/YYYY   |
|  |                         | Chinese(Simplified)   | YYYY/MM/DD   |
|  |                         | Chinese(Traditional)  | YYYY/M/D   |
|  |                         | Korean  | YYYY-MM-DD   |
|  |                         | French  | DD/MM/YYYY   |
|  |                         | Spanish   |  |
|  |                         | German  | DD.MM.YYYY   |
|  |                         | Russian   |  |
|  |                         | Portuguese  | DD-MM-YYYY   |
|  |                         | Czech   | DD.MM.YYYY   |
|  |                         | Thai  | DD/MM/YYYY   |
|  |                         | Vietnamese  |  |

\*1: Special strings in date fields and time fields are also machine-translated. As a result, such machine-translated strings may not be supported by List Creator. Be sure to preview and check the characters.

# ***Part 3 Management and Maintenance***

---

This part explains how to centrally manage and maintain the form descriptor information (maintaining forms).

# **Chapter 12**

## *Managing Form Descriptor Information*

---

This chapter explains the method for managing created form descriptor information.

Form descriptor information is managed in the [Form Descriptor List] window.

The method for displaying the [Form Descriptor List] window, as well as its structure and operating instructions are explained below.

## 12.1 About Form Descriptor Information Management

The form descriptor information is saved in the form storage directory (the folder for storing form descriptor information).

Create form storage directories for each type of form descriptor information in order to save/manage and categorize form descriptor information according to its type.

Management of form descriptor information is centralized from the [Form Descriptor List] window.

The following operations can be performed from the [Form Descriptor List] window:

- Listing Form Descriptor Information
- Specifying the Form Storage Directory
- Creating, opening, copying, deleting, and renaming form style information
- Setting and Viewing Form Output Methods
- Checking the output image of the form style information
- Outputting the List of Form Descriptor Information
- Printing Reports of Form Style Information
- Creating a Format Property Replace File

The following sections discuss each operation.

For instructions on displaying the [Form Descriptor List] window, refer to the following:

=> ["3.1.1 \[Form Descriptor List\] Window"](#)

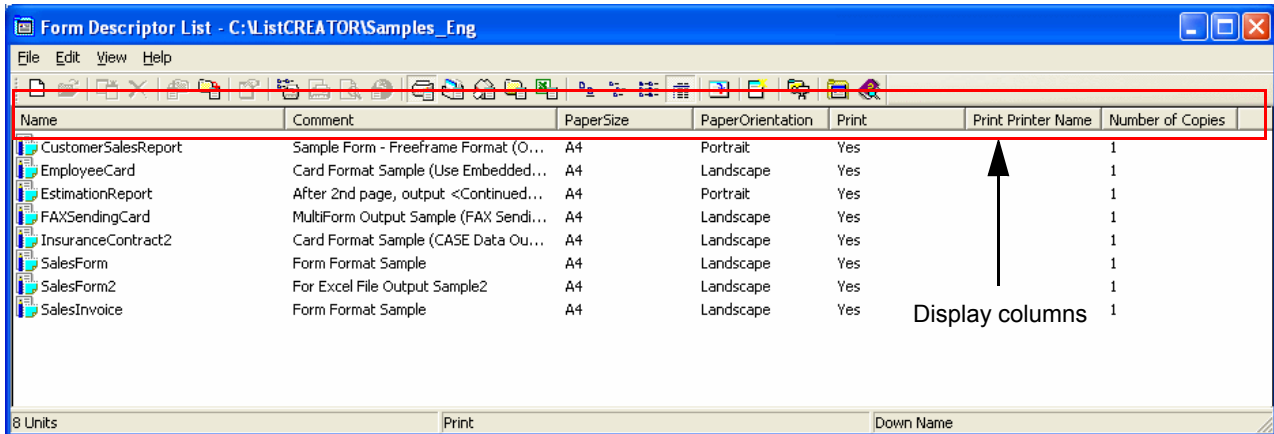


**Caution**

- If the form storage directory does not exist when opening the [Form Descriptor List] (e.g. to delete a form storage directory), the [Forms Directory Specification] window is displayed. When this is displayed, specify the form storage directory.

## 12.2 Listing Form Descriptor Information

Form descriptor information that is saved to the form storage directory is listed to the [Form Descriptor List] window. Display columns can be changed according to the output method, or form descriptor information can be resorted by display columns. For operation details, refer to the Designer Help guide.

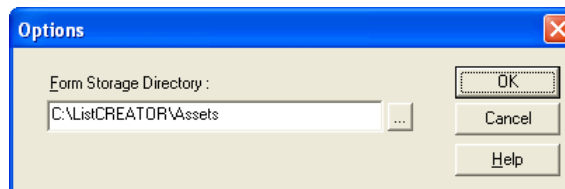


## 12.3 Specifying the Form Storage Directory

Forms in the specified form storage directory are listed on the [Form Descriptor List] window. This section explains how to specify the form storage directory.

### = Procedure =

- 1) Select [View] - [Options] in the [Form Descriptor List] window.  
=>The [Options] window is displayed.
- 2) Specify the form storage directory.  
This can also be specified from the window displayed by clicking the [...] button to the right of the input area.  
When browsing a directory, specify it within 259 bytes.



- 3) Click the [OK] button.  
=>The form in the specified form storage directory is displayed on the [Form Descriptor List] window.



### Note

- For Windows, the form storage directory can also be switched from the [General] tab in the [List Creator Settings] window. If the form storage directory is changed on one, the change is reflected to the other.
- To change the form storage directory in the [Options] window in the [Form Descriptor List] window, first close the [List Creator Settings] window.

## 12.4 Creating, Opening, Copying, Deleting, and Renaming Form Style Information

The following operations can be performed to the form style information from the [Form Descriptor List] window:

- New  
This creates new form style information. The form style information is saved to the displayed form storage directory.
- Open  
This opens form style information. The definitions can be viewed and changed.
- Copy  
A copy of the form style information can be created. The copied form style information is automatically assigned a name.
- Copying from other directory  
Form descriptor information that is saved to other form storage directory can be copied to the form storage directory displayed in the [Form Descriptor List] window.
- Copying to other directory  
Form descriptor information that is saved in the form storage directory displayed in the [Form Descriptor List] window can be copied to other form storage directory.
- Rename  
Form descriptor information can be renamed.

For the New operation, refer to the following:

=> ["Chapter 5 Creating Form Layout Data"](#)

For the Open operation, refer to the following:

=> ["13.1 Changing the Form Style Information"](#)

For the Copy operation, Copying to and from another form storage directory, and the Rename operation, refer to the Designer Help guide.

## 12.5 Setting and Viewing Form Output Methods

Form output methods can be defined to form style information.

For details on defining form output methods, refer to the following:

=> ["Chapter 9 Defining the Form Output Methods"](#)

## 12.6 Checking the Output Image of the Form Style Information

You can check the output image of the form style information from the [Form Descriptor List] window.

For checking the output image, refer to the following:

=> ["7.2 Preview"](#)

## 12.7 Outputting the List of Form Descriptor Information

Output the List of Forms is a function that outputs the list of form descriptor information shown in the [Form Descriptor List] window.

There are two modes for outputting form lists:

- Output the List of Forms

This outputs information such as form storage directories, number of forms, form names, and comments. Lists can be printed, or output as PDF files, Excel files, or text files (in CSV format).

- Output the List of Forms and Existence of change

In addition to the list of the form descriptor information, existence of any changes is output by comparing the current data with the directory storing the form descriptor information before the changes are made. Lists can be printed, or output as PDF files or Excel files.

Three types of lists are output:

- Form List

The existence of change in each form is output.

- Form List (Change Information: Show Type)

Existence of change in form output methods and existence of change in form descriptor information (form layout data, data definitions) are output.

- Form List (Change Information: Field and Object)

Change/addition/deletion information for form layout data fields and objects is output.



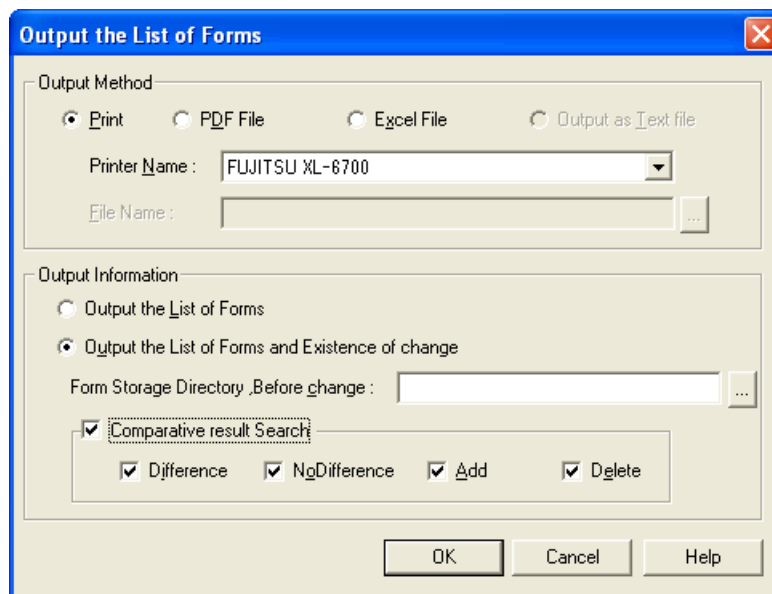
**Note**

- When printing, the list is printed on a cut-sheet of A4 paper according to the properties of the specified printer driver. Place A4 paper in the printer, then set the properties of the printer driver to print with A4 paper before printing.
- When outputting as a PDF file, the paper size is A4.
- The list of form descriptor information is output in the order displayed in the [Form Descriptor List] window.

The following shows an example for outputting the list of forms and existence of change.

**= Procedure =**

- 1) In the [Form Descriptor List] window, select [File] - [Output the List of Forms].  
=>The [Output the List of Forms] window is displayed.



- 2) Specify the output method. Here, select "Print".



- 3) Specify the name of the printer to use for output. The printer name can be selected from printers defined in the computer.



**Note**

- When you select "PDF File" or "Excel File", specify "File Name".

- 4) In the [Output Information], select "Output the List of Forms and Existence of change".
- 5) Specify a directory in the "Form Storage Directory Before Change". The form descriptor information stored in this directory is compared with the form descriptor information stored in the form storage directory. This directory can also be specified from the window displayed by clicking the [...] button to the right of the input area.



**Caution**

- The following characters cannot be used in a directory name:  
"/", "\", ":", ":", "\*", "?", " ", "<", ">", "|"

- 6) Check the "Comparative result Search" checkbox to filter the output comparison results. Multiple comparative result search conditions can be specified.

-Difference

If the form exists in both the form storage directory and the directory specified in "Form Storage Directory, Before Change", and its form descriptor information has been changed, the form is output.

- NoDifference

If the form exists in both the form storage directory and the directory specified in "Form Storage Directory, Before Change", and its form descriptor information has not been changed, the form is output.

- Add

The form descriptor information that exists only in the form storage directory is output.

- Delete

The form descriptor information that exists only in the directory specified in "Form Storage Directory Before change" is output.



**Note**

- When the "Comparative result Search" checkbox is checked, and "Difference", "NoDifference", "Add", or "Delete" is not checked, comparison is not performed.

- 7) Click the [OK] button.  
=>A result corresponding to the setting is output.

## 12.7.1 Identifying Changes from Output Results

When you output the list of forms and existence of change, the following information is output in the "Form List (Change Information: Field and Object)":

**Category** : Category of the change to a field/object is shown.  
"Add", "Remove" or "Change" is output. (\*1)

**Type** : Type of the field/object is shown.

**Information** : Information for identifying the field/object output in the "Type" is shown.

\*1: If a partition graphic, overlay text or overlay graphic has been moved, it is considered that the original object has been deleted and new object has been added. and both "Delete" and "Add" are shown.

The details of the information shown in the "Type" and "Information" are as follows:

| Type              | Information  |
|-------------------|--|
| Frame             | The frame name is output.  |
| Partition         | The partition name is output.  |
| Field             | The change information is output in the following format:<br>Field name (Partition name in which the field is placed)<br><br>Information of the following fields is output:<br><ul style="list-style-type: none"> <li>- Numeric field</li> <li>- Text field</li> <li>- Block text field</li> <li>- OCR-B field</li> <li>- Radio button</li> <li>- Check box</li> <li>- Barcode field (including U.S. Postal FIM field)</li> <li>- Image field</li> <li>- Post code field</li> <li>- Hidden field</li> <li>- Group item (including body area group item in simple table, sticky label, or block repeat format)</li> </ul> |
| Literal           | The change information is output in the following format:<br>Field name (Partition name in which the field is placed)<br><br>Information of the following fields is output:<br><ul style="list-style-type: none"> <li>- Static field</li> <li>- Block static field</li> <li>- Date field</li> <li>- Time field</li> </ul>  |
| Partition graphic | The change information is output in the following format:<br>Types of partition graphic (Partition name in which the partition graphic is placed, (Start coordinates)) (*1)  |
| Overlay text      | The start coordinates are output. (*2)   |
| Overlay graphic   | The change information is output in the following format:<br>Type of overlay graphic (Start coordinates) (*2)  |

\*1: In the start coordinates, the relative coordinates from the top left of the partition in which the partition graphic is placed are output in dots.

\*2: In the start coordinates, the absolute coordinates from the top left of the edit window are output in dots.

You can use following methods to identify changed/added/deleted fields and objects from the information output in "Type" and "Information".

| Type of the field/object | Identification method (Procedure)  |
|--------------------------|--|
| Frame                    | <ol style="list-style-type: none"> <li>1) Display the target form descriptor information in [Form Style Descriptor] window.</li> <li>2) Search for the "Frame name" in the partition list.</li> </ol>  |
| Partition                | <ol style="list-style-type: none"> <li>1) Display the target form descriptor information in [Form Style Descriptor] window.</li> <li>2) Search for the "Partition name" in the partition list.</li> </ol>  |
| Field                    | <ol style="list-style-type: none"> <li>1) Display the target form descriptor information in [Form Style Descriptor] window.</li> <li>2) Search for the "Field name" in the partition or record list.</li> </ol>  |
| Literal                  | <ol style="list-style-type: none"> <li>1) Display the target form descriptor information in [Form Style Descriptor] window.</li> <li>2) Search for the "Field name" in the partition or record list.</li> </ol>  |
| Partition graphic        | <ul style="list-style-type: none"> <li>●For multipart table format                             <ol style="list-style-type: none"> <li>1) Display the target form descriptor information in [Form Style Descriptor] window.</li> <li>2) Select "Unit setting" from the "View" menu, and set the display unit to "Dot".</li> <li>3) Display the property list.</li> <li>4) Select "Select Editing Objects" from the "Edit mode" menu, and specify only the "Partition graphic" as the edit target.</li> <li>5) Select "Layout creation" from the "Edit mode" menu, and select "Layout creation (Standard)".</li> <li>6) Select "Partition" from the "View" menu, and select "Hide All Partition".</li> <li>7) Select the output partition from the partition list, and select "View" menu - "Partition" - "Show Partition".</li> <li>8) Compare the output start coordinates and the position displayed in the status bar (mouse coordinates), and select the nearest partition graphic.</li> <li>9) Check the output start coordinates and the location displayed in the property list.</li> </ol> </li> <li>●For free frame format                             <ol style="list-style-type: none"> <li>1) Display the target form descriptor information in [Form Style Descriptor] window.</li> <li>2) Select "Unit setting" from the "View" menu, and set the display unit to "Dot".</li> <li>3) Display the property list.</li> <li>4) Select "Select Editing Objects" from the "Edit mode" menu, and specify only the "Partition graphic" as the edit target.</li> <li>5) Select "Layout creation" from the "Edit mode" menu, and select "Layout creation (Standard)".</li> <li>6) Select "Partition" from the "View" menu, and select "Hide All Partition".</li> <li>7) Select the output partition from the partition list, and select "View" menu - "Partition" - "Show Partition".</li> <li>8) Add the location of the frame in which the partition is placed to the displayed start coordinates, and calculate the absolute coordinates.</li> <li>9) Compare the calculated absolute coordinates and the position displayed in the status bar (mouse coordinates).</li> <li>10) Select the nearest partition graphic, and check the output start coordinates and the location displayed in the property list.</li> </ol> </li> </ul> |

| Type of the field/object | Identification method (Procedure)   |
|--------------------------|---|
| Overlay text             | <ol style="list-style-type: none"> <li>1) Display the target form descriptor information in [Form Style Descriptor] window.</li> <li>2) Select "Unit setting" from the "View" menu, and set the display unit to "Dot".</li> <li>3) Display the property list.</li> <li>4) Select "Select Editing Objects" from the "Edit mode" menu, and specify only the "Overlay Text" as the edit target.</li> <li>5) Compare the output start coordinates and the position displayed in the status bar (mouse coordinates), and select the nearest overlay text.</li> <li>6) Check the output start coordinates and the location displayed in the property list.</li> </ol>       |
| Overlay graphic          | <ol style="list-style-type: none"> <li>1) Display the target form descriptor information in [Form Style Descriptor] window.</li> <li>2) Select "Unit setting" from the "View" menu, and set the display unit to "Dot".</li> <li>3) Display the property list.</li> <li>4) Select "Select Editing Objects" from the "Edit mode" menu, and specify only the "Overlay Graphic" as the edit target.</li> <li>5) Compare the output start coordinates and the position displayed in the status bar (mouse coordinates), and select the nearest overlay graphic.</li> <li>6) Check the output start coordinates and the location displayed in the property list.</li> </ol> |

## 12.8 Printing Reports of Form Style Information

Reports (descriptor data) of form style information can be printed in a list format, or output to a PDF file or an Excel file.

When printing reports of form style information, the following fields can be set:

### Form image

Output image of the form is output.

### Form summary report

Information regarding the entire form layout is output.

### Field Details Report

Information regarding the fields defined to the form layout is output. There are two output formats: standard format and list format.

For the output fields, select to output all fields including static fields or to output fields with the exception of static fields (including date fields and time fields).

### Partition details report

Information regarding partitions is output. Partition details reports can only be output for free frame format or multipart table format forms.

For free frame format forms, partition information is output for each frame.

### Input data report

When the input data format is "Variable Length", "Fixed Length", or "CSV Format", information required for designing the input data is output.

### = Procedure =

- 1) Select the form style information whose report you want to output on the [Form Descriptor List] window, and select [File] - [Form Contents Report].  
=>The [Form Contents Report] window is displayed.

2) Select an output method and report settings.



**Note**

- If "Print" is selected for the "Output Method", specify the printer name to use for output in "Printer Name" ("Printer Name" can be specified only when "Print" is selected). If "PDF File" or "Excel File" is selected, specify the "Output Destination Directory" ("Output Destination Directory" can be specified only when "PDF File" or "Excel File" is selected).
- Partition details reports can only be output for free frame format or multipart table format forms.

3) Click the [OK] button.

=>The report of the form style information is output in the specified output method.



**Note**

- If "Print" is selected for the "Output Method", one copy of the form style information report is printed on a single side of A4 paper according to the properties of the specified printer driver. Place A4 paper in the printer, then set the properties of the printer driver to print with A4 paper before printing.



**Caution**

- When outputting a report of the form style information that uses Unicode characters, the Unicode strings in the following fields may not be output if "PDF File" is selected for the "Output Method":
  - "ForbiddenAtStart-of-line-char" or "ForbiddenAtEnd-of-line-char" of the "Form Specifications"
  - "DittoString" in category "Field Information" of the "Field Specifications"
  - "EditFormatString" in category "Formatting" of the "Field Specifications"
  - "MinusSign" in category "Formatting" of the "Field Specifications"
  - "CurrencySymbol" in category "Formatting" of the "Field Specifications"
  - "String" in category "Static Field" of the "Field Specifications"
  - "RadioButtonLabel" in category "Radio Button" of the "Field Specifications"
  - "CheckBoxLabel" in category "Check Box" of the "Field Specifications"In this case, select "Print" or "Excel File" for the "Output Method".

## 12.9 Creating a Format Property Replace File

A format property replace file can be created in the [Form Descriptor List] window.

For details, refer to the following:

=> ["11.2.1 Creating Format Property Replace File"](#)

# Chapter 13

## Maintaining Form Descriptor Information

---

During form output, the layout or output method of the form may need to be modified according to the changes of the way the form is used. In this case, modify (update) the form descriptor information. Modifying the form descriptor information is referred to as "maintaining forms" in List Creator.



- Do not modify (update) form resources while a form is being output. Modifying (updating) them during the output operation may cause an unexpected output result or an error and there is no guarantee of operation.


## 13.1 Changing the Form Style Information

When the form layout data needs to be changed, open the form descriptor information from the [Form Descriptor List] window to change it. Information for the selected several fields can be collectively changed.

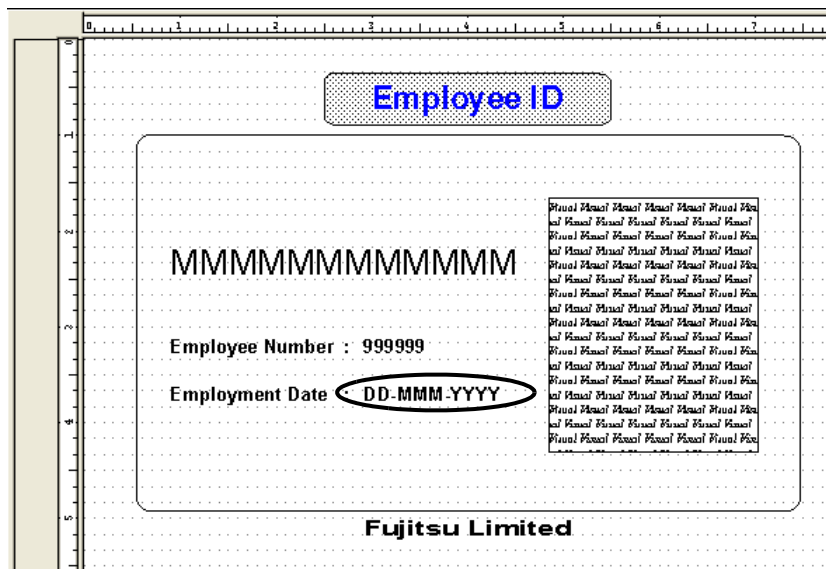
Below is an explanation of the method for changing form style information. In the following example, the formatting for "EmploymentStartDate" in the sample form "EmployeeID" is converted from the British style DD-MMM-YYYY (e.g. 18-Dec-1994) date format to the American style MM-DD-YYYY (i.e. 12-18-1994) date format.

**= Procedure =**

- 1) Select the "EmployeeID" sample form in the [Form Descriptor List] window, and select [File] - [Open].

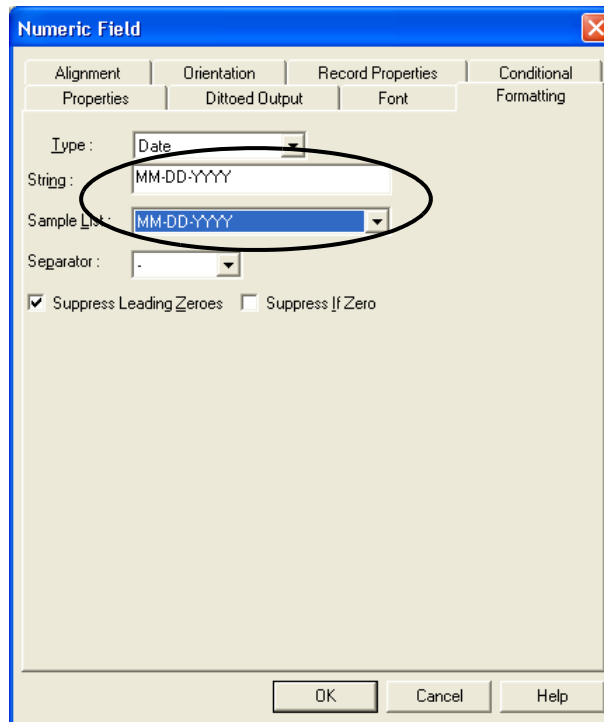
 **Note** - The [Form Style Descriptor] window can also be displayed by double clicking a form name.

=>The [Form Style Descriptor] window is displayed.

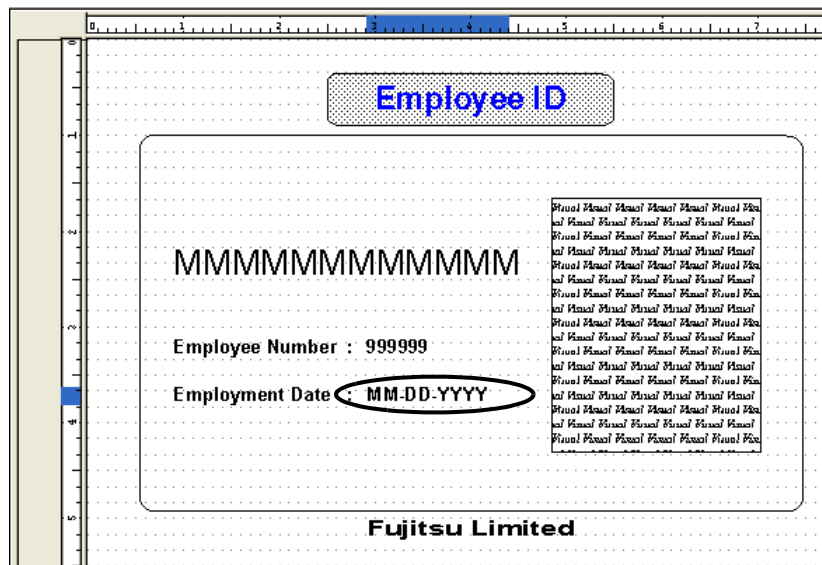


- 2) Select the "EmploymentStartDate" numeric field, and display the properties window of the numeric field. For details on how to display the numeric field properties window, refer to the Designer Help guide.
- 3) Select "MM-DD-YYYY" (Month-Day-Year) from the "Sample List" pull-down on the [Formatting] tab. =>The "MM-DD-YYYY" format appears in the [String] area.





- 4) Click the [OK] button.  
=>The display is as shown below.



**Note** - You can also change several form style information items simultaneously by selecting multiple fields.

- 5) Select [File] - [Close].  
=>The changes are saved, and the display returns to the [Form Descriptor List] window.

## 13.2 Changing the Form Output Methods

When the form output method or operation during form output needs to be changed, you can change the data easily by displaying the [Form Properties] window from the [Form Descriptor List] window.

To display the [Properties] window, select the form descriptor information to be changed in the [Form Descriptor List] window and click the [Form Properties] button on the toolbar. You can also change several form output methods simultaneously by selecting multiple form descriptor information items and displaying the [Form Properties] window.

For details on changing form output methods, refer to the following:

=> ["Chapter 9 Defining the Form Output Methods"](#)

---

## ***Part 4 Overlay Utility***

---

This part explains the procedures for performing the environment setting of the Overlay Utility and for creating and adding overlays and forms.

# **Chapter 14**

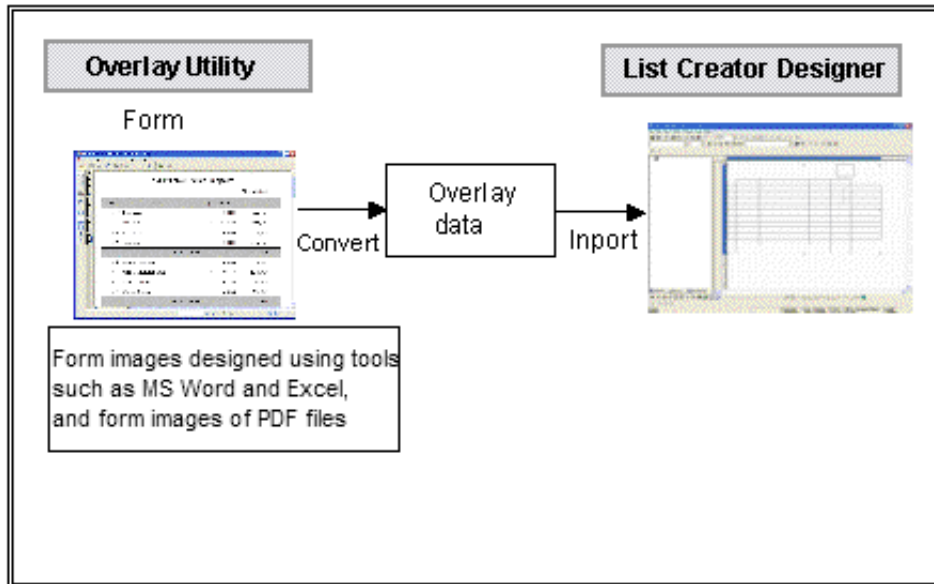
## *The Overlay Utility*

---

This chapter provides an overview of the Overlay Utility.

## 14.1 The Overlay Utility

When designing forms in List Creator, you need to define the data regarding the form layout, such as fields, positions, and formats. You can also define graphic (such as lines, rectangles, and ellipses), and image output in fixed locations. Graphics (such as lines) that are overlapped over the fields are referred to as overlay. The Overlay Utility allows the creation of overlay and form quickly and easily from electric forms such as Microsoft Word, PDF files, etc.



## 14.2 Features of the Overlay Utility

The features of the Overlay Utility are shown below.

### 14.2.1 Overlay Creation Efficiency

Previously, a form preserved as electronic data such as a word processor text file had to be printed on a sheet, then scanned. Now, using the Overlay Utility, an overlay and a form can be created simply by printing from your word processing software or spreadsheet software.

### 14.2.2 Form Design using Your Favorite Tools

Overlay and form can be created simply by printing. Therefore, you can continue to use your favorite tools, such as word processing software and spreadsheet software, for form designing.

### 14.2.3 Forms Published on the Internet

PDF file form published on the Internet can easily be converted into overlay and form.

# **Chapter 15**

## *Installation*

---

This chapter explains installation and setup for the Overlay Utility.

## 15.1 Installation

The installation procedures for the Overlay Utility are as follows.



### Caution

- Administrators privileges are required for installation. Logon as a user with administrators privileges.
- Close all open Windows applications before installation.
- List Creator Designer is required for using the Overlay Utility. Check that it has already been installed.
- The version of the Overlay Utility and List Creator Designer must be same.
- If a previous version of Overlay Utility is installed, uninstall it before installation procedure. Refer to the following for procedures and points to note of uninstall.  
=> "[15.2 Uninstall](#)"

### 15.1.1 Installing from Product Media

Display the List Creator Designer product media in Explorer, and execute the following install command:

```
List_Creator_Product_Media¥japanese¥Ovdut¥ISsetup.exe
```

When installing in an English environment, execute the following install command:

```
List_Creator_Product_Media¥english¥Ovdut¥ISsetup.exe
```

The windows which appear during installation are shown below.

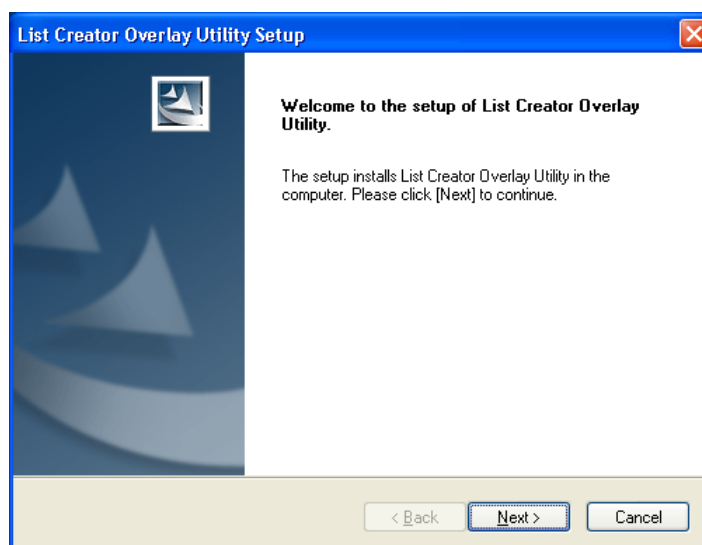


### Note

- For Windows Vista, Windows 7, and Windows Server 2008, when ISsetup.exe is run, the User Account Control screen will appear. Click [Continue].

#### (1) Installing Shield Wizard Startup

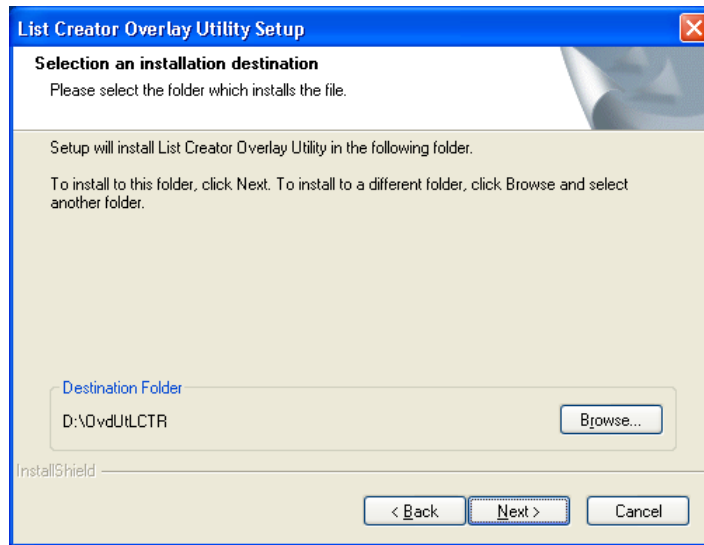
The window below is shown when installation begins. Click [Next].





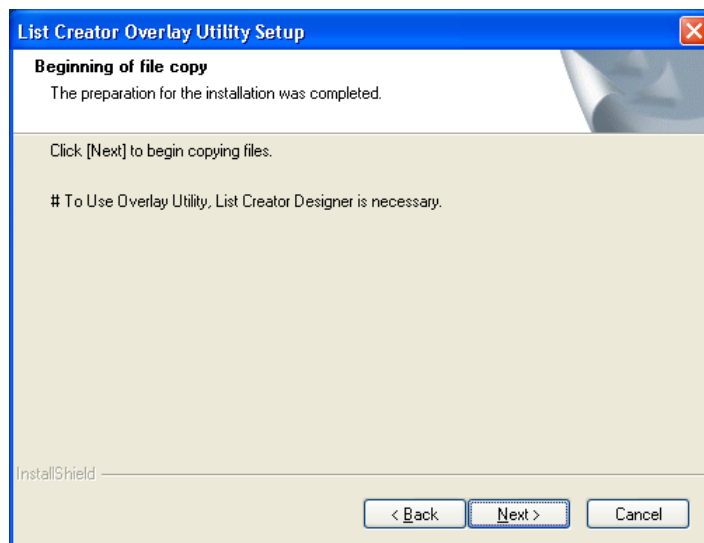
(2) Selection an installation destination

Confirm the installation folder. If there is no problem with the destination, click [Next]. To change the destination, click [Browse] and select an installation folder.



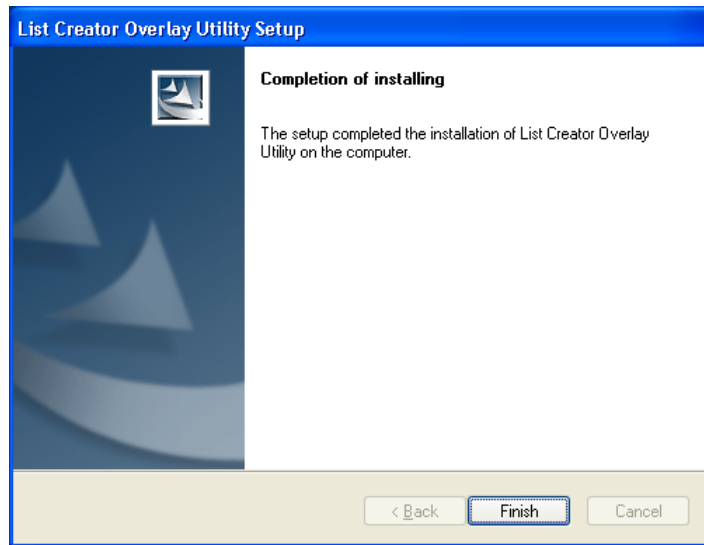
(3) Beginning of file copy

Click [Next] to begin of file copy.



#### (4) Completion of installing

Click [Finish]. Installation is complete.



## 15.2 Uninstall

---



- During uninstallation, the installation directory is deleted. Therefore, any necessary resources (completed overlays, etc.) located in the installation directory must be backed up before uninstallation.

Use the following procedure to uninstall the Overlay Utility.

- 1 Logon as a user with administrators privileges.
- 2 Close all open Windows applications.
- 3 Delete all Overlay Utility-related printer definition information from the Printer and Faxes folder window.
- 4 Display the [Add/Remove Programs] window from [Start] menu, by selecting [Settings] - [Control Panel]. For Windows XP and Windows Server 2003, display the [Add or Remove Programs] window from [Start] menu, by selecting [Control Panel].  
  
For Windows Vista, Windows 7, and Windows Server 2008, select [Start] - [Control Panel] to display the Control Panel. In the Control Panel, select [Uninstall a Program] to display the "Uninstall or Change a Program" screen.
- 5 Select the "List Creator Overlay Utility", and run uninstall.
- 6 After uninstall, from the [Printer] window, select [File] - [Server Properties] to open the [Print Server Properties] window. Select the [Ports] tab, and check whether "Fujitsu ListCREATOR OVD Port" exists or not. If it exists, delete the port that is using "FUJITSU ListCREATOR OVD Port". Select the [Drivers] tab, and delete the "FUJITSU Overlay Create Driver".

## 15.3 Setup

To use the Overlay Utility, a printer must be defined in a configuration with a printer driver and a print monitor. The example below shows the procedures for defining the Overlay Utility printer.

### 15.3.1 Printer Definition in Windows

Use the Add Printer Wizard, and define the printer.

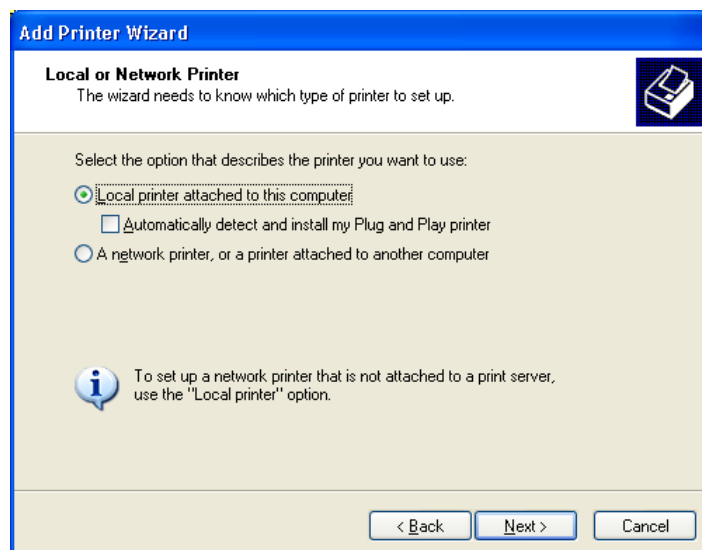


**Note**

- For Windows 7, use the following procedure to add a port before defining a printer.
  - 1) Click [Start] - [Control Panel] - [System and Security] - [Administrative Tools] to open "Print Management".
  - 2) In the console tree, select [Print Management] - [Print Servers] - [Server Name (local)] - [Ports].
  - 3) Select [Operations] - [Add Port].  
=> The [Printer Ports] screen appears.
  - 4) For "Available Port Types" select "FUJITSU ListCREATOR OVD Port", and then click the [New Port] button.
  - 5) Set the required information in the ListCREATOR Overlay Utility Environment Setting window, and then click the [OK] button.
- For Windows Vista and Windows Server 2008, the "Add Printer" process must be performed by an administrator. The following example for this procedure is based on Windows Vista.
  - 1) Click [Start] - [Control Panel] - [Hardware and Sound] - [Printer] to open the printer folder.
  - 2) Right-click the printer you want to manage, and then select [Run as administrator] - [Add Printer].
  - 3) The User Account Control screen will appear. Select [Continue].
  - 4) Click the [Next] button and the Windows Security screen will appear. Select "Install this driver software anyway".

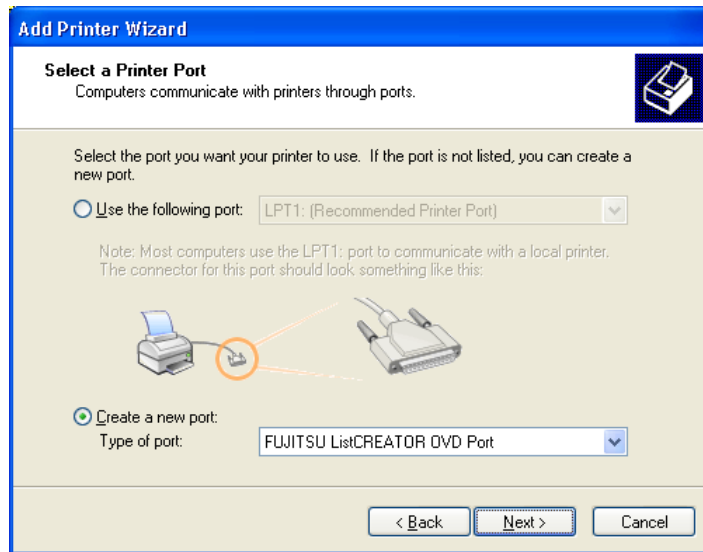
#### (1) Adding a printer

Select [Add Printer] from the [Printer] folder, and the "Add Printer Wizard" will start. Select "Local printer attached to this computer", then click the [Next] button. At this point, uncheck "Automatically detect and install my Plug and Play printer".

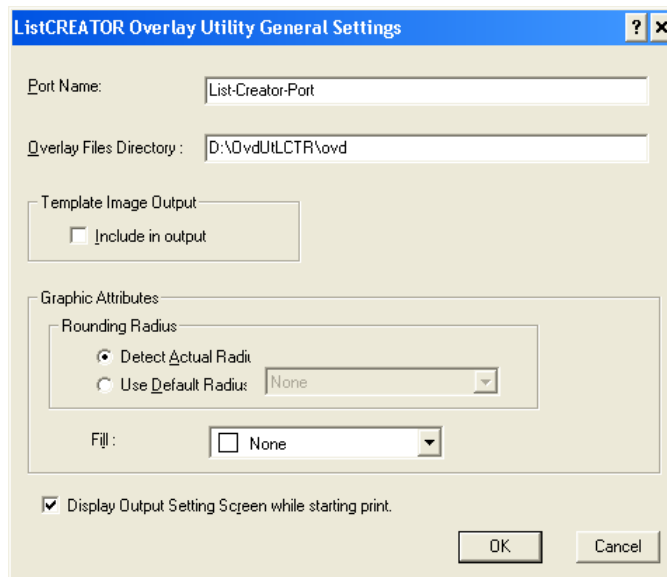


(2) Defining printer port

Select "Create a new port", then specify "FUJITSU ListCREATOR OVD Port", and click [Next].



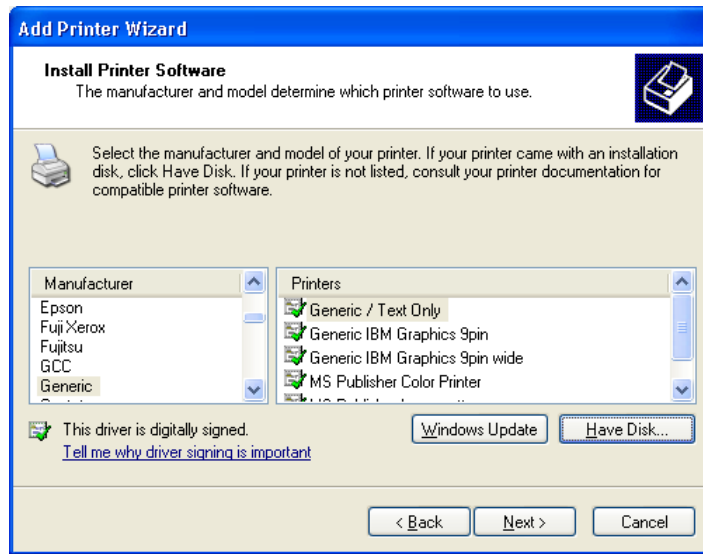
The [ListCREATOR Overlay Utility General Settings] window will be displayed. Input the port name, click [OK], and close the [ListCREATOR Overlay Utility General Settings] window. The port name "List-Creator-Port" is input in the following example.



- The content set in the [ListCREATOR Overlay Utility General Settings] window will be shown as the default value of the Overlay Utility. Field to be set can be also changed after printer definition. For details on the procedure for changing and field to be set, refer to following. => "15.3.2 Setup (Port Definition Window)"

(3) Adding a printer driver

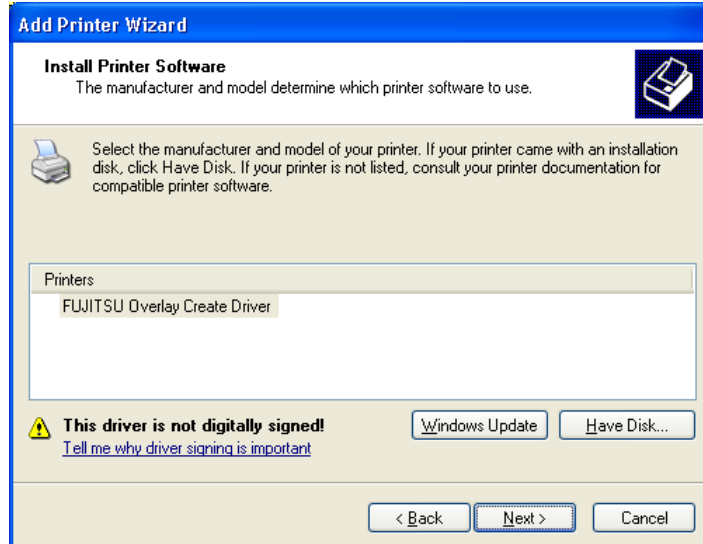
The printer driver setting window will be shown, so the Overlay Utility printer driver can be added. Click the [Have Disk] button.



Specify the driver directory under the Overlay Utility installation directory. Then, on the window shown, select "FUJITSU Overlay Create Driver", and click [Next].

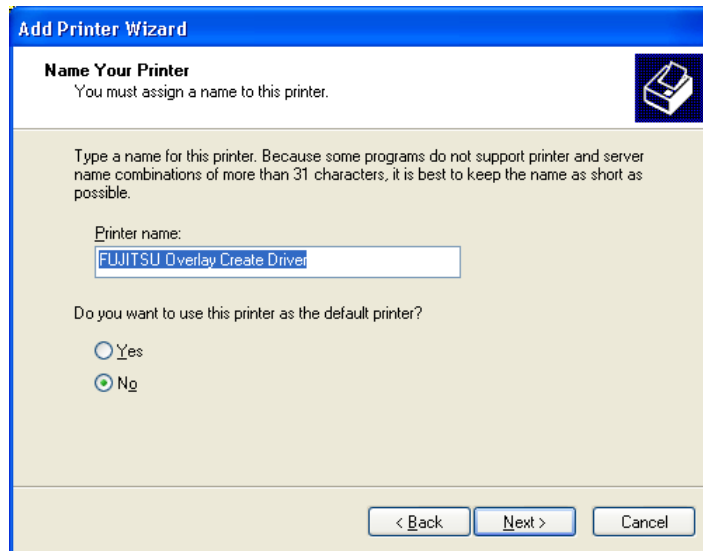


- Be sure to click the [Have Disk] button to install a new driver regardless whether "FUJITSU Overlay Create Driver" is shown or not for a printer.



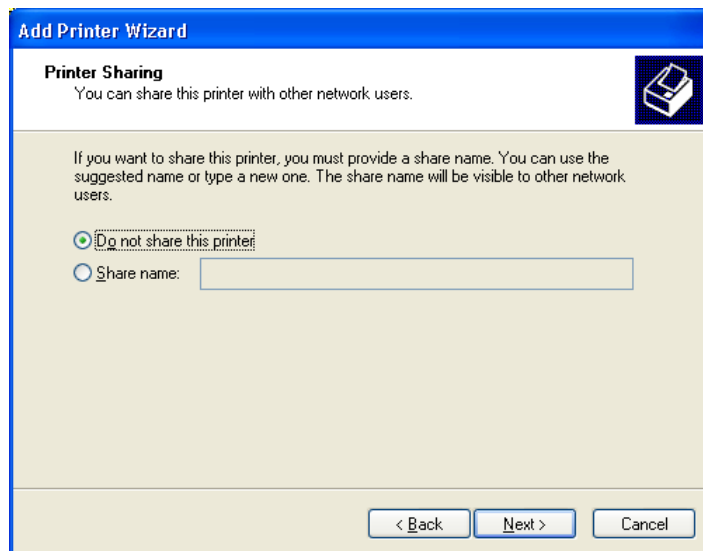
(4) Setting the printer name

Set a name for the printer to be added, and click [Next]. The default printer name "FUJITSU Overlay Create Driver" is specified in the following example.



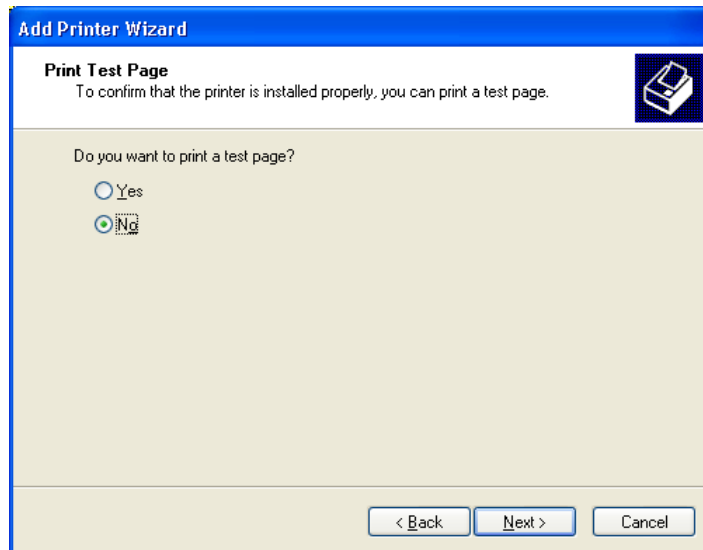
(5) Setting a printer to be not shared

Select [Do not share this printer], and click [Next].

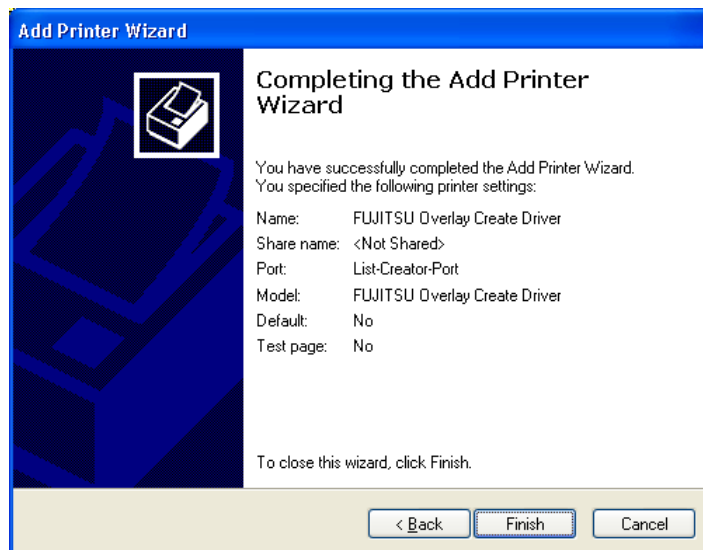


**(6) Printing a test page**

Select [No], and click [Next].

**(7) Completing a printer definition**

Click [Finish] to begin adding the printer.



**Note**

- For Windows XP and Windows Server 2003, when clicking the [Finish] button, a hardware installation message will be shown; select [Continue].

This completes the process of adding a printer.

Now, just by printing from this printer, an overlay and a form can be created.

**15.3.2 Setup (Port Definition Window)**

On the [ListCREATOR Overlay Utility General Settings] window, all operations for overlay creation can be set. Follow the procedures below to perform the setup.

- 1 Logon as a user with Administrators privileges.
- 2 From the [Start] menu, select [Settings] - [Printer].
- 3 Select the printer to be used ("FUJITSU Overlay Create Driver" in this example), then select [File] - [Properties] - [Ports] tab.



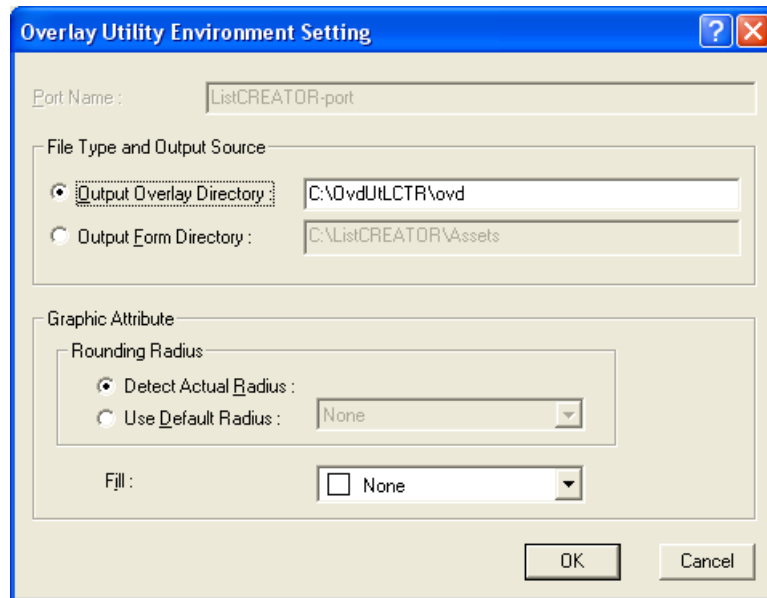
- 4 Click the [Configure Port] button.



**Note**

- For Windows Vista and Windows Server 2008, right-click the printer from Step 3 and select [Run as administrator] - [Properties].
- For Windows 7, use the following procedure to make changes.
  - 1) Click [Start] - [Control Panel] - [System and Security] - [Administrative Tools] to open "Print Management".
  - 2) In the console tree, click [Print Management] - [Print Servers] - [Server Name (local)] - [Ports], and select the name of the port that was added in the previous steps.
  - 3) Select [Operations] - [Configure Port].
  - 4) Set the required information in the ListCREATOR Overlay Utility Environment Setting window, and then click the [OK] button.

The following shows the [ListCREATOR Overlay Utility General Settings] window, and explains its content.



**Port Name**

The name of the port to be used must be set using 6 to 32 characters. A port name is only needed when a port is added. Characters which can be used are shown below.

"0" - "9", "a" - "z", "A" - "Z", ".", "-", "

A port name can not be made of periods only.

**File Type and Output Source**

Specify the type of file to output. You can select either "Overlay" or "Form".

**Overlay Files Directory**

Specify the full path of the overlay output destination directory within 200 bytes. The directory for output must have been set up in advance as shown below.

- Created in the fixed disk.
- For an NTFS format file system, [Change] access privileges must be specified for the user.

This directory will also be used as a work directory. The default value is set as "the ovd directory in the product installation directory".

Refer to the following for details regarding required disk capacity.

=> ["17.1 Points to Note during Printing Forms"](#)

**Output Form Directory**

Specify the full path (up to 200 bytes) of the directory to which to output forms.

It is necessary to create the output directory in advance as indicated below.

- The directory is created on a fixed disk.
- If the file system is NTFS, [Change] access rights are set for the user.

This directory can also be used as a work directory. The default value is set as the "Form Storage Directory" specified in List Creator Designer. Refer to the following regarding required disk space.

=> "17.1 Points to Note during Printing Forms"

#### Template Image Output (Include in output)

Check when outputting a template image file. An output template image file can be processed by the form style descriptor template image function in List Creator Designer, in order to check lines during processing.

When checking this, a template image file is output in the same directory as the output destinations specified in "Overlay Storage Directory" or "Form Storage Directory".

This is not checked by default.

#### Graphic Attributes

Specify rounding radius and shading for overlay conversion objects.

##### Rounding Radius

Specify the rounding radius for an overlay. [Detect Actual Radius] and [Use Default Radius] can be set.

The default value is set to [Detect Actual Radius].

##### Detect Actual Radius

Use this to set [Detect Actual Radius] without setting rounding radius.

##### Use Default Radius

Select [Use Default Radius] to recognize all rounding radius in the same way. After selecting [Use Default Radius], choose rounding radius from among the types listed below.

None: 0dt

Small: 36dt

Medium: 54dt

Large: 72dt

The default value is [None].

##### Fill

Set the overlay conversion shading. Set [None] if not using shading; otherwise, select the type. The possible selections are as follows.

None

5.0%

10.0%

12.5%

25.0%

50.0%

75.0%

100.0%

The default value is [None].

#### Display Output Settings Screen while starting printing

Check to display the output settings window when starting to print. When checked, this window will allow changing settings for each print job.

This is checked by default. Refer to the following for [Output Settings] window.

=> "16.2 Output Settings Window"

#### [OK] button

Enables setting information, and closes the general settings window.

[Cancel] button

Disables information which was set, and closes the general settings window.



**Note**

- Setup content is managed according to each port definition.

# **Chapter 16**

## *Overlay/Form Creation*

---

This chapter explains the actual procedures involved in overlay and form creation.

## 16.1 Form Printing Operation

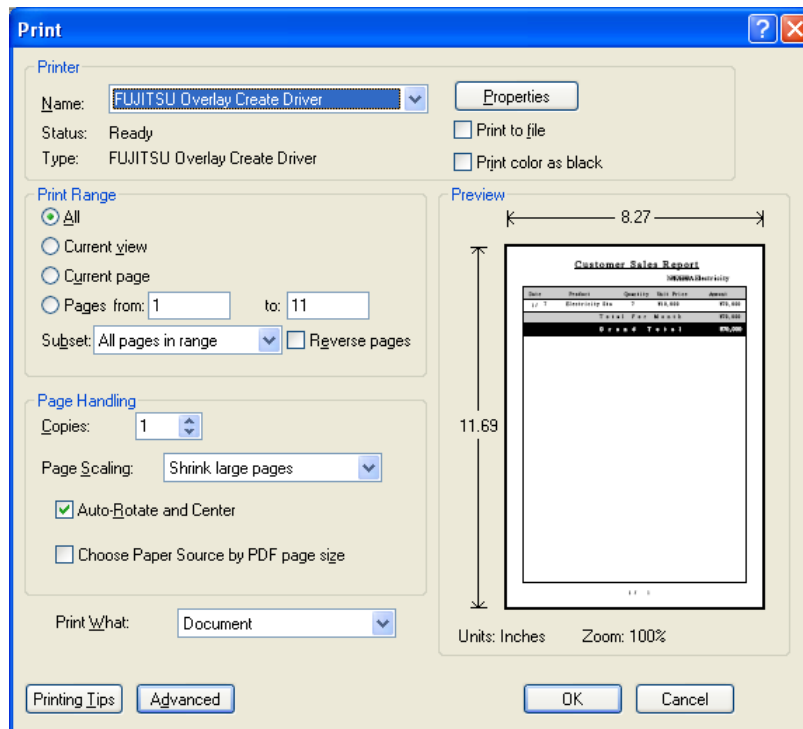
You can print forms (forms created using word processing software or spreadsheet software, or PDF file forms) from the application. When performing this, specify the printer created for the Overlay Utility.

Below is an example of the procedure for a form from a PDF file.

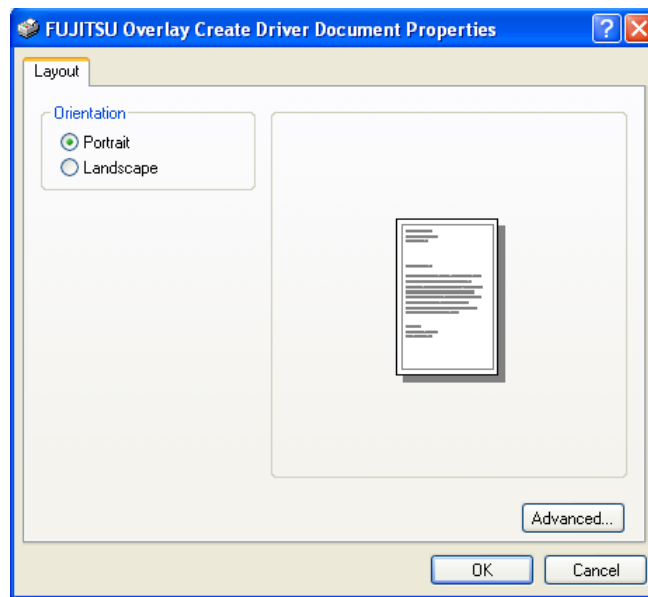
- 1 Activate Adobe Reader, and open the form to be used.



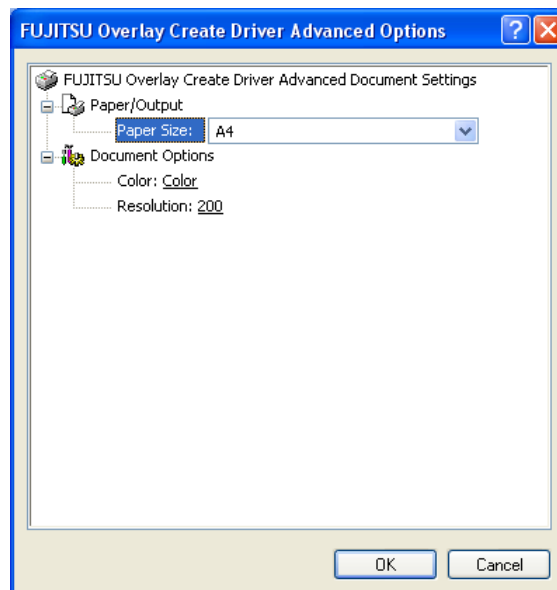
- 2 Select [File] - [Print] from the menu.
- 3 Set the printer to be used for the Overlay Utility. (In this case, "FUJITSU Overlay Create Driver".)



- Click the [Properties] button in order to specify the form type (monochrome/color). The properties window for FUJITSU Overlay Create Driver documents is displayed.



- Click the [Advanced] button to specify the form types (monochrome print/color print) in the [More Options] window.



**Paper Size**

Specify the paper size to be used during output.

**Color**

Specify either monochrome print or color print.

**Resolution**

Specify the resolution.

Monochrome print: Only 400dpi can be specified.

Color print: Either 200dpi or 400dpi can be specified.  
It is recommended that 200dpi is specified.

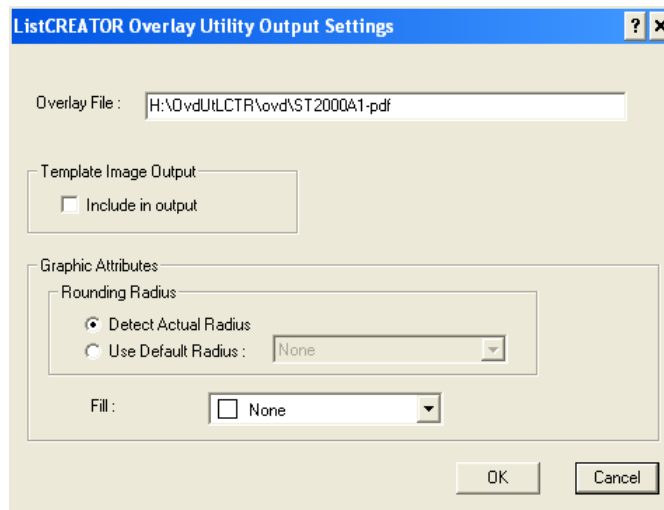
For 400dpi, recognition rate may be degraded. In this case, you can improve the recognition rate by thickening lines.



## 16.2 Output Settings Window

On the [ListCREATOR Overlay Utility General Settings] window, if "Display Output Settings Screen while starting printing" has been checked, the [ListCREATOR Overlay Utility General Settings] window will be shown when starting to print. On this window, the content specified in the setup can be changed for every print request. The content set in the general settings window will be shown as the default value.

The [ListCREATOR Overlay Utility General Settings] window is shown below.



This section explains the overlay name to be output and output template image. For other fields, refer to the following.  
=> "15.3.2 Setup (Port Definition Window)"

### Overlay File

Specify the full path of the overlay name to be output using 250 bytes or less. The overlay name to be output is named by adding the extension ".ovd" to the specified file name.

The directory specified must have been set up in advance as shown below.

- Created in the fixed disk.
- For an NTFS format file system, [Change] access privileges must be specified for the user.

The characters shown below cannot be used.

"/", ".", ":", ";", "\*", "?", "!", "<", ">", "|", "¥"

The name shown below cannot be used regardless of uppercase or lowercase.

"CON", "AUX", "PRN", "COMn (n is 1-9)", "LPTn (n is 1-9)", "NUL"

The default value is "Document Name in the Output Directory specified in Overlay Utility General Settings window".

### Form

Specify the full path of the form name to be output using 250 bytes or less. The files to be output are named by adding the file extensions (pmd/ovd/bip/psf) to the specified form name. The directory specified must have been set up in advance as shown below.

The directory specified must have been set up in advance as shown below.

- Created in the fixed disk.
- For an NTFS format file system, [Change] access privileges must be specified for the user.

The characters shown below cannot be used in a form name.

"/", ".", ":", ";", "\*", "?", "!", "<", ">", "|", "¥"

- "Tilde (~)" as the first character

The name shown below cannot be used regardless of uppercase or lowercase.

"CONFIG", "DEFAULT", "CON", "AUX", "PRN", "COMn (n is 1-9)", "LPTn (n is 1-9)", "NUL"

The directory name specified in the [Output Form Directory] of the Overlay Utility Environment Setting window, plus the printed document name is set as the default value.



Template Image Output (Include in output)

Check when outputting a template image file (bmp). When checking this, a template image file is output in the same directory as the output destinations specified in "Overlay" or "Form".

[OK] button

Processing will continue according to the information set.

[Cancel] button

The information set will be disabled, and processing will be cancelled.

## 16.3 Completion of Overlay/Form Creation

Confirm that the overlay/form has been created in the output directory specified in [Output Setting] window.

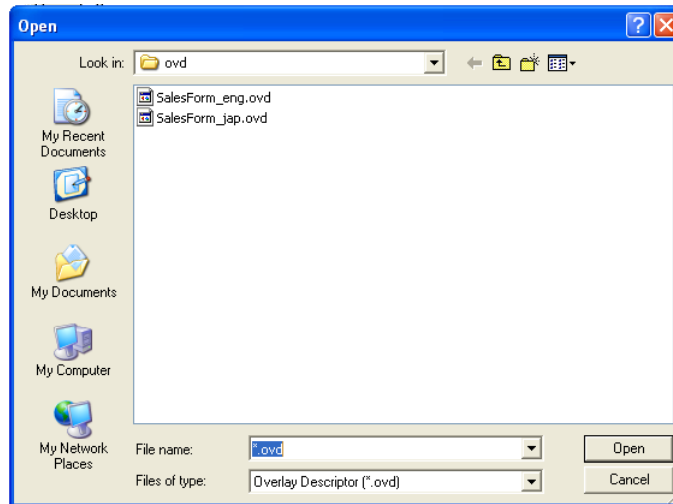
## 16.4 Adding the Overlay to List Creator

Below is an example of the procedure for adding the overlay and the form created with the Overlay Utility to List Creator.

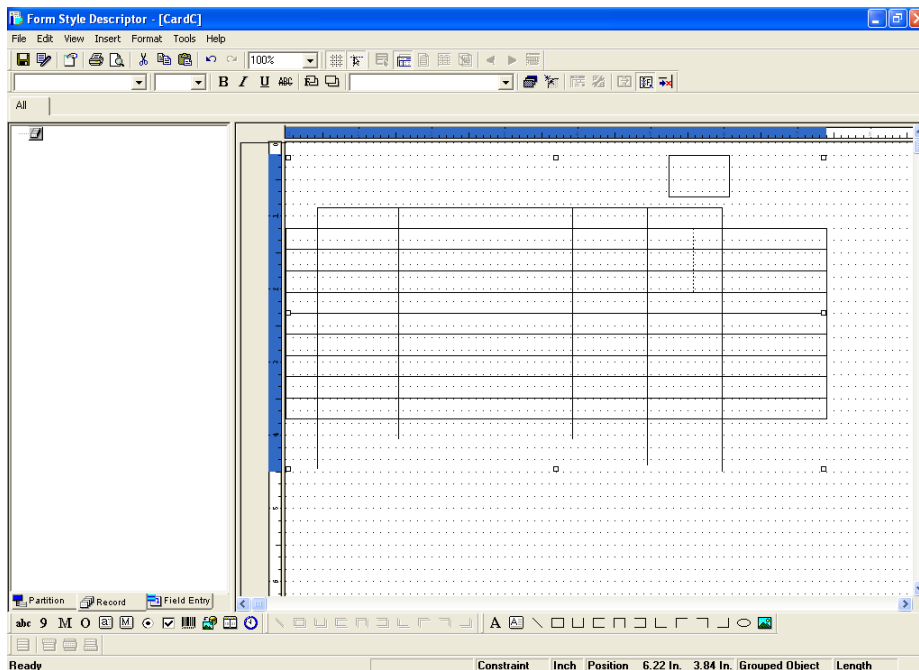
### 16.4.1 Adding Overlay

Run the form style descriptor from the form descriptor list in the List Creator Designer, and select [File] - [Add Overlay Pattern] in order to add the overlay.

The example shown here is "SalesForm\_eng.ovd".



The overlay will be included as shown below.



Adjust the position as needed.

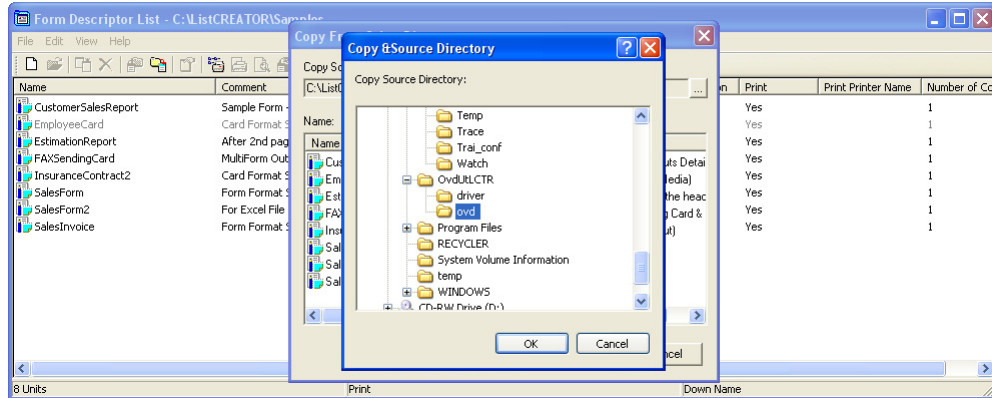


- After overlay addition, the overlay will be grouped. In order to change lines, ungroup the overlay.

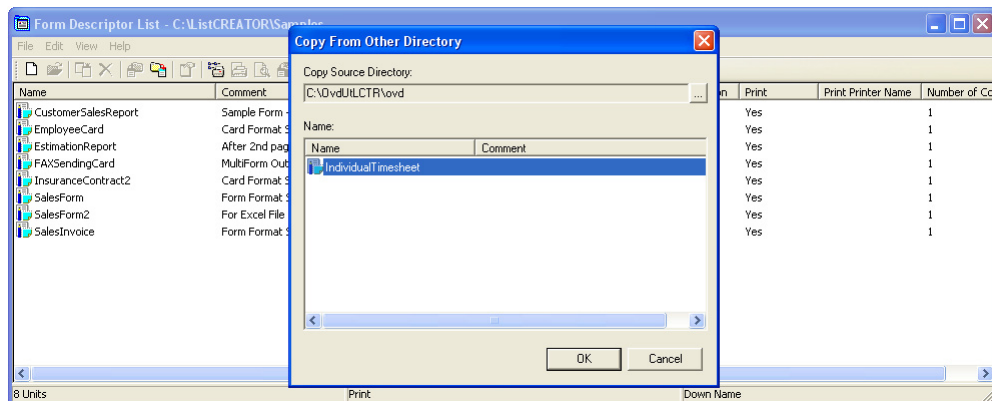
## 16.4.2 Adding Forms

This section describes the procedure for adding forms when the output directory has been changed from the Form Storage Directory.

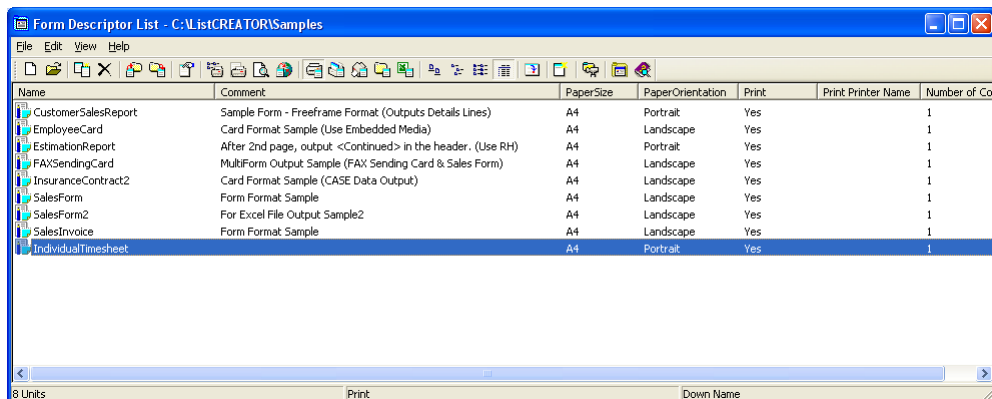
- 1 From the Form Descriptor List in List Creator Designer, select [Edit] - [Copy From Other Directory] - [As is], and then specify the directory in which the form was created for "Copy Source Directory".



- 2 Specify the created form in the Copy Source Directory.



- 3 The created form is added to the [Form Descriptor List] window.



- If the output directory has been changed, select [View] - [Options] in the [Form Descriptor List] window, and then change the Form Storage Directory. The specified directory is displayed in the list.



# **Chapter 17**

## *Points to Note*

---

This chapter provides points to note regarding details of Overlay Utility usage.

## 17.1 Points to Note during Printing Forms

When using the Overlay Utility, be careful of the following points.

- When printing multiple pages, only the first page undergoes overlay conversion. The other pages are ignored. Therefore, if you want to make overlay and form for the other pages, you must print only the target page. Also, for data that contains many pages, processing may take a long time, so we recommend printing only the target page(s).
- During processing, the work file is temporarily created in the output directory set from the [General Settings] window. The following disk space is required for creating the work file.

| Form type  | Resolution | Paper size |      |
|------------|------------|------------|------|
|            |            | A4         | A3   |
| Monochrome | 400dpi     | 2MB        | 4MB  |
| Color      | 200dpi     | 12MB       | 24MB |
|            | 400dpi     | 45MB       | 96MB |

- Sometimes the [Output Setting] window or error message is not displayed on the top page, but is hidden behind another window. In that case, move the other window to show the [Output Setting] window.
- The paper size to be recognized is shown below.
  - Width: From 3.5 to 22.5 inches.
  - Height: From 3.5 to 22.5 inches.
- When you want to use a paper size that is not displayed in document property with the Overlay Utility during printing, set the paper by performing the following procedure.
  - 1 Select the [File] - [Server Properties] in [Printer] window of the Windows to create the paper. Specify the size to be recognized for the paper size.
  - 2 Reboot the computer.
- During processing, if "Cancel" or other operation is performed from the Windows "Printer" window, processing can be stopped. However, if the [Output Setting] window or a message box appears, processing will not stop until a reply is input.
- When "Output" in "Output Template Image" is checked in [General Settings] window, the template image file is created in the resolution specified during printing.
- Remote desktop and terminal services are not supported. Use it on the machine in which the software has been installed.
- If the user has been switched in Windows XP, adding/configuring/deleting ports (Preferences) is not possible.
- Overlay and form creation processing is done in only one request at a time. Therefore, if multiple requests are processed at the same time, the next request will have to wait.
- The format of forms created with the Overlay Utility is "Free Frame". For more details about the Free Frame format, refer to the List Creator Designer Help guide.

## 17.2 Points to Note regarding Overlay Conversion

The points to note below explain details of overlay conversion regarding recognition of rectangles, lines, and color. Items that cannot be recognized should be corrected in the List Creator Designer form style descriptor.

### 17.2.1 Points to Note regarding Line Recognition

This section explains the points to note for line recognition.

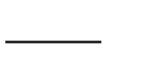
#### 17.2.1.1 Line that can be Recognized

The following lines can be recognized.

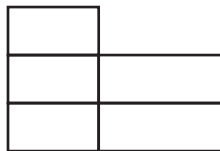
- Solid lines, Dotted lines



- Straight lines (Horizontal, Vertical)



- Rectangles (Except dotted rectangles)



- Shading

| No. | Name |
|-----|------|
|     |      |
|     |      |
|     |      |

- Rounding



Line width for overlay conversion is decided according to form line width, as shown below.

- Form line width in the range between 0.5 - 0.75pt: 0.5pt
- Form line width in the range between 0.76 - 0.75pt: 1.0pt
- Form line width in the range between 1.26 - 1.75pt: 1.5pt
- Form line width 1.76pt or over: 2.0pt

Overlay lines are output in the position where line centers on image and line centers of overlay get together.



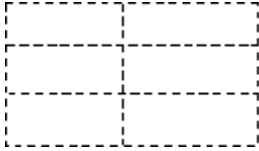
### 17.2.1.2 Lines which cannot be Recognized

Lines which cannot be recognized are shown below.

#### Dotted table lines

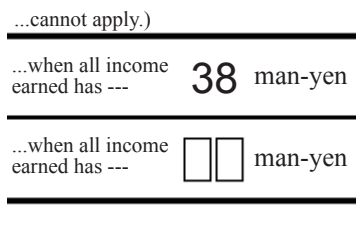
A rectangle with dotted lines for the outer frame cannot be recognized.

In the example below, the outer frame will not be recognized, but the lines within the table will be recognized.



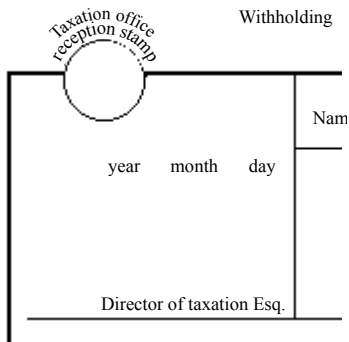
#### Character rectangles

When 2 or more rectangles to insert characters in are output together as shown below, they will not be recognized.



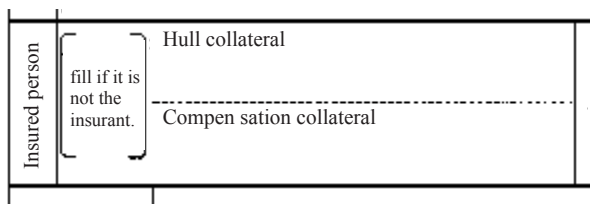
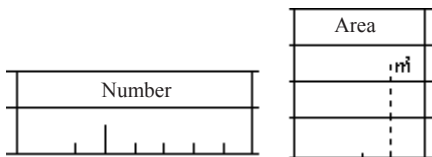
#### Incomplete lines

As in the example below, if a line is incomplete, the line cannot be recognized.



#### Half-lines (Solid, Dotted)

As in the example below, if a line's start point and end point do not match other lines exactly, the lines will not be recognized.



#### Circles and ellipses (Solid, Dotted)

Circles and ellipses are not recognized.

## 17.2.2 Points to Note regarding Recognizing the Color Form



This section explains the points to note regarding recognizing the color form

### 17.2.2.1 Color that can be Recognized

Points to note for color are as follows.

#### The number of colors that can be recognized

The number of colors that can be recognized are 16 (16 basic colors) when outputting an overlay, and 32 (16 basic colors + 16 custom RGB colors) when outputting a form.

When a form with more colors than the number of recognizable colors is recognized, similar colors are recognized as a mean color. Therefore, some colors may not be recognized correctly.

#### Table background color that can be recognized

The number of table background colors that can be recognized are 16 (16 basic colors) when outputting an overlay, and 32 (16 basic colors + 16 custom RGB colors) when outputting a form.

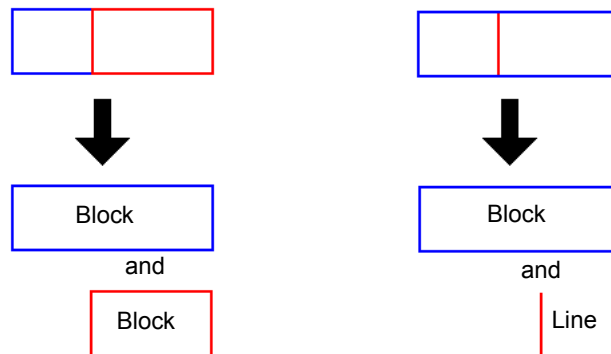
#### Recognition of lines

##### Line color

The number of line colors that can be recognized are 16 (16 basic colors) when outputting an overlay, and 32 (16 basic colors + 16 custom RGB colors) when outputting a form.

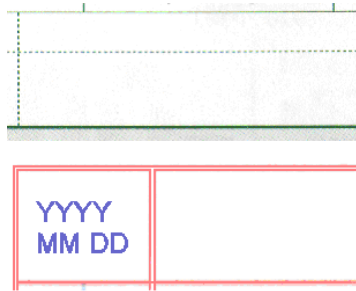
If the line color changes along one side (other than the start/end points) within a cell so that a form is recognized to have two or more colors along a side, a mean color is set for that cell.

In some cases, as shown in the following examples, the outer rectangle of a cell might not be the same color, the entire area might be recognized as a block, or it might be recognized as a block + lines.



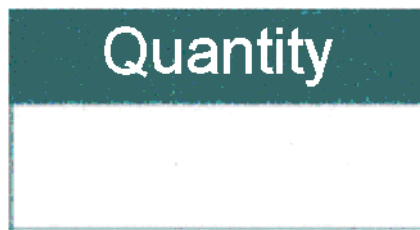
The table with dotted lines and double lines

Dotted lines and double lines, as shown in the following example, may not be recognized properly.



The table whose boundaries are shown by color without lines

As in the example below, the table whose boundaries are shown by color without lines, the boundaries may not be recognized.



Short lines

Short lines, as shown in the following example, may not be recognized.



Recognition of rectangles with color shading

For the following rectangles with color shading, shading parts are recognized as filled. The strings in the shading rectangles may not be recognized properly due to shading.



Recognition of the sheer form

For the sheer forms, the lines of backside may be selected incorrectly.

# Appendixes

---

These appendixes explain the following topics.

- Graph output
- Cautions for designing forms
- Cautions regarding input data
- XML files used in List Creator
- Combination-form descriptor file
- COPY clause file creation function

# **Appendix A**

## *Graph Output*

---

For details on the graph output function summary, graph output method, and cautions, refer to the Graph Utility Help guide.

# **Appendix B**

## *Cautions for Designing Forms*

---

This appendix explains the following cautions when designing forms:

- Cautions for designing forms
- Cautions for each output method
- Cautions when designing forms for multi-form output
- Cautions for importing fields in a file
- Cautions regarding Environment for Designing Forms
- Cautions when using Unicode

## B.1 Cautions for Designing Forms

This section explains the cautions when designing forms.

### B.1.1 Cautions regarding Specification of Form Name

- Do not enter a full-width space at the end of form name. Doing so may result in an error during form output and prevent output.
- When the List Creator print server runs on UNIX OS, pay attention to the characters and the number of characters that can be used for form name.

### B.1.2 Cautions When Changing the Grid Type

- When changing the grid type from "fixed-spacing grid" to "custom-spacing grid"  
If "Dot" is specified for the grid type, it becomes "Dotted line" after the change.  
Also, if "Constraint Mode" is specified as "Past", and "Grid Spacing" is specified as "Input (mm)", the grid spacing values are changed to the following values.

Vertical : 1/6 inch  
Horizontal : 1/10 inch

- When changing the grid type from "custom-spacing grid" to "fixed-spacing grid"  
"Constraint Mode" is specified as "Standard" after the change.

### B.1.3 Cautions regarding Custom-Spacing Grid

Cautions when using a custom-spacing grid are described in the following order:

- Cautions for specifying the default values and unit of a grid
  - Cautions When Operating a Custom-spacing Grid
  - Cautions regarding Snap to Grid
  - Cautions regarding Automatic Correction Position
- **Cautions for specifying the default values and unit of a grid**
    - A custom-spacing grid cannot be specified as the default value of grid. The default grid value is always set to the fixed-spacing grid.
    - Immediately after the grid type is changed to custom-spacing grid, the following values for the grid spacing are set:
 

Vertical: 120 dots  
Horizontal: 360 dots
    - In the Form Style Descriptor, it is recommended to specify "Dot" for the grid unit specification when a custom-spacing grid is used.  
When specifying "Dot", values must be specified as the resolution at 720 dpi conversion.
  - **Cautions When Operating a Custom-spacing Grid**
    - Once the grid is operated, it cannot be restored to the state before the operation by using the [Edit] - [Undo].
    - When the [Form Style Descriptor] window is in the [Layout creation (Fixed partition)] or [Layout Preview] window mode, the grid cannot be operated. The operation must be performed in the [Layout creation(Standard)] window.
    - The minimum and maximum values for the grid spacing are shown below. If the operation results in a value of less than the following minimum value or greater than the following maximum value, the operation is canceled.
 

Minimum value: 4 dots (at 720 dpi conversion)  
Maximum value: 16200 dots (at 720 dpi conversion)
    - To adjust the position or spacing of grids, use the grid handles and ruler segments on the ruler tick area in the edit area of the [Form Style Descriptor] window.

- You cannot select (vertical) ruler segments across multiple partitions. Specify grid settings for each partition.
- When changing the grid spacing in a partition, you cannot change the spacing to be greater than the height of the partition.  
Change the grid spacing to be less than or equal to the height of a partition.

- **Cautions regarding Snap to Grid**

- When a custom-spacing grid is used, the constraint spacing of the grid is also changed according to the changes made to the grid spacing, if any.
- When a custom-spacing grid is used, Snap to Grid is not enabled during the following operations:
  - Changing the position or size in the [Properties] window (property list)
  - Setting the spacing of repeats when specifying repeat of fields (including body area group items)
  - Aligning objects
  - Setting the spacing of repeat copy
  - Changing the size of a partition
  - Changing the start position of a fixed partition
- When you select multiple objects and then move them, change their sizes, or copy them with a mouse operation, the objects are moved or copied so that one of the four (top right, top left, bottom right, and bottom left) corners of the selected area can be placed at the gridline position. Therefore, each object may not be aligned to the gridline positions.

- **Cautions regarding Automatic Correction Position**

- When "All" is specified for "Target Range" in the [Automatic Correction Position] window, all objects in the partition including objects that are not displayed in the edit area are subject to constraint.
- "Fit Character spacing" cannot be specified in the [Automatic Correction Position] window.

## B.1.4 Cautions when defining Overlay Lines and Partition Graphics

This section explains cautions when defining overlay lines, overlay rectangles, overlay text, partition graphics, and partition rectangles.

- When the units for form designing operations are set to "Inch" or "mm", the line may not be vertical/horizontal even if you specify the same value for the start and end positions in the properties of the overlay lines or partition graphics.  
If this is the case, perform one of the following methods to resolve the issue:
  - Hold down the [Shift] key and define (change) the overlay lines or partition graphics
  - Change the units for form designing operations to "Dot" and align the value of the start and end positions
- The character range of the overlay text may vary from the minimal value that can be set by the resolution of the connected printer. If there is a printer driver for output, select the printer driver to output with from the printer settings, set the printer driver resolution to the value to use during output, then design the form.  
Also, be sure to confirm the output results before running.

## B.1.5 Cautions regarding the Transparency of Image Fields

When specifying the transparency color from the image file, note the following points:

- Image data of two-color expressed in RGB can be output, and supports only the following formats.
  - Bitmap data
  - TIFF data
  - PNG data

If unsupported image data is used, problems occur such as an error occurs during the form output, or images are output without transparency.



When printing forms that specify transparency for image fields, note the following points:

- When printing in monochrome printers, the media color may not be recognized (colors close to white become white and colors that are close to black become black). Therefore, the printouts may not produce the desired results. To workaroud this phenomenon, perform the following:
  - Use monochrome images
  - Use a color printer rather than a monochrome printer
- For monochrome printers, the printouts may not always be completed; images may be chopped or the quality is poor (grainy). To workaroud this phenomenon, perform the following:
  - Increase the printer resolution.
  - Change the drawing method using STAMPTYPE (transparency output specification of image) in the form output information file (or form output environment setting file)  
For details, refer to the Online Manual "Setup & Operation".
- Since the draft quality varies depending on the output printer, ensure the draft is correct before actually running the system.

### B.1.6 Output Order of Parts (Objects)

- **Output Order of Different Types of Parts (Objects)**

Parts (objects) are output in the following order (1 to 2 to 3).

- 1 Overlays
- 2 Partition graphics
- 3 Fields

Therefore, when the parts (objects) are overlapped, the following parts (objects) are output in foreground.

| Overlapped parts (objects)      | Parts (objects) output in foreground |
|---------------------------------|--------------------------------------|
| Overlays and partition graphics | Partition graphics                   |
| Overlays and fields             | Fields                               |
| Partition graphics and fields   | Fields                               |



**Note**

- When outputting to an Excel file, refer to the description on the overlapping of fields in the Online Manual "Setup & Operation".

- **Output order of same type of parts (objects)**

- Output order of overlays

Overlays are output in the order defined in the edit window for Form Style Descriptor.

Output order of overlays can be changed. For the methods to change the output order of overlays, refer to the section "Overlap of Fields/Overlays" in the Designer Help guide.

- Output order of partition graphics

Partition graphics are output in the order defined in the partition.

- Output order of fields

Fields are output from the top left corner to the lower right corner using the base point of each field located in each area as reference point. Output order of fields can be checked based on the order of the partition list in the Form Style Descriptor.

If some field base points have same coordinates, the field which is displayed lower in the record list in the Form Style Descriptor is output earlier. Therefore, the field which is uppermost in the list will be output in foreground.



**Caution**

- The field which attribute is changed by moving in the list or changing properties, may be displayed in a position different from the actual output order in the partition list in the Form Style Descriptor, because it is displayed on top of the fields having the same base point coordinates. To check the actual output order, close the Form Style Descriptor window after moving a field or changing any field attribute, and open the window again to update the partition list..



**Note**

- The field base point varies according to the type of the field.  
The base point of the following fields is the bottom-left corner.

- Numeric field
- Text field
- OCR-B field
- Radio button
- Check box
- Static field
- Date field
- Time field

The base point of the following fields is the top-left corner.

- Block text field
- Barcode field
- Image field
- Block static field

## B.2 Cautions for Each Output Method

This section explains the cautions when designing forms for each operation setting when outputting form descriptor information.

- **Cautions Common to All Output Methods**

The available functions and the expressions vary depending on the output method and the operating system of the List Creator print server.

Refer to the following manuals:

- Online Manual "Setup & Operation" attached with List Creator on the List Creator print server
- Manuals of each related software product

- **Cautions when Designing Forms to be saved as e-Record**

Before designing forms to be saved as e-Record, check the unavailable functions and the cautions regarding the operations.

For details, refer to the Japanese Online Manual.

- **Cautions when Designing Forms to be Faxed**

Before designing forms to be Faxed, check the unavailable functions and the cautions regarding the operations.

For details, refer to the Japanese Online Manual.

- **Notes when Designing Forms to Save as PDF**

Before designing forms to save as PDF file, check the unavailable functions and the cautions regarding the operations.

For details, refer to the Online Manual "Setup & Operation".

- **Cautions when Designing Forms to Output to Excel File**

Before designing forms to be output to Excel files, check the unavailable functions and the cautions regarding the operations.

For details, refer to the Online Manual "Setup & Operation".

## B.3 Cautions when Designing Forms for Multi-Form Output

Note the following points when designing a form for multi-form output.

- The following fields must use the same values in all forms to be used for the multi-form outputs:
  - Paper Type
  - Paper Size
  - Orientation
  - Print Area
  - Multi-Up Printing
  - Input Data Format
  - Separator Character
  - Block Text Field New Line Codes
  - "Alternative text of Control Code" in the [Barcode] tab of the [Form Properties] window.
  - "Fill with a blank if data length does not reach field length" in the [Barcode] tab of the [Form Properties] window.
- When specifying the variable "%" for the SET operator, the same replacement strings specified during the form output are applied to all forms used in multi-form output.
- For the form output methods, including the output method, the specification defined in the first form to be output will become valid.

However, depending on the output interface of the form being used, the following form output methods of each form are enabled:

- Paper source/feed name when printing
- Specification of duplex printing option when printing
- Specification to feed a new sheet of paper when a new form starts printing for multi-form output
- Paper source/feed name when output the forms after making e-Record
- Specification to feed a new sheet of paper when a new form starts outputting the forms for multi-form output after making e-Record

Some printers do not allow the changing of paper sources for each form. Make sure to check the operations of the printer actually being used.

## B.4 Cautions for Importing Fields in a File

The following explains the cautions when importing fields in a file.

- Immediately after a field info file is imported, the "Record List (Data List)" is displayed on the list pane.
- Once a field info file is imported, it cannot be restored to the state before the import by using [Undo] of the [Edit] menu.
- After a field info file is imported, the actions prior to the import cannot be "undone" or "redone".
- A field info file is not imported when the names of the following fields specified in the field info file are the same as those in the existing form:
  - Group item
  - Static field
  - Block static field
  - Date field
  - Time field
  - U.S. Postal FIM Field

- If an error occurs while a field info file is being imported, the condition prior to the import will be restored.
- If a form in multipart table format is being updated and any RH (report header) partition is defined in it, the [Data Definitions] window is displayed as shown below:
  - All the fields added by the field info file are displayed on the [Data Definitions] tab (not on the [Report Header] tab).

To handle the hidden fields as RH (report header) partition fields, move them to the [Data Definitions (Report Header)] tab in the [Data Definitions] window.

- For an existing form that contains any group items, when a field that is part of a group item precedes a field to be added, the added field will also be part of the group item.
- When no field precedes a field to be added in the field info file, the added field will be inserted as the first field (under the root).
- Any field layer in an existing form remains as specified in the existing form. Therefore, fields are sorted only within each layer.
- A field info file containing more than 2000 fields cannot be imported.
- If the total number of pre-defined fields and fields to be added by importing a field info file exceeds 2000, the field info file will not be imported.
- Even when a field info file is imported, data definitions operands and extended definitions are not changed.

## B.5 Cautions regarding Environment for Designing Forms

This section explains the cautions regarding environment for designing forms.

### B.5.1 Cautions regarding Access Privileges

This section explains cautions regarding access privileges.

- If the user does not have update privileges to the form storage directory and each file of the form descriptor information (bip, pmd, psf, ovd, dse), the files cannot be loaded. Be sure to grant update privileges.
- If the user has update privileges to the form storage directory, but no update privileges to each file of the form descriptor information (bip, pmd, psf, ovd, dse), a message will display when importing the files, and the files cannot be updated. Be sure to grant update privileges.
- For both the form descriptor information and the form storage directory, make sure to set access privileges so that users are able to create, read, modify and delete. If access privileges are not set to allow creation, reading, modification and deletion of files, forms may not display in the [Form Descriptor List] window, and display, save, and output of individual forms may fail.

### B.5.2 Cautions regarding the Location (Source) of Form Resources

This section explains cautions regarding the location (source) of form resources.

- If form descriptor information on a network is modified, the data may not be defined. Be sure to place the form descriptor information in the local directory before modifying it.
- The directory in which List Creator is installed cannot be specified as the form storage directory.
- If form descriptor information on a network is edited (updated), the data may not be defined. Be sure to place the form descriptor information in the local directory before editing (updating) it.

## **B.6 Cautions when using Unicode**

For notes or restrictions on designing form descriptor information that allows Unicode characters, refer to the Designer Help guide.

For cautions when using multi-language forms, refer to the descriptions on multi-language form operation in the Online Manual "Setup & Operation".

# **Appendix C**

## *Cautions regarding Input Data*

---

This appendix explains the cautions regarding input data.

For the points to note when using input data in "XML Format", refer to the following:

=> ["D.9Points"](#)



## C.1 Operation when Changing the Input Data Format

This section explains how the data list and record list of the [Form Style Descriptor] window change when changing the input data format from the [General] tab of the [Data Definitions] window.

Changing the input data format creates or deletes data root names and tag name. Also, it switches the display list and partially changes the data structure.

### C.1.1 When Changing from Variable Length, Fixed Length, or CSV Format to XML Format

The following occur when changing the input data format from a variable length, fixed length, or CSV format to an XML format:

- **Display list**

The record list switches to the data list.

- **Data root element**

Created as "DataRoot".

- **Tag Name**

The tag name is created based on the field name.

If the tag name cannot be assigned, an auto-generated name is set.



**Note**

- The tag name of List Creator is to be specified within 128 characters which consisted of Japanese characters (available symbol is only katakana-hiragana prolonged sound mark (?)), half-width alphanumeric characters and some symbols (A-Z, a-z, 0-9, hyphen (-), period (.) and underscore (\_)). A name including both Japanese characters and half-width alphanumeric characters can be used.

However, the following restrictions apply:

- Full-width alphanumeric characters or full/half-width spaces cannot be used for a tag name.
- Any symbols other than underscore (\_), or half-width numeric characters cannot be used at the beginning of a tag name.
- Whether the character is uppercase or lowercase, "xml" or "fxml" cannot be used at the head of a tag name.

- **Data structure**

The following apply to data structures for form layout data of free frame formats:

- All group items defined in the record list are deleted.
- The sequence of fields in the data list is as follows:
  - For each header, frame(\*), body, and footer, fields are ordered from the top left corner to the lower right corner (this is the same order as the partition list) using the bottom left corner of the field as a base point. (\*) For a form with multiple frames, frames are ordered from the top left corner to the lower right corner (this is the same order as the partition list) using the top left corner of the frame as a base point.
  - Add-on data of a barcode field is placed after the barcode field.
  - Hidden fields (records) are added to the end of the data list.
- For each frame, a frame group item and details group item are created.

The tag names and field names of the frame group item and details group item are assigned in the following manner:

- **Frame group item**

The frame name is assigned to the tag name.

The default field name is the frame name. However, if the frame name is already applied to a field, the name is set as "FRMnnnn (nnnn being a sequential number)".

- **Details group item**

The "DE" is assigned to the tag name.


The default field name is "DE". However, if "DE" is already assigned as a field name, the "nnnn (sequential number)" is appended to the default name.

- All the fields defined in a frame become configuration items of the corresponding frame group item. All the fields defined to a DE (Details) partition of a frame become configuration items of the details group item of the corresponding frame.

For form layout data for formats other than the multipart table format and free frame format, the structure of the record list is simply transferred to the data list.

The following apply to data structures for form layout data of multipart table formats:

- All group items defined in the record list are deleted.
- The sequence of fields in the data list is as follows:
  - Starting from the top partition of the edit area, fields in each partition are ordered from the top left corner to the lower right corner (this is the same order as the partition list) using the bottom left corner of the field as a base point.
  - Add-on data of a barcode field is placed after the barcode field.
  - Hidden fields (records) are added to the end of the data list.
- A details group item is created. The "DE" is assigned to the tag name of the details group item. The default name of the details group item is "DE". However, if "DE" is already assigned as a field name, the "nnnn (sequential number)" is appended to the default name.
- All the fields defined to a DE (Details) partition become configuration items of the details group item.

|   |   |
|---|---|
| <br><b>Caution</b> | <ul style="list-style-type: none"> <li>- If "Oracle" is set as the [Database Type] of the [Database Link] tab in the [Data Definitions] window, the input data format cannot be set to "XML Format".</li> <li>- If repeats are specified for fields other than static fields and block static fields for form layout data of the single page format, the input data format cannot be changed to "XML Format".</li> <li>- For form layout data of the free frame format or multipart table format, the input data format cannot be changed to "XML Format" for the following situations:                     <ul style="list-style-type: none"> <li>• When no visible fields are defined to a DE (Details) partition (or for free frame formats, when no visible fields are defined to the DE (Details) partitions of each frame).</li> <li>• When a clone field exists.</li> <li>• If changing to "XML Format" results in the number of fields to exceed 2000 (which includes the frame group items and details group items that are created along with other defined fields).</li> </ul> </li> </ul> |
|---|---|

## C.1.2 When Changing from XML Format to Variable Length, Fixed Length, or CSV Format

The following occur when changing the input data format from an XML format to a variable length, fixed length, or CSV format:

- **Display list**  
The data list switches to the record list.
- **Data root element**  
This is deleted.
- **Tag Name**  
This is deleted.
- **Data structure**
  - The structure of the data list is simply transferred to the record list.
  - Details group items become standard group items.

---

## C.2 Cautions for Code Type

This section explains the cautions regarding the code type of input data when designing forms.

For the characters available in List Creator and the cautions regarding the character code, refer to the Online Manual "Setup & Operation".

For notes regarding form descriptor information which can define Unicode, refer to the Designer Help guide.

### C.2.1 When the Input Data is Shift-JIS

When the input data is Shift-JIS code, pay attention to the following points when designing the form and input data.

- When designing forms

- For performing form output using the input data in "XML format", calculate the total number of bytes as three bytes per Japanese character (including half-width katakana) and one byte per half-width alphanumerical character, and specify the result as the data length. If the data length to be specified is not sure because Japanese and half-width alphanumerical characters are mixed, multiply the number of characters by three and specify the result as the data length.

The string may get clipped during the output if the data length is too short.

- When designing input data

- NULL values (0x00) cannot be specified in the input data.
- When the input data format is "XML Format", register the user defined characters in UNICODE (for printing, saving as PDF file, or outputting to Excel file).
- Do not specify the code which contains 0xFF in the input data, like 0xFF or 0xFFnn.  
If the code contains 0xFF, it will result in garbage characters or error and there is no guarantee of operation.

### C.2.2 When the Input Data is UNICODE (UTF16)

When the input data is UNICODE (UTF16), pay attention to the following points when designing the form and input data.

- When designing forms

- For the data length, calculate the total number of bytes as three bytes per Japanese character (including half-width katakana) and two bytes per half-width alphanumerical character. If the data length to be specified is not sure because Japanese and half-width alphanumerical characters are mixed, multiply the number of characters by three and specify the result as the data length.

The string may get clipped during the output if the data length is too short.

- In the condition expression of CASE output operator, or in the output condition for DE (Details Area Partition) specified in DETAIL (Details Area Section) in the Extended definition, half-width yen sign "¥" or half-width overline "¯" does not match half-width yen sign (0x00A5) or half-width overline (0x00AF) in the input data, respectively.

A half-width yen sign "¥" or a half-width overline "¯" specified in the CASE output operator or the DETAIL output condition in the Extended definition matches a half-width backslash (0x005C) or a half-width tilde (0x007E) in the input data, respectively.

- When designing input data

- If the input data format is "Fixed Length" and the "Yes" setting is specified for the separator character, specify a separator character code of 0x007F or less in the input data.
- If "Variable Length" input data is used when specifying a custom separator character, specify a character code of 0x007F or less.
- When creating input data with UNICODE (UTF16) code type and the input data format is "CSV Format", "Variable Length" or "Fixed Length", do not put BOM (Byte Order Mark) at the beginning of the data.  
BOM: 0xFFFE (Little Endian), 0xFEFF (Big Endian)

### C.2.3 When the Input Data is UNICODE (UTF8)

When the input data is UNICODE (UTF8) code, pay attention to the following points when designing the form and input data.

- When designing forms

- For the data length, calculate the total number of bytes as three bytes per Japanese character (including half-width katakana) and one byte per half-width alphanumerical character. If the data length to be specified is not sure because Japanese and half-width alphanumerical characters are mixed, multiply the number of characters by three and specify the result as the data length.  
The string may get clipped during the output if the data length is too short.
- Do not specify a half-width yen sign (0x00A5) or a half-width overline (0x00AF) in the [Block Text Field New Line Codes].
- In the condition expression of CASE output operator, or in the output condition for DE (Details Area Partition) specified in DETAIL (Details Area Section) in the Extended definition, half-width yen sign "¥" or half-width overline "¯" does not match half-width yen sign (0xC2A5) or half-width overline (0xC2AF) in the input data, respectively.  
A half-width yen sign "¥" or a half-width overline "¯" specified in the CASE output operator or the DETAIL output condition in the Extended definition matches a half-width backslash (0x5C) or a half-width tilde (0x7E) in the input data, respectively.

- When designing input data

- If "Variable Length" input data is used when specifying a custom separator character, specify a character code of 0x7F or less.
- If the input data format is "Fixed Length" and the "Yes" setting is specified for the separator character, specify a separator character code of 0x7F or less in the input data.
- When creating input data with UNICODE (UTF8) code type and the input data format is "CSV Format", "Variable Length" or "Fixed Length", do not put BOM (Byte Order Mark) at the beginning of the data.  
BOM: 0xEF (1st byte) + 0xBB (2nd byte) + 0xBF (3rd byte)
- Do not specify the code which contains 0xFF in the input data, like 0xFF or 0xFFnn.  
If the code contains 0xFF, it will result in garbage characters or error and there is no guarantee of operation.

### C.2.4 When the Input Data is EUC

When the input data is EUC code, pay attention to the following points when designing the form and input data.

- When designing forms

- For the data length, calculate the total number of bytes as two bytes per Japanese character (including half-width katakana) and one byte per half-width alphanumerical character. If the data length to be specified is not sure because Japanese and half-width alphanumerical characters are mixed, multiply the number of characters by two and specify the result as the data length.  
The string may get clipped during the output if the data length is too short.

- When designing input data

- If "Variable Length" input data is used when specifying a custom separator character, specify a character code of 0x7F or less.
- If the input data format is "Fixed Length" and the "Yes" setting is specified for the separator character, specify a separator character code of 0x7F or less in the input data.
- Do not specify the code which contains 0xFF in the input data, like 0xFF or 0xFFnn.  
If the code contains 0xFF, it will result in garbage characters or error and there is no guarantee of operation.

# **Appendix D**

## *XML Files Used in List Creator*

---

In List Creator, an XML file can be used as input data when outputting forms. The form layout can be created by referring to the XML file and/or the DTD file as the field information to be defined in the form.

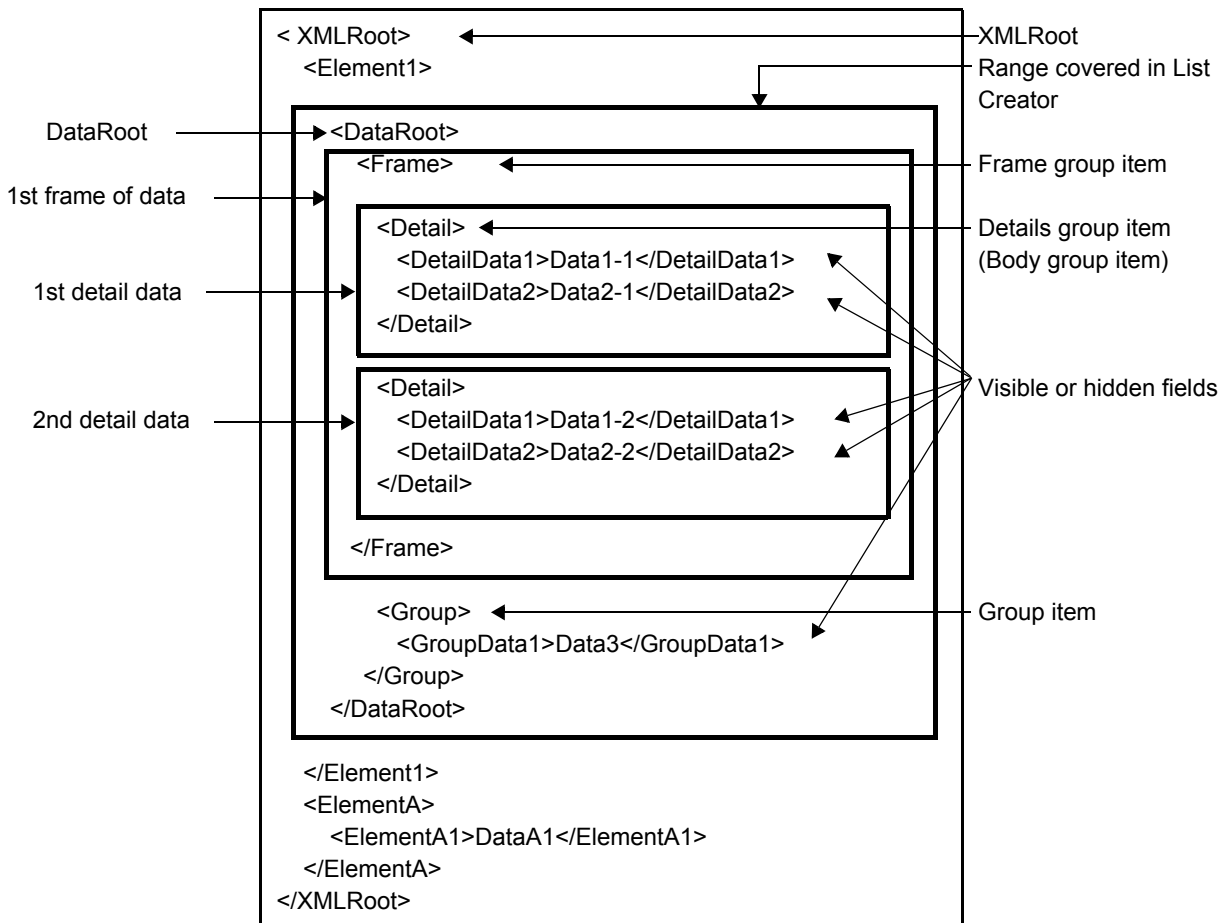
This appendix explains the following details of XML files and DTD files that can be used in List Creator.

- Terms regarding XML files used in List Creator
- Conditions under which XML files and DTD files can be used in List Creator
- Specifying the Form Format responding to the XML Data Configuration
- Conditions under which Elements can be Specified as Data Root
- Conditions under which Elements can be Specified as Body Group Items
- Conditions under which Elements can be Specified as Details Group Items
- Conditions under which Elements can be Specified as Frame Group Items
- Data Root and Page Feed
- Cautions

## D.1 Terms regarding XML Format Files

This section explains the terms regarding XML format files used in List Creator.

- XML root  
The highest element.
- Data root  
The highest element handled in List Creator. One element is specified for each form.
- Body group item  
This is an element which organizes the child elements of a detail under the data root element in simple table, block repeat, or sticky label forms. One element is specified for each form. This is assigned to the body area.
- Details group item  
This is an element which organizes the child elements of a detail under the data root element in multipart table forms.  
One element is specified for each form. This is assigned to the details partition.
- Frame group item  
This is an element which organizes the child elements of a detail in a frame in free frame forms.  
One element is specified for each frame. This is assigned to the details partition of a frame.
- Frame group item  
This is an element which organizes the child elements of a frame under the data root element in free frame forms. The same number of elements as the frames specified for each form is specified. This is assigned to a frame.
- Group item  
An element which has a child element.
- Output field  
An element without a child element, which is output to a form.
- Hidden field  
An element without a child element, which is not output to a form.



## D.2 Conditions under which XML Files and DTD Files can be Used in List Creator

This section explains the conditions under which XML files and DTD files can be used in List Creator.



- XBRL is a kind of XML, however, since XBRL is a special format of XML using technology called XLink, it cannot be used as XML file in List Creator.

### D.2.1 Character codes of XML Files and DTD Files that can be Used in List Creator

In List Creator, XML files and DTD files written in the following character codes can be used. In an XML file, however, UNICODE (UTF8) is to be used when omitting the XML declaration or the setting of the character code at the XML declaration. In a DTD file, UNICODE (UTF8) is to be used when text declaration is omitted.

- UNICODE (UTF8)
- Shift-JIS
- Japanese EUC (EUC-JP) (only when designing forms, or outputting forms on UNIX OS)

### D.2.2 When using an XML File

This section explains the conditions under which XML files can be used as the input data when outputting forms or the field information defined in the form at designing.

- **Common (when designing the form/outputting the form)**

This section explains the common conditions when referring to the XML file during form designing or when using the XML file during form output.

- Well-formed XML documents  
A well-formed XML document is an XML document in which the XML instances are written according to the defined XML tag rules, with matching start and end tags, and nesting of child elements within their parents.

- No mixed content exists  
Mixed content is an element of which child element holds both data and element.  
The following shows an example of the mixed content.

```
<Element1>
  Data
  <Element2/>
</Element1>
```

- No attribute specified  
When the attribute is specified, it will be unloaded though it is not processed as an error (When outputting the form, the data will be output if the element name is specified in the form. When designing the form, only the element name is to be loaded).  
The following shows an example of the attributes.

```
<FirstName Furigana="Nihon">Nihon</FirstName>
      ↑
      Attribute
```

- No element name which violates the tag naming rules exists  
The tag name of List Creator is to be specified within 128 characters which consisted of Japanese characters (available symbol is only katakana-hiragana prolonged sound mark (ー)), half-width alphanumeric characters and some symbols (A-Z, a-z, 0-9, hyphen (-), period (.) and underscore (\_)). A name including both Japanese characters and half-width alphanumeric characters can be specified.



**Caution**

- Full-width alphanumeric characters or full/half-width spaces cannot be used for a tag name.
- Any symbols other than underscore (\_), or half-width numeric characters cannot be used at the beginning of a tag name.
- Whether the character is uppercase or lowercase, "xml" or "fjxml" cannot be used at the head of a tag name.

- An element which can be specified as a data root exists  
For the conditions under which an element can be specified as a data root, refer to the following:  
=> ["D.4 Conditions under which Elements can be Specified as a Data Root"](#)
- For simple table, block repeat or sticky label forms, an element which can be specified for a body area group item exists  
For the condition under which an element can be specified as a body area group item, refer to the following:  
=> ["D.5 Conditions under which Elements can be Specified as Body Group Items"](#)
- For multipart table forms, an element which can be specified for a details group item exists  
For the conditions under which an element can be specified for a details group item, refer to the following:  
=> ["D.6 Conditions under which Elements can be Specified as Details Group Items"](#)
- When specifying a frame in a free frame form, an element which can be specified for a frame group item and details group item exists  
For the conditions under which an element can be specified for a frame group item, refer to the following:  
=> ["D.7 Conditions under which Elements can be Specified as Frame Group Items"](#)  
For the conditions under which an element can be specified for a details group item, refer to the following:  
=> ["D.6 Conditions under which Elements can be Specified as Details Group Items"](#)

- **When designing forms**

This section explains the conditions required for designing the form and referring to the XML file.

- Any namespaces are not used  
Because identification by namespace is not performed, operations will vary as follows depending on the namespace specification method:
  - When using the default namespace  
When an element using a default namespace exists, only the element name is to be loaded ignoring the namespace of which belongs to.
  - When using prefix of the namespace for an element name ("Prefix: Element\_name")  
When an element with the prefix namespace exists, the element will be failed and unloaded.
  - When using prefix of the namespace for an attribute name ("Prefix: Attribute\_name")  
Even when the prefix of space name is attached on the attribute name, only the element name will be loaded, meanwhile, the attribute name will be unloaded.
- The number of element layers under one XML root is within 48
- The number of elements under one XML root is within 1995

- **When outputting forms**

This section explains the conditions required for outputting the form and using the XML file.

- Any namespaces are not used  
Because identification by namespace is not performed, operations will vary as follows depending on the namespace specification method:
  - When using the default namespace  
When an element using the default namespace exists, only refer to the element name ignoring the space name of which belong to. If the element name is defined in the form, the data will be output.
  - When using prefix of the namespace for an element name ("Prefix: Element\_name")  
When an element with the prefix namespace exists, the element data will not be output since it is to be unloaded.



- When using prefix of the namespace for an attribute name ("Prefix: Attribute\_name")  
Even when the prefix of space name is added on the attribute name, only the element name will be referred to. If the element name is specified in the form, the data will be output. Attributes of the data will not be loaded.
- The number of elements under one data root is within 48

### D.2.3 When using a DTD File

This section explains the conditions under which a DTD file can be used as field information which is to be specified in the form when designing forms.

- No mixed content exists  
Mixed content is an element of which child element holds both data and element.  
The following shows an example of the mixed content.

```
<!ELEMENT Element (#PCDATA|Element1)*>
```

- ANY declaration does not exist  
The following shows an example of the ANY declaration.

```
<!ELEMENT Element ANY>
```

- No attribute specified  
When the attribute is specified, it will be unloaded though it is not processed as error.  
The following shows an example of the attributes.

```
<!ELEMENT First name (#PCDATA)>  
<!ATTLIST First Name Furigana CDATA #IMPLIED> <-Attribute
```

- The number of element layers within the highest element is within 48
- The number of elements within the highest element is within 1995
- No element name which violates the tag naming rules exists  
The tag name of List Creator is to be specified within 128 characters which consisted of Japanese characters (available symbol is only katakana-hiragana prolonged sound mark (ー)), half-width alphanumeric characters and some symbols (A-Z, a-z, 0-9, hyphen (-), period (.) and underscore (\_)). A name including both Japanese characters and half-width alphanumeric characters can be specified.



#### Caution

- Full-width alphanumeric characters or full/half-width spaces cannot be used for a tag name.
- Any symbols other than underscore (\_), or half-width numeric characters cannot be used at the beginning of a tag name.
- Whether the character is uppercase or lowercase, "xml" or "fjxml" cannot be used at the head of a tag name.

- An element which can be specified as a data root exists  
For the conditions under which an element can be specified as a data root, refer to the following:  
=> ["D.4 Conditions under which Elements can be Specified as a Data Root"](#)
- For simple table, multipart table, free frame, block repeat or sticky label forms, no elements with same name exist in one element hierarchy  
The following shows an example when elements with same name exist in an element hierarchy.

```
<!ELEMENT Element (Child_element,Child_element)>
```

- For simple table, block repeat or sticky label forms, an element which can be specified for a body area group item exists  
For the condition under which an element can be specified as a body area group item, refer to the following:  
=> ["D.5 Conditions under which Elements can be Specified as Body Group Items"](#)

- For multipart table forms, an element which can be specified for a details group item exists  
For the conditions under which an element can be specified for a details group item, refer to the following:  
=> "D.6 Conditions under which Elements can be Specified as Details Group Items"
- When specifying a frame in a free frame form, an element which can be specified for a frame group item and details group item exists  
For the conditions under which an element can be specified for a frame group item, refer to the following:  
=> "D.7 Conditions under which Elements can be Specified as Frame Group Items"  
For the conditions under which an element can be specified for a details group item, refer to the following:  
=> "D.6 Conditions under which Elements can be Specified as Details Group Items"

## D.2.4 Range of Input Data for each form format (per Page, per Detail)

The range of input data and input unit used in List Creator differs by form format type.  
The following tables show the range of input data and input unit used in List Creator by form format type respectively.

**Table D.1** Range of one page or one detail of the input data in XML format (free frame format)

| Form format       |   | Data Unit             | Range per unit       | Remarks  |
|-------------------|---|-----------------------|----------------------|--|
| Free Frame format | When no frame is defined  | 1 page                | 1 data root          | -  |
|                   | When one frame is defined   | 1 detail              | 1 details group item | - Data other than the data located in the frame group items of the data root will be header, body, and footer data.<br>- Data other than the data located in the details group items of the frame group item will be CH (Control Header) and CF (Control Footer) partition data. |
|                   | When the multiple frames are defined and the all frames are linked in one |                       |                      |  |
|                   | When the multiple frames are defined and the link is not set              | 1 output frame detail | 1 details group item | - Data other than the data located in the frame group items of the data root will be header, body, and footer data.<br>- Data other than the data located in the details group items of the frame group item will be CH (Control Header) and CF (Control Footer) partition data. |
|                   | When multiple frames are defined and multiple links are set               |                       |                      |  |

**Table D.2** Range of one page or one detail of the input data in XML format (except for free frame format)

| Form format            | Data Unit | Range per unit         | Remarks  |
|------------------------|-----------|------------------------|--|
| Single page format     | 1 page    | 1 data root            | -  |
| Simple table format    | 1 detail  | 1 body area group item | Data other than the data located in the body area group items of the data root will be heads and tails of data.  |
| Multipart table format | 1 detail  | 1 details group item   | Data other than the data located in the details group items of the data root will be data of report header, page header, control header, control footer and page footer partitions (RH, PH, CH, CF, PF). |
| Block repeat format    | 1 block   | 1 body area group item | Data other than the data located in the body area group items of the data root will be heads and tails of data.  |
| Sticky label format    | 1 label   | 1 body area group item | Data located in the body area group items of the data root will only be enabled.   |

## D.3 Specifying the Form Format responding to the XML Data Configuration

When the format of the input data is "XML format", the form format is selected according to the data configuration of the XML file.

For the data units of each form format, refer to the following:

=> ["Table D.2 Range of one page or one detail of the input data in XML format \(except for free frame format\)"](#)

If the XML file only contains one page worth of data or if the data is in single page units, select a single page form. Refer to the following:

=> ["D.3.1 If there is only Page Worth of Data or If the Data is in Single Page Units"](#)

If the XML file contains multiple details of data (an undetermined amount of data), select a form format after determining the following conditions:

- If the detail data is not layered
  - For only detail data  
The sticky label format is recommended. Refer to the following:  
1) of ["If the detail data is not layered"](#) in => ["D.3.2 If there are Multiple Details of Data \(an Undetermined Amount of Data\)"](#)
  - If there is data other than detail data such as header information within the same hierarchy as the detail data  
The simple table or block repeat format is recommended. Refer to the following:  
2) of ["If the detail data is not layered"](#) in => ["D.3.2 If there are Multiple Details of Data \(an Undetermined Amount of Data\)"](#)
  - If there is data other than detail data such as header information in the hierarchy above the detail data  
The multipart table or free frame format is recommended. Refer to the following:  
3) of ["If the detail data is not layered"](#) in => ["D.3.2 If there are Multiple Details of Data \(an Undetermined Amount of Data\)"](#)
- If the detail data is layered  
The multipart table or free frame format is recommended. Refer to the following:  
=> ["D.3.2 If there are Multiple Details of Data \(an Undetermined Amount of Data\)"](#) in ["If the detail data is layered"](#)
- If there are multiple layered detail data  
The free frame format is recommended. Refer to the following:  
=> ["D.3.2 If there are Multiple Details of Data \(an Undetermined Amount of Data\)"](#) in ["If there are multiple layered detail data"](#)

### D.3.1 If there is only Page Worth of Data or If the Data is in Single Page Units

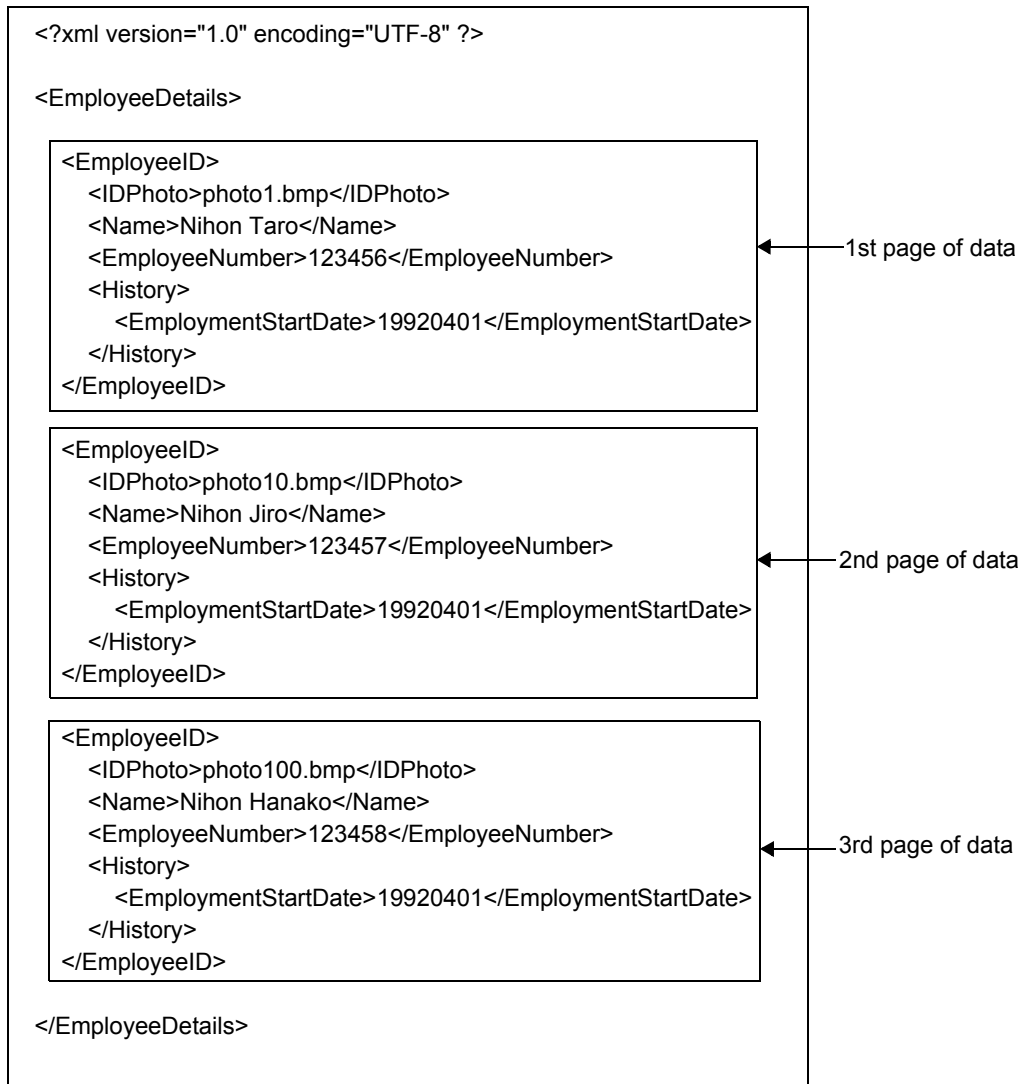
If the XML file only contains one page worth of data or if the data is in single page units, select a free frame form without any frames defined or a single page form.

- Sample XML data with one page of data

```
<?xml version="1.0" encoding="UTF-8" ?>

<EmployeeID>
  <IDPhoto>photo.bmp</IDPhoto>
  <Name>Nihon Taro</Name>
  <EmployeeNumber>123456</EmployeeNumber>
  <EmploymentStartDate>19920401</EmploymentStartDate>
</EmployeeID>
```

- Sample XML data in single page units



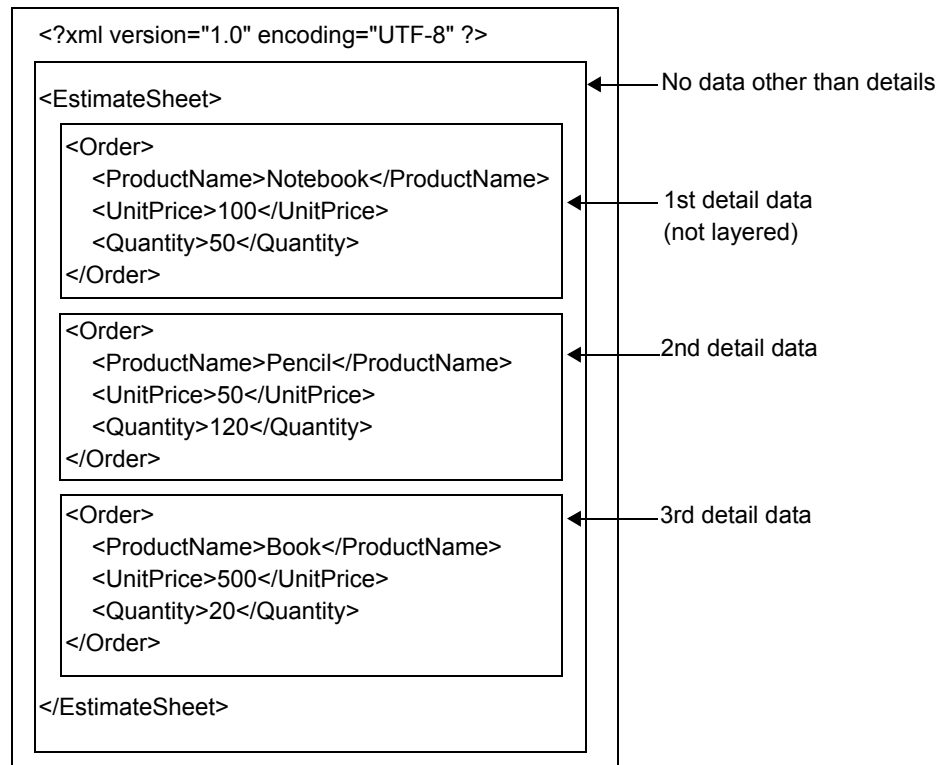
### D.3.2 If there are Multiple Details of Data (an Undetermined Amount of Data)

If there are multiple details of data (an undetermined amount of data) in the XML file, the following form formats are recommended:

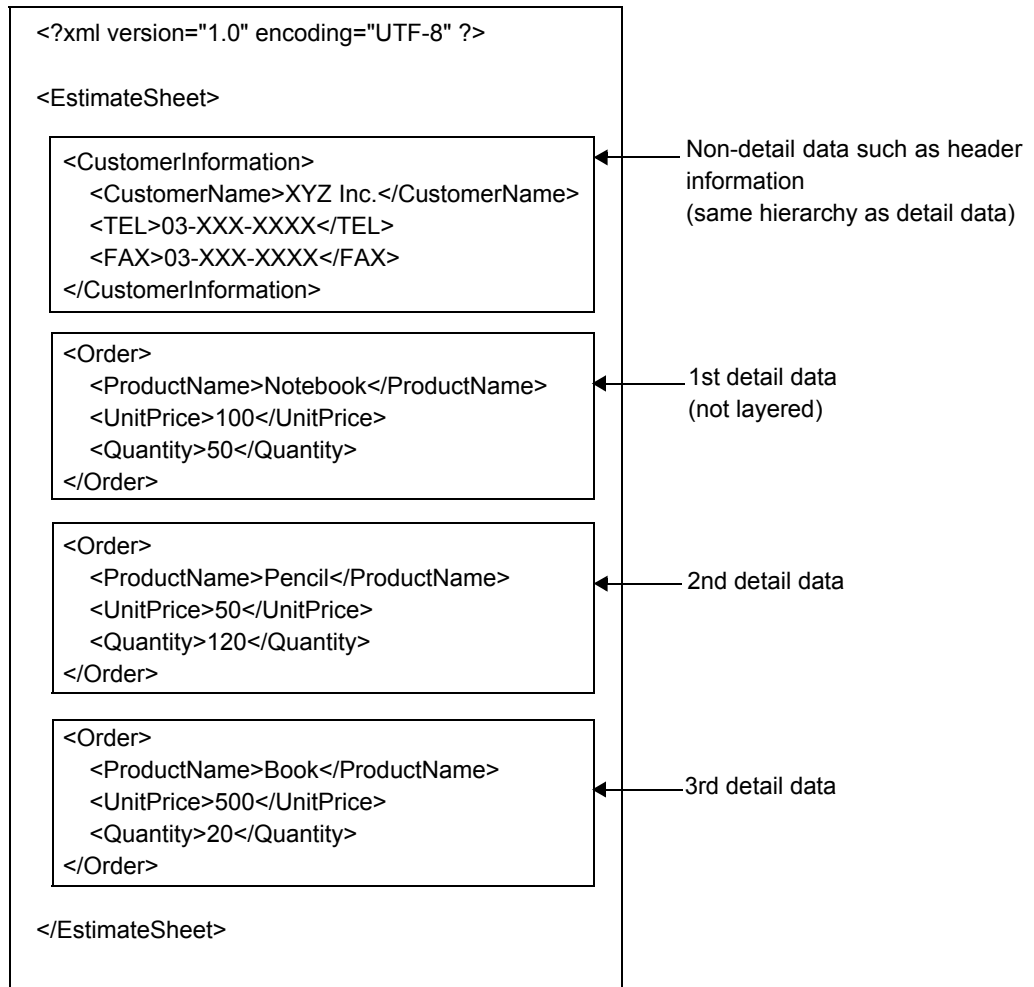
- If the detail data is not layered

If the detail data is not layered, refer to the following three samples and select the format of the form to create.

- 1) For only detail data  
Select the sticky label format.



- 2) If there is data other than detail data such as header information within the same hierarchy as the detail data  
Select a simple table or block repeat form.



- 3) If there is data other than detail data such as header information in the hierarchy above the detail data  
Select a multipart table or free frame form.



- If the detail data is layered

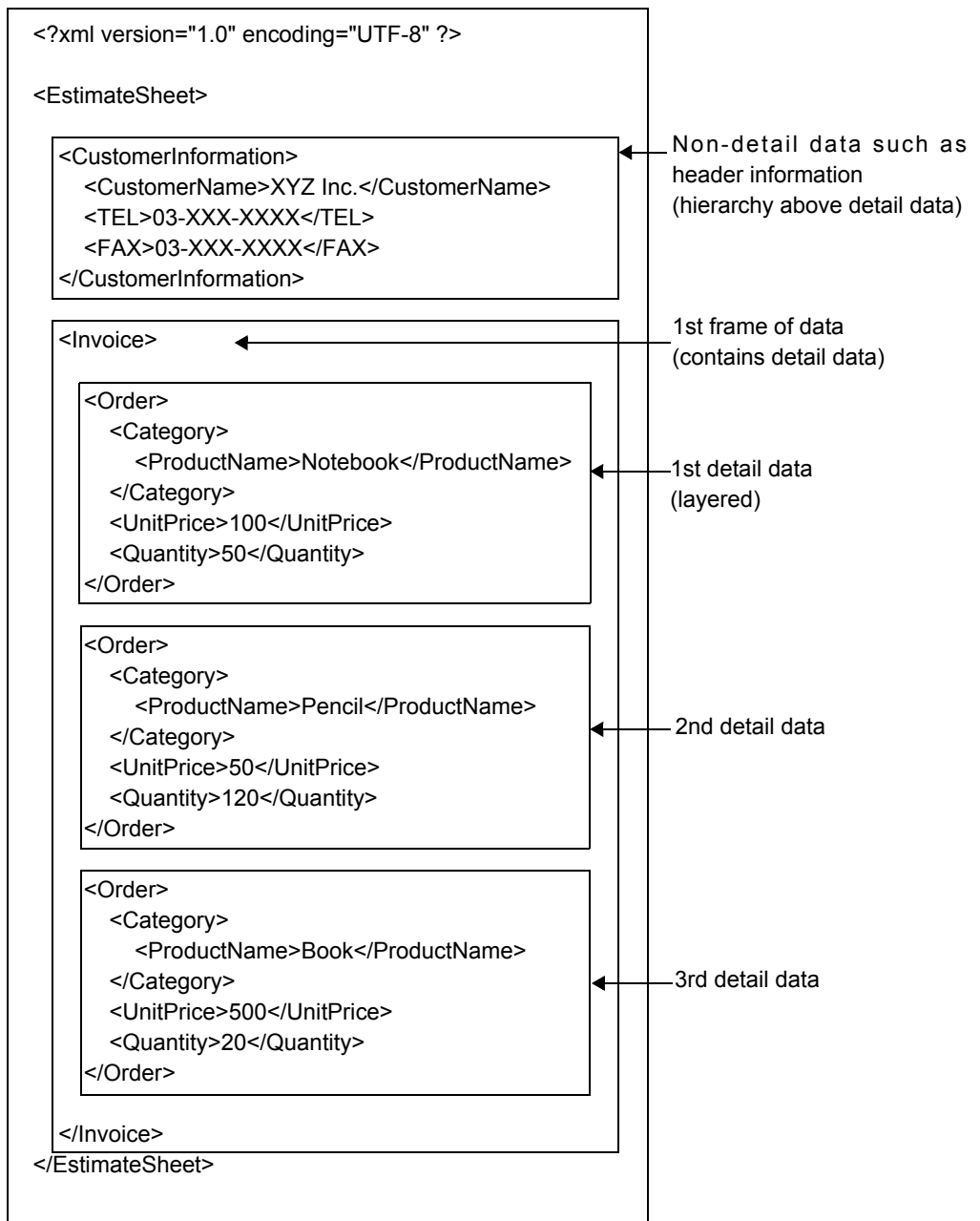
If the detail data is layered, select a multipart table or free frame form.

For example, for the following data, select a multipart table form.



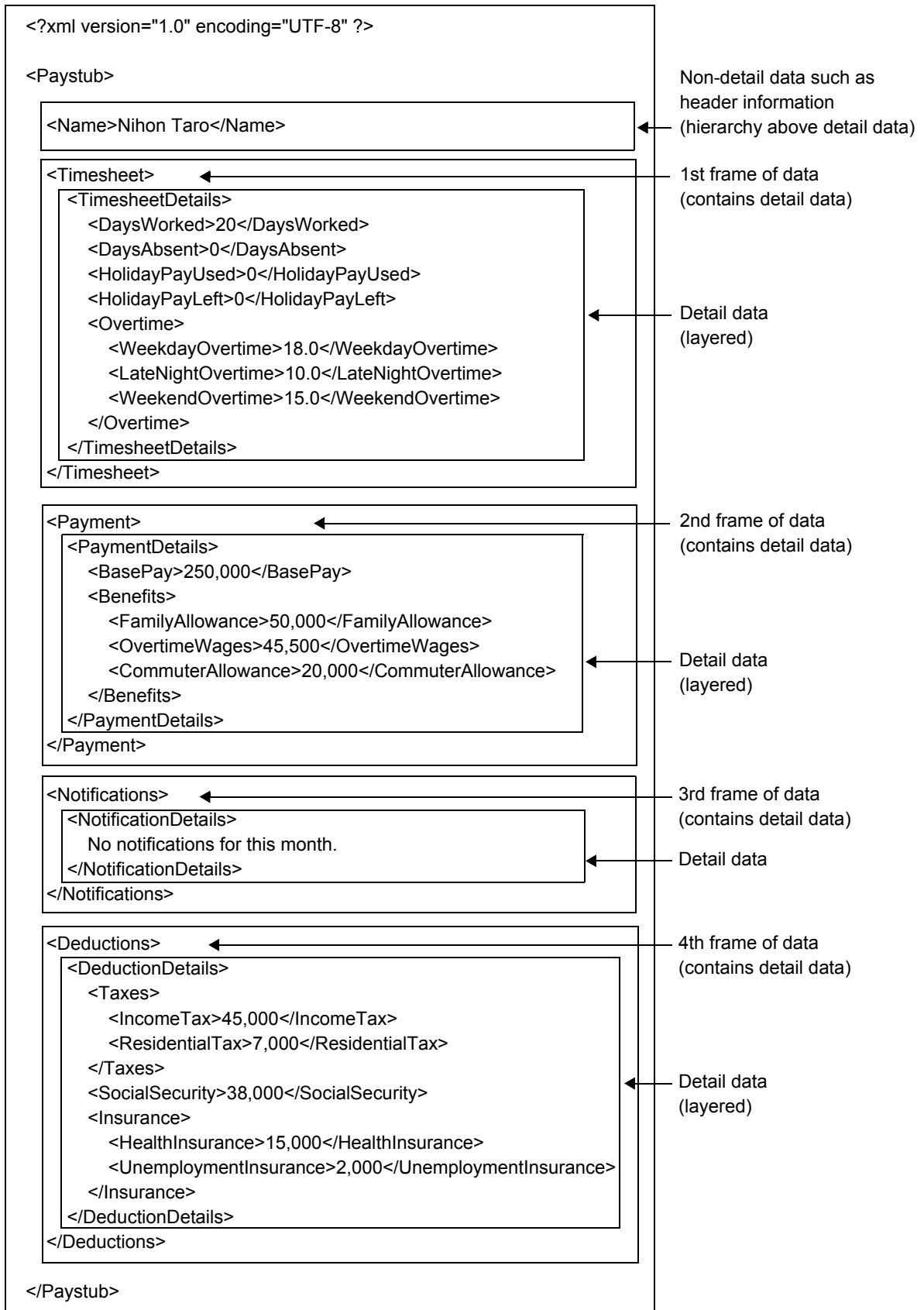


If the detail data is under the same tag such as shown below, you can also select a free frame form.



- If there are multiple layered detail data

If there are multiple layered detail data, select a free frame form.



## D.4 Conditions under which Elements can be Specified as a Data Root

The data root refers to the highest element in the XML file or DTD file handled by List Creator. This section explains the conditions under which elements can be specified as a data root.

Elements that satisfy the following conditions can be specified as the data root.

- Conditions under which XML files or DTD files can be used in List Creator are met  
=> ["D.2 Conditions under which XML Files and DTD Files can be Used in List Creator"](#)
- The element has child elements

Example element that can be specified as a data root (XML file):

```
<DataRoot>
  <Child element/>
</DataRoot>
```

Example element that cannot be specified as a data root (XML file):

```
<DataRoot>
</DataRoot>
```

Example element that can be specified as a data root (DTD file):

```
<!ELEMENT DataRoot (Child element)>
```

Example element that cannot be specified as a data root (DTD file):

```
<!ELEMENT DataRoot (#PCDATA)>
```

- For simple table, block repeat or sticky label forms, an element which can be specified for a body area group item exists under the data root  
=> ["D.5 Conditions under which Elements can be Specified as Body Group Items"](#)
- For multipart table forms, an element which can be specified for a details group item exists under the data root  
=> ["D.6 Conditions under which Elements can be Specified as Details Group Items"](#)
- When specifying a frame in a free frame form, an element which can be specified for a frame group item and details group item exists  
For the conditions under which an element can be specified for a frame group item, refer to the following:  
=> ["D.7 Conditions under which Elements can be Specified as Frame Group Items"](#)  
For the conditions under which an element can be specified for a details group item, refer to the following:  
=> ["D.6 Conditions under which Elements can be Specified as Details Group Items"](#)

## D.5 Conditions under which Elements can be Specified as Body Group Items

A body area group item is an element under the data root and has a collection of child elements of a single detail. This is assigned to the body of a simple table, block repeat, and sticky label form.

The following displays an example XML file and DTD file. In the examples, <Body> tag is specified as the body area group item.

Example XML file:

```
<DataRoot>
  <Body>
    <Data1/>
    <Data2>Data</Data2>
    <Data3/>
  </Body>
</DataRoot>
```

Example DTD file:

```
<!ELEMENT DataRoot (Body*)> *1
<!ELEMENT Body (Data1, Data2, Data3)> *2
<!ELEMENT Data1 (#PCDATA)>
<!ELEMENT Data2 (#PCDATA)>
<!ELEMENT Data3 (#PCDATA)>
```

Elements that satisfy the following conditions can be specified as the body area group item.

- Common (when designing the form/outputting the form)
    - Conditions under which XML files or DTD files can be used in List Creator are met  
=> ["D.2 Conditions under which XML Files and DTD Files can be Used in List Creator"](#)
    - It is under the data root (see \*1 in the above example DTD file)
    - It contains child elements (see \*2 in the above example DTD file)
    - It does not contain grandchild elements (see \*3 in the above example DTD file)
    - For sticky label forms, it does not contain child elements other than body area group items under the data root
- The following shows an example:

Example element that can be specified as a body area group item (XML file):

```
<DataRoot>
  <Body>...</Body>
</DataRoot>
```

Example element that cannot be specified as a body area group item (XML file):

```
<DataRoot>
  <Head>...</Head>
  <Body>...</Body>
  <Tail>...</Tail>
</DataRoot>
```

Example element that can be specified as a body area group item (DTD file):

```
<!ELEMENT DataRoot (Body*)>
```

Example element that cannot be specified as a body area group item (DTD file):

```
<!ELEMENT DataRoot (Head,Body*,Tail)>
```

- When designing forms

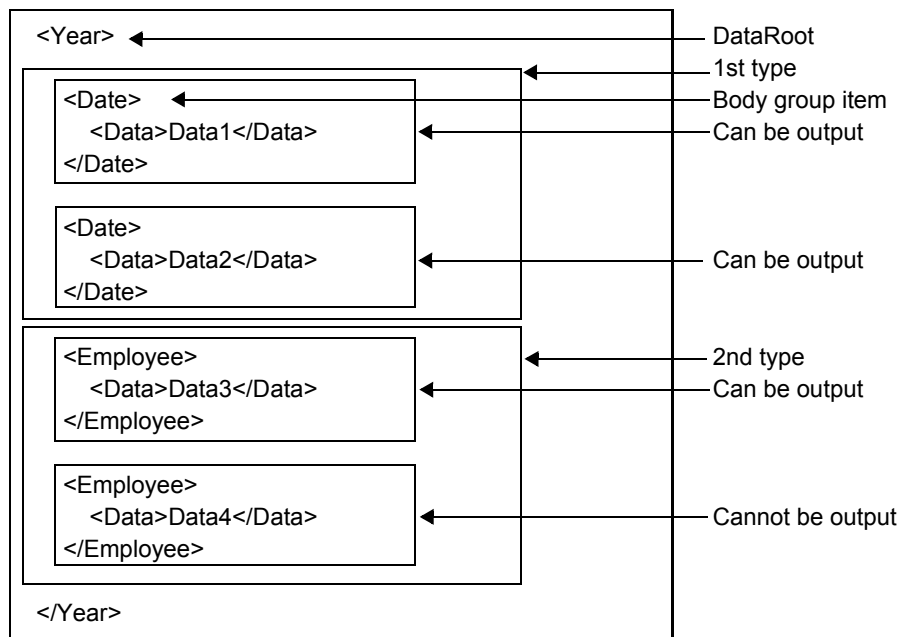
- When referencing DTD files, it contains child elements that can occur multiple times (has a "+" or "\*") (\*1 of "D.5 Conditions under which Elements can be Specified as Body Group Items" in Example DTD file:)

- When outputting forms

- There is only one type of element that has a collection of child elements of a detail (i.e. an element that is a body area group item)  
 If there are two or more types of elements that have a collection of child elements of a detail (i.e. an element that is a body area group item), all the data of the first element will be output, however, only the first data of the second element will be output.

The following is an example where there are two types of elements that have a collection of child elements of a detail (i.e. an element that is a body area group item).  
 In this example, there are two tags (<Date> tag and <Employee> tag) that are elements containing child elements for a single detail (i.e. body area group items). All the data of the <Date> tag will be output, however, only the first data of the <Employee> tag will be output.

For example:



## D.6 Conditions under which Elements can be Specified as Details Group Items

A details group item is an element under the data root and has a collection of child elements of a single detail. This is assigned to the details partition of a multipart table or free frame form.

The following displays an example XML file and DTD file. In the examples, <Detail> tag is specified as the details group item.

Example XML file:

```
<DataRoot>
  <MultipartTable>
    <Detail>
      <Data1>Data</Data1>
      <Data2>
        <Data21>Data</Data21>
        <Data22>Data</Data22>
      </Data2>
    </Detail>
  </MultipartTable>
</DataRoot>
```

Example DTD file:

```
<!ELEMENT DataRoot (MultipartTable*)>
<!ELEMENT MultipartTable (Detail*)> ← *1
<!ELEMENT Detail (Data1,Data2)> ← *2
<!ELEMENT Data1 (#PCDATA)>
<!ELEMENT Data2 (Data21, Data22)>
<!ELEMENT Data21 (#PCDATA)>
<!ELEMENT Data22 (#PCDATA)>
```

Elements that satisfy the following conditions can be specified as a details group item.

- Common (when designing the form/outputting the form)
  - Conditions under which XML files or DTD files can be used in List Creator are met  
=> ["D.2 Conditions under which XML Files and DTD Files can be Used in List Creator"](#)
  - It contains child elements (see \*2 in the above example DTD file)
- When designing forms
  - When referencing DTD files, it contains child elements that can occur multiple times (has a "+" or "\*")  
(see \*1 in the above example DTD file)

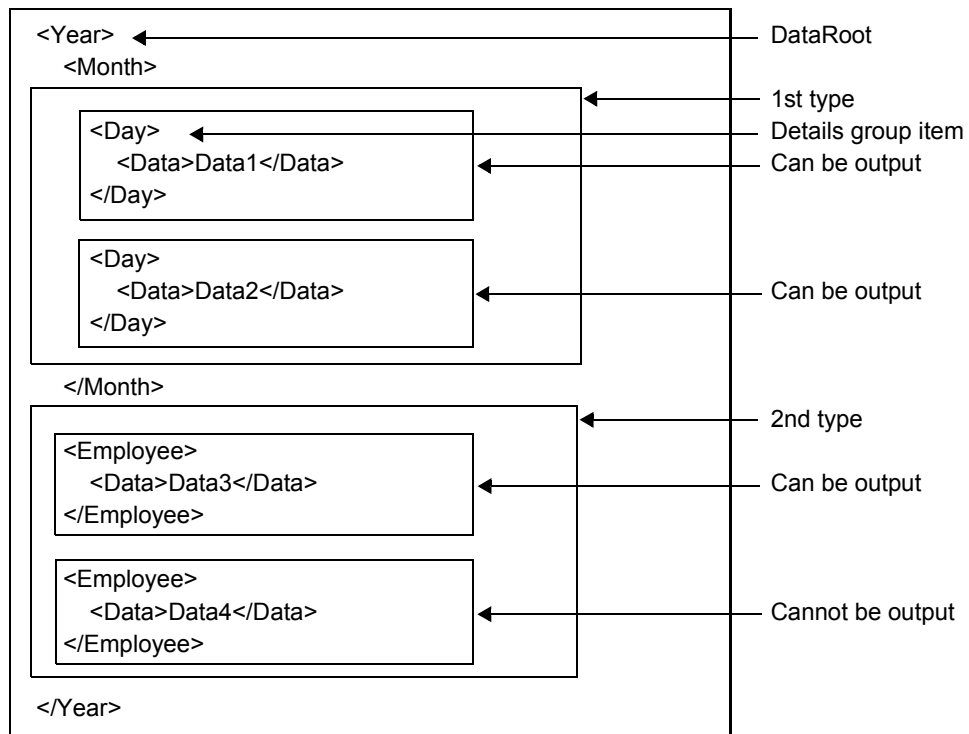
• When outputting forms

- There is only one type of element that has a collection of child elements of a detail (i.e. an element that is a details group item)  
 If there are two or more types of elements that have a collection of child elements of a detail (i.e. an element that is a details group item), all the data of the first element will be output, however, only the first data of the second element will be output.

The following is an example where there are two types of elements that have a collection of child elements of a detail (i.e. an element that is a details group item).

In this example, there are two tags (<Day> tag and <Employee> tag) that are elements containing child elements for a single detail (i.e. details group items). All the data of the <Day> tag will be output, however, only the first data of the <Employee> tag will be output.

For example:

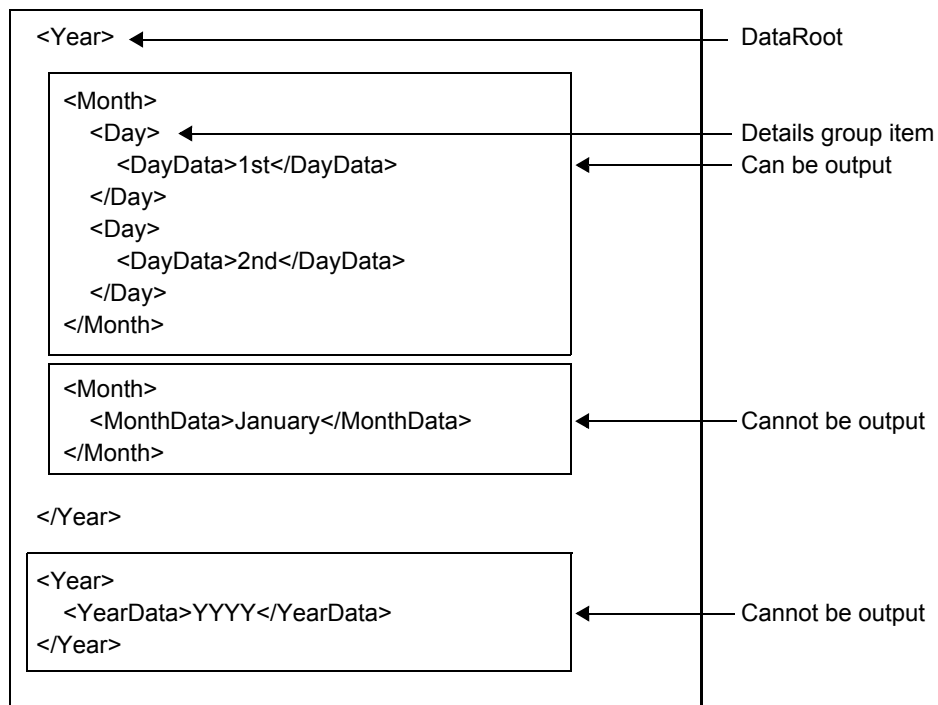


- In the same hierarchy of any element above the element that contains the child elements of a single detail (i.e. an element that is a details group item), there are no elements with the same name but with different data structures under them

The following is an example where in the same hierarchy of any element above the element that contains the child elements of a single detail (i.e. an element that is a details group item), there are elements with the same name but with different data structures under them.

In this example, there are two tags (<Year> tag and <Month> tag) with different data structures under them in the same hierarchy. All the data of the <DayData> tag under the first <Year> tag and <Month> tag will be output, however, the data of the <YearData> tag under the second <Year> tag and the <MonthData> tag under the second <Month> tag will not be output.

For example:



**Caution** - One detail can be output for one details group item. Therefore, if the elements specified for a details group item have a hierarchical structure, specify the element in the lowest level of the hierarchy as the details group item.



## D.7 Conditions under which Elements can be Specified as Frame Group Items

A frame group item is an element under the data root and has a collection of child elements of a single frame. This is assigned to the frames of a free frame form.

The following displays an example XML file and DTD file. In the examples, <Frame> tag is specified as the frame group item.

Example XML file:

```
<DataRoot>
  <Body>
    <Frame>
      <Detail>
        <Data1>Data</Data1>
        <Data2>
          <Data21>Data</Data21>
          <Data22>Data</Data22>
        </Data2>
      </Detail>
    </Frame>
  </Body>
</DataRoot>
```

Example DTD file:

```
<!ELEMENT DataRoot (Body)>
<!ELEMENT Body (Frame*)> ← *1
<!ELEMENT Frame (Detail*)> ← *2
<!ELEMENT Detail (Data1,Data2)>
<!ELEMENT Data1 (#PCDATA)>
<!ELEMENT Data2 (Data21, Data22)>
<!ELEMENT Data21 (#PCDATA)>
<!ELEMENT Data22 (#PCDATA)>
```

Elements that satisfy the following conditions can be specified as the frame group item.

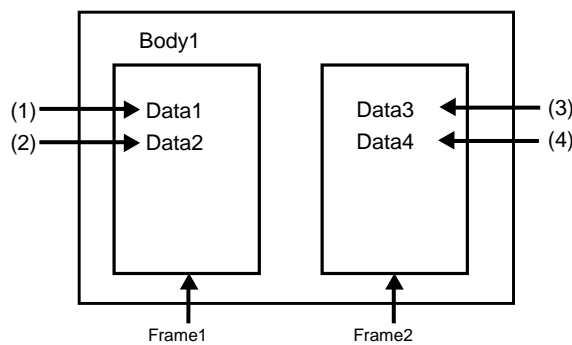
- Common (when designing the form/outputting the form)
  - Conditions under which XML files or DTD files can be used in List Creator are met  
=> ["D.2 Conditions under which XML Files and DTD Files can be Used in List Creator"](#)
  - It contains child elements (see \*2 in the above example DTD file)
  - It contains child elements that are details group items
- When designing forms
  - When referencing DTD files, it contains child elements that can occur multiple times (has a "+" or "\*") (see \*1 in the above example DTD file)

• When outputting forms

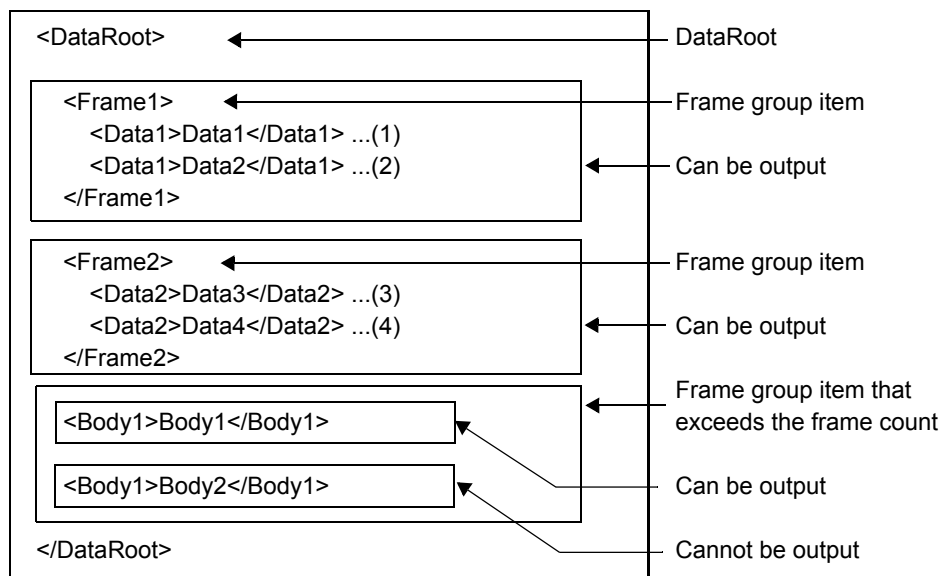
- There are only elements, which contain child elements for a single frame (i.e. elements that are frame group items), for the number of frames defined in the form  
 If the number of elements, which contain child elements for a single frame (i.e. elements that are frame group items), exceeds the number of frames defined in the form, only the first data that exceeds the frame count will be output.

The following is an example where the number of elements, which contain child elements for a single frame (i.e. elements that are frame group items), exceeds the number of frames defined in the form.  
 In this example, two frames are defined in the form. Also, there are three tags (<Frame1> tag, <Frame2> tag, and <Body1> tag) that are elements containing child elements for a single frame (i.e. frame group items). All the data under the <Frame1> tag and the <Frame2> tag will be output, however, only the first data in the <Body1> tag will be output.

Form example:



Example XML file:

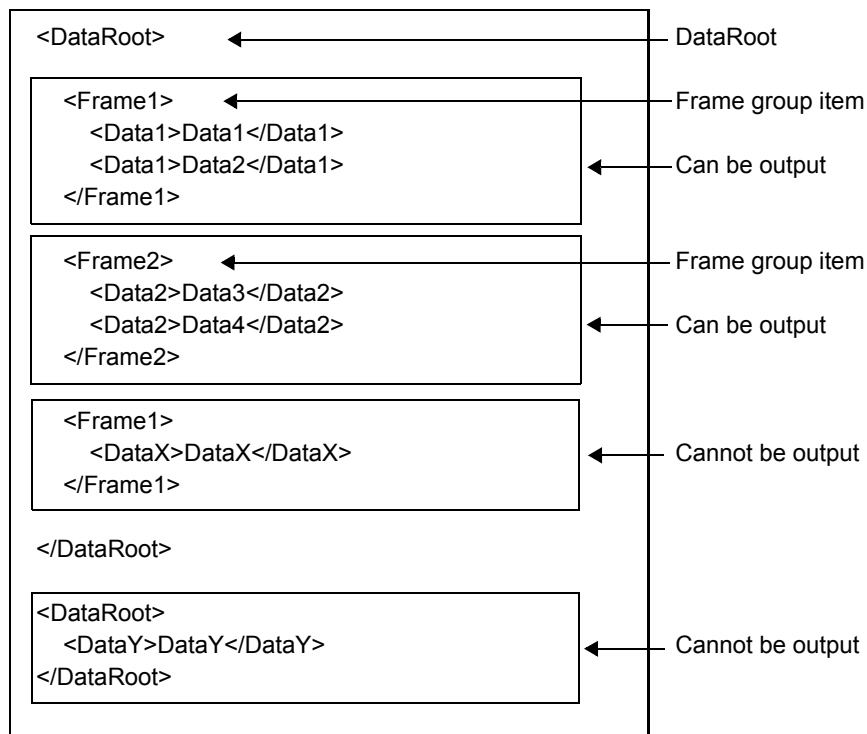


- In the same hierarchy of any element above the element that contains the child elements of a single frame (i.e. an element that is a frame group item), there are no elements with the same name but with different data structures under them

The following is an example where in the same hierarchy of any element above the element that contains the child elements of a single frame (i.e. an element that is a frame group item), there are elements with the same name but with different data structures under them.

In this example, there are two tags (<DataRoot> tag and <Frame1> tag) with different data structures under them in the same hierarchy. All the data of the <Data1> tag under the first <DataRoot> tag and <Frame1> tag will be output, however, the data of the <DataX> tag under the second <Frame1> tag and the <DataY> tag under the second <DataRoot> tag will not be output.

For example:



- One frame can be output for one frame group item. Therefore, if the elements specified for a frame group item have a hierarchical structure, specify the element in the lowest level of the hierarchy as the frame group item.

## D.8 Data Root and Page Feed

When a form is output, the relationship between the data root and the page feed varies for each form format. The relationship between the data root and the page feed is explained for each form format.

For details on the page feed when outputting forms, refer to the following:

=> ["8.5 New Page/New Frame of Form"](#)

- **For single page forms**  
A new page starts for each data root.
- **For simple table, multipart table, block repeat, or sticky label forms**  
A new page does not start for the end tag when there are multiple data roots.
- **For free frame forms**  
A new page does not start for the end tag when there are multiple data roots.

## D.9 Points

Note the following points when performing form outputs for "XML Format" input data:

### D.9.1 When Designing Forms

When using "XML Format" input data, design the form with attention to the following points.

- **Form format (common)**

Consider the following points when designing forms.

- For the data length, calculate the total number of bytes as three bytes per Japanese character (including half-width katakana) and two bytes per half-width alphanumerical character. If the data length to be specified is not sure because Japanese and half-width alphanumerical characters are mixed, multiply the number of characters by three and specify the result as the data length.  
The string may get clipped during the output if the data length is too short.
- Linefeed code or new page code control cannot be specified when designing forms.
- XBRL is a kind of XML, however, since XBRL is a special format of XML using technology called XLink, it cannot be used as XML file in List Creator.
- Output image cannot be checked by clicking [Preview] button on the standard toolbar in the [Form Style Descriptor] window.

- **For single page format**

Repeats cannot be specified for fields (with the exception of static fields and block static fields).

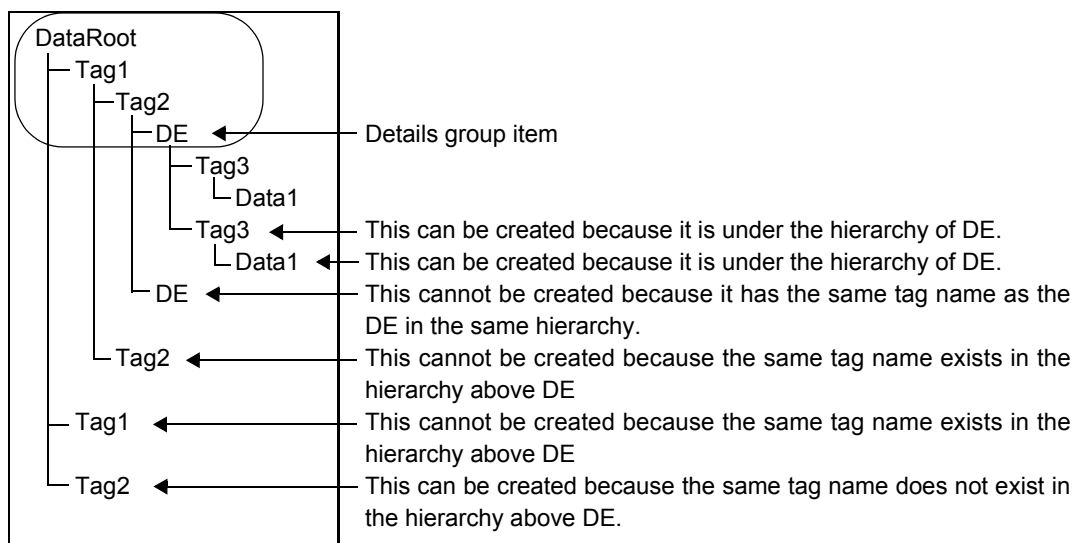
- **For simple table, block repeat, or sticky label format**

Do not define a field that has the same tag name as a tag of a body area group item within the same hierarchy. The form layout data cannot be saved if such a field is defined.

- **For multipart table format**

Do not define a field with the same tag name under the hierarchy from the data root to the details group item. The form layout data cannot be saved if such a field is defined.

The following displays examples of fields that may be created and that may not be created. In the following example, DE is the details group item.



The following are the points to note regarding data lists.

- Details group items cannot be deleted.
- If there are multiple DE (details) partitions, all the fields defined to a DE (details) partition become configuration items of the details group item.

- When a field is inserted, pasted, or copied to an edit area, the field is inserted to the data list in the following manner:
  - When defining to a DE (details) partition
    - The field is inserted to the end of the configuration items of the details group item.
  - When defining to somewhere other than DE (details) partition
    - The field is inserted to the end of the data list.
- When defining hidden fields (records) as visible fields, the following apply:
  - For configuration items of a details group item
    - The field cannot be defined to a partition other than DE (details) partition.
  - For non-configuration items of a details group item
    - The field cannot be defined to a DE (details) partition.
- When moving fields in the data list, the following apply:
  - Configuration items (with the exception of hidden fields (records)) of a details group item cannot be moved outside of the details group item.
  - Fields other than configuration items (with the exception of hidden fields (records)) of a details group item cannot be moved to a details group item.
  - If a details group item only contains one field, it may not be moved outside of the details group item even if it is a hidden field (record).

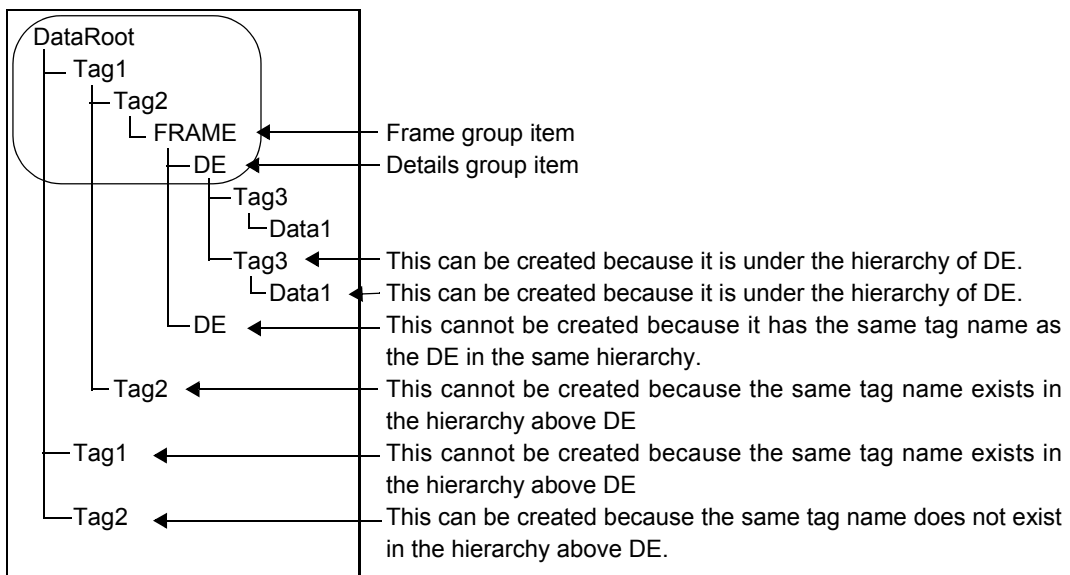
The following are the points to note regarding the edit area.

- With the exception of static fields, block static fields, date fields, time fields, and barcode fields (U.S. Postal FIM), fields that are defined to a DE (Details) partition cannot be moved to a partition of another type.
- With the exception of static fields, block static fields, date fields, time fields, and barcode fields (U.S. Postal FIM), fields that are defined to a partition other than DE (Details) partition cannot be moved to a DE (Details) partition.
- A DE (Details) partition cannot share record areas with other partitions.
- If visible fields no longer exist in a DE (Details) partition, the following operation cannot be performed.
  - Changing Visible Field to a Hidden Field
  - Delete
  - Cutting

• For free frame format

Do not define a field with the same tag name under the hierarchy from the data root to the details group item. The form layout data cannot be saved if such a field is defined.

The following displays examples of fields that may be created and that may not be created. In the following example, DE is the details group item.



The following are the points to note regarding data lists.

- Details group items cannot be deleted.
- Frame group items cannot be deleted.
- If there are multiple DE (Details) partitions within a frame, all the fields defined to a DE (Details) partition become configuration items of the details group item corresponding to that frame.
- When a field is inserted, pasted, or copied to an edit area, the field is inserted to the data list in the following manner:
  - When defining to a CH (Control Header) partition or CF (Control Footer) partition of a frame  
The field is inserted to the end of the configuration items of the frame group item.
  - When defining to a DE (Details) partition of a frame  
The field is inserted to the end of the configuration items of the details group item corresponding to the frame.
  - When defining to somewhere other than frame partition  
The field is inserted to the end of the data list.
- When defining hidden fields (records) as visible fields, the following apply:
  - For configuration items of a frame group item  
The field cannot be defined to a partition other than CH (Control Header) partition or CF (Control Footer) partition of a corresponding frame.
  - For configuration items of a details group item of a frame  
The field cannot be defined to a partition other than DE (Details) partition of a corresponding frame.
  - For non-configuration items of a details group item or frame group item  
The field cannot be defined to a frame partition.
- When moving fields in the data list, the following apply:
  - Configuration items (with the exception of hidden fields (records)) of a frame group item cannot be moved outside of the frame group item.
  - Fields other than configuration items (with the exception of hidden fields (records)) of a frame group item cannot be moved to a frame group item.
  - Configuration items (with the exception of hidden fields (records)) of a details group item of a frame cannot be moved outside of the details group item.
  - Fields other than configuration items (with the exception of hidden fields (records)) of a details group item of a frame cannot be moved to a details group item.
  - If a details group item of a frame only contains one field, it may not be moved outside of the details group item even if it is a hidden field (record).
  - Details group items of a frame cannot be moved outside of the frame group item.

The following are the points to note regarding the edit area.

- With the exception of static fields, block static fields, date fields, time fields, and barcode fields (U.S. Postal FIM), fields that are defined to a frame partition (with the exception of DE (Details) partitions) cannot be moved to a DE (Details) partition of the same frame, a frame partition of another frame or outside of the frame.
- With the exception of static fields, block static fields, date fields, time fields, and barcode fields (U.S. Postal FIM), fields that are defined to a DE (Details) partition of a frame cannot be moved to a partition other than the DE (Details) partition of the same frame.
- With the exception of static fields, block static fields, date fields, time fields, and barcode fields (U.S. Postal FIM), fields that are defined outside of a frame cannot be moved to the frame partition.
- A frame partition cannot share record area with a frame partition of a different frame.
- A frame partition cannot share record area with one outside the frame.
- A DE (Details) partition cannot share record area with other frame partitions within the same frame.
- If visible fields no longer exist in a frame, the following operation cannot be performed.
  - Changing Visible Field to a Hidden Field
  - Delete
  - Cutting

## D.9.2 When Designing Input Data

When using "XML Format" input data, design the input data with attention to the following points.

- When outputting forms of XML data referring external subset DTD file, specify the following URL in the document type declaration (DOCTYPE declaration) of XML data.
  - When outputting forms by using remote form output and connector linkage  
Place the DTP file on a public location which other computers can access, and specify the full path to the location's URI in XML data.

In the following cases, specify the full path to the DTD file URI in the XML data. When the URI is specified with a relative path, forms are not output.

- When outputting forms by using FormsStream class of Java interface
- When outputting forms by using prprintx command
- When outputting forms by using incoming data monitoring function

### Character code

- The character code of XML data is converted before output depending on the environment. Therefore, a character in the XML data is not output if the character does not belong to the character code which is used for output.
  - When the List Creator print server runs on Windows  
XML data is output after being converted from its native character code to UTF8 format.
  - When the List Creator print server runs on Solaris  
In EUC and Shift-JIS environment, XML data is output after being converted to U90, S90 or Shift-JIS according to the environment variable LANG and OA\_EUCTYPE.
  - When the List Creator print server runs on Linux  
In EUC environment, XML data is output after being converted to U90 or S90 according to the environment variable LANG and OA\_EUCTYPE.

### Output of User Defined Characters

- When the List Creator print server runs on Windows
  - When the character code of input data is Shift-JIS, register the user defined characters in UNICODE (for printing, saving as PDF file, or outputting to Excel file).
  - For making e-Record, register the user defined characters in UNICODE.
- When the List Creator print server runs on UNIX OS  
The available characters vary depending on the environment where the form is output.



# Appendix E

## Combination-Form Descriptor File

---

This appendix explains the Combination-Form Descriptor file used in Combination-Form output.

Combination-Form Descriptor file is a file in which the information regarding combination of forms is set. Combination-Form Descriptor file can be created by performing the following operations.

- Select [Update Combination-Form Descriptor] in the [Combination-Form Descriptor List].  
A Combination-Form Descriptor file is created in the form storage directory with the file name "Lccbform.ini".  
The character code is Shift-JIS.
- Select [Export Combination-Form Descriptor] in the [Combination-Form Descriptor List].  
A Combination-Form Descriptor file is created in the specified directory with the specified file name.  
The character code is Shift-JIS.



- When using the export function, "Lccbform.ini" cannot be specified for the Combination-Form Descriptor file name.

## E.1 Format of Combination-Form Descriptor File

The description format of Combination-Form Descriptor file is as follows:

[Section Name]  
 Setting value, or Keyword=Setting value  
 .....

- [Section Name] : Indicates a category of Combination-Form Descriptor.
- Keyword : Indicates a keyword of Combination-Form Descriptor.
- Setting value : Indicates a value to be set for the Combination-Form Descriptor or the keyword.

The contents should be specified according to the following rules.

- To place the Combination-Form Descriptor file in an environment where the form is output, the following character code must be used.
  - For Windows  
Use the Shift-JIS character code to create the file.
  - For UNIX OS  
Use the Shift-JIS, EUC, or UNICODE (UTF8) character code to create the file.
- It should be described using the character code appropriate for the location of the Combination-Form Descriptor File, or the character code to be used at the time of running user application.
- For details of character codes, refer to the descriptions about preparing and deploying form resources in the Online Manual "Setup & Operation".
- A section ends at the next section name or at the end of the file.
- A half-width semicolon (;) at the start of a line indicates a comment line.
- Any string other than the section names and keywords is ignored.
- Section names and keywords are not case sensitive.
- When the same section name or keyword is specified more than once, or the section name or keyword is omitted, the action varies according to section name or keyword. For details, refer to the descriptions of section name and keyword.
- The string from the character after the "=" up to the new line code is treated as a setting value.
- Number or types of characters which can be specified in the setting value varies according to keyword. For details, refer to the description of keyword.

The following is a list of sections and keywords.

- Section

Section names that can be set are shown below.

| Section Name            | Description   |
|-------------------------|---|
| [GROUP]                 | This is the section to enter a list of Combination-Forms. This specification cannot be omitted. |
| [Combination-Form Name] | Each Combination-Form name is set as a section name. This specification cannot be omitted.      |

• Keyword and Setting Value

Keywords and setting values that can be set in each section are shown below.

[GROUP] Section

| Setting Value                | Description  |
|------------------------------|--|
| <i>Combination-Form Name</i> | Specify the Combination-Form name within 64 bytes. This specification cannot be omitted. |

[Combination-Form Name] Section

| Keyword Name | Outline  |
|--------------|--|
| Forms        | Specify the form name used in the Combination-Form by placing a comma (,) in between. This specification cannot be omitted.  |
| Comment      | Specify comments regarding the Combination-Form within 64 bytes. Comment can be omitted.<br>The following characters cannot be used:<br>- "Semicolon (;)", "Double quotation mark (")", "Linefeed" |

The following is an example of Combination-Form Descriptor file.

```

Combination-Form name "Combination1" : Combination of Form "HeaderA" and Form "DetailX"
                                     Comment "Original Form"
Combination-Form name "Combination2" : Combination of Form "HeaderB" and Form "DetailY"
                                     Comment "Customer Copy"
Combination-Form name "Combination3" : Combination of Form "HeaderC" and Form "DetailZ"
                                     Comment "Shop Copy"
    
```

Description Example of Combination-Form Descriptor File

|   |
|---|
| <pre> [GROUP] Combination1 Combination2 Combination3 [Combination1] Forms=HeaderA, DetailX Comment=Original Form [Combination2] Forms=HeaderB, DetailY Comment=Customer Copy [Combination3] Forms=HeaderC, DetailZ Comment=Shop Copy     </pre> |
|---|

# **Appendix F**

## *COPY Clause File Creation Function*

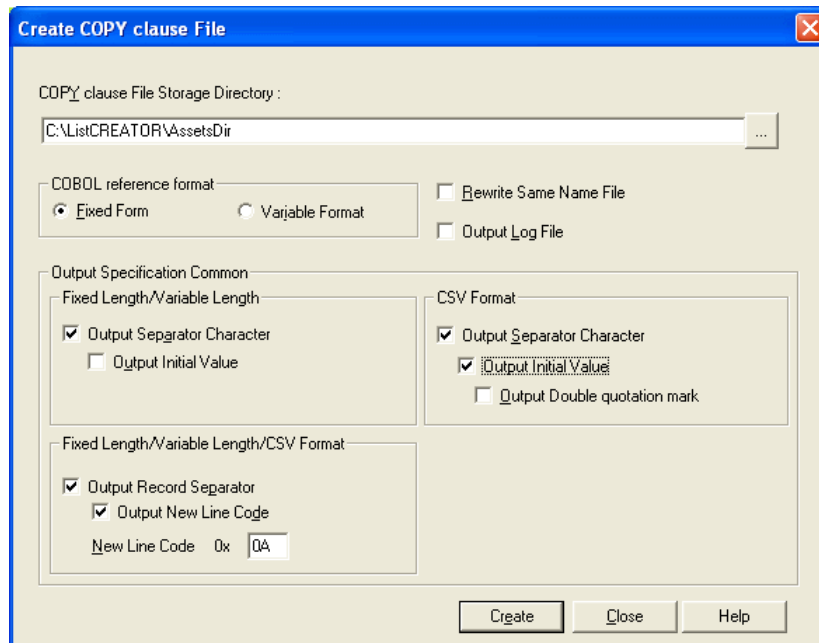
---

This appendix explains the COPY clause file creation function.

## F.1 COPY Clause File Creation Function

COPY clause file creation function converts the information of input data defined in the form into COBOL record format file. This function is valid when creating input data in the COBOL application.

To create a COPY clause file, select the target form in the [Form Descriptor List] window and then click [File] - [Create COPY clause File], or right-click the selected form to open the pop-up menu and then click [Create COPY clause File].



**Figure F.1** [Create COPY clause File] window

For details of the COPY clause file creation, refer to the Designer Help guide for List Creator Designer.

# Index

## Symbols

|  |     |
|--|-----|
| [Convert Template Image To Overlay] Window ..... | 83  |
| [Data Definitions] Window .....                  | 109 |
| [Form Contents Report] Window .....              | 205 |
| [Form Descriptor List] Window .....              | 47  |
| [Form Style Descriptor] Window .....             | 48  |

## A

|   |         |
|---|---------|
| Adding the Overlay to List Creator .....                      | 235     |
| Area .....  | 19      |
| Areas for Each Form Format .....                              | 19      |
| Association Data .....  | 18, 102 |
| Association Data for Form Layout Data and<br>Input Data ..... | 102     |
| Association data for form layout data and input data ..       | 18      |
| Automatic Calculation of Output Data .....                    | 102     |
| Automatic Creation of Items .....                             | 66      |

## B

|   |          |
|---|----------|
| Barcode .....                           | 132      |
| Barcode field .....                     | 21       |
| Barcode Field (U.S. Postal FIM) .....   | 22       |
| Basic Knowledge of Form Designing ..... | 15       |
| BASIC1 .....                            | 72       |
| bip .....                               | 41       |
| Block .....                             | 19       |
| Block Area .....                        | 34       |
| Block Repeat Format .....               | 62       |
| Block repeat format .....               | 33       |
| Block Static field .....                | 23       |
| Block Text .....                        | 132      |
| Block Text Field .....                  | 21       |
| Body (Block Repeat Format) .....        | 34       |
| Body (BODY) .....                       | 20       |
| Body (Free Frame Format) .....          | 25       |
| Body (Simple Table Format) .....        | 30       |
| Body Area .....                         | 19       |
| Body Group Item .....                   | 262      |
| BREAK .....                             | 105, 107 |
| BREAK Field .....                       | 103      |
| Byte Size .....                         | 141      |

## C

|  |          |
|--|----------|
| CASE Data Output .....   | 108      |
| Category Name .....  | 51       |
| Cautions .....   | 164      |
| Cautions (when Designing a Combination-Form) ....                        | 164      |
| Cautions (when Designing Forms to be Converted to<br>PDF) .....          | 251      |
| Cautions (when Designing Forms to be Faxed) .....                        | 251      |
| Cautions (when Designing Forms to be saved as e-Re-<br>cord) .....       | 251      |
| Cautions for Designing a Combination-Form .....                          | 164      |
| Cautions for Designing Forms .....                                       | 246      |
| Cautions for Each Output Method .....                                    | 251      |
| Cautions regarding Custom-Spacing Grid .....                             | 247      |
| Cautions regarding Environment for Designing<br>Forms .....              | 254      |
| Cautions regarding Input Data .....                                      | 256      |
| Cautions when Changing the Grid Type .....                               | 247      |
| CF (Control Footer) .....  | 26, 32   |
| CH (Control Header) .....  | 26, 31   |
| Change Data List Structure .....   | 98       |
| Change Field Info .....  | 87       |
| Changing Form Output Methods .....                                       | 210      |
| Changing Form Style Information .....                                    | 208      |
| Changing the Form Output Methods .....                                   | 210      |
| Changing the Form Style Information .....                                | 208      |
| Character Spacing .....  | 57       |
| Check Box .....  | 21, 132  |
| Checking Fields and Contents to which Operators are<br>Specified .....   | 104      |
| Checking Input Data Reports .....  | 122      |
| Checking the Output Image of Combination-Form ..                         | 162      |
| Client ID .....  | 181      |
| Client Secret .....  | 181      |
| Color .....  | 230      |
| Color that can be Recognized .....                                       | 242      |
| Combination-Form Defining Flow .....                                     | 159      |
| Combination-Form Descriptor .....  | 160      |
| Compensate for Image Skew .....  | 83       |
| Completion of Overlay Creation .....                                     | 234      |
| Components of Form Layout Data .....                                     | 19       |
| Condition Judgment .....   | 105, 108 |
| Conditions under which Elements can be Specified as a<br>Data Root ..... | 275      |

|  |          |   |          |
|--|----------|---|----------|
| Conditions under which Elements Can be Specified as<br>Body Area Group Items ..... | 276      | Definition of Association Data .....                                  | 101, 109 |
| Conditions under which Elements can be Specified as<br>Details Group Items .....   | 278      | Definition of BREAK Control .....                                     | 103      |
| Conditions under which Elements can be Specified as<br>Frame Group Items .....     | 281      | Definition of Calculations .....                                      | 102      |
| Confirming the Output Image .....  | 111      | Definition of Schema File (schema.ini) .....                          | 73       |
| Confirming the Output Image of Form Layout Data                                    | 111      | Delete .....  | 199      |
| Control Level .....  | 141      | Description Format of Field Info Files .....                          | 71       |
| Convert Hidden Field to Visible Field Window .....                                 | 94       | Designing Input Data .....  | 121, 124 |
| Convert Template Image to Overlays Function ....                                   | 66, 82   | Designing Input Data (a Data File with the Field Name<br>Label) ..... | 148      |
| Convert the Template Image to an Overlay Pattern ....                              | 83       | Designing Input Data based on Input Data Reports ..                   | 124      |
| Converting Template Image to Overlays .....  | 82       | Designing Input Data for Combination-Form .....                       | 167      |
| Copy .....   | 199      | Designing Input Data for Multi-form Output .....                      | 142      |
| COPY Clause File Creation Function .....   | 293      | Designing Input Data for Multi-form Output .....                      | 142      |
| CPI .....  | 57       | Designing the Form Output Methods .....                               | 156      |
| Created Files .....  | 41       | Designing the Input Data .....  | 121      |
| Creating an Image File .....   | 69       | Details Group Item .....  | 262      |
| Creating Field Info Files .....  | 69       | Determining the Form Format .....                                     | 62       |
| Creating Form Layout Data .....  | 75       | Determining the Methods for Creating Form<br>Layout Data .....        | 66       |
| Creating forms Based on Field Information .....                                    | 86       | Determining the Type of Input Data .....                              | 104      |
| Creating the Combination-Form Descriptor .....                                     | 160      | Display the Combination-Form Descriptor List .....                    | 160      |
| Creation of Field Information from a Schema File (sche-<br>ma.ini) .....           | 73       | Displaying [Form Style Descriptor] Window .....                       | 48       |
| Criteria for Selecting a Form Format .....   | 62       | Displaying Template Image .....                                       | 83       |
| CSV format .....   | 38       | Displaying Template Image in Color .....                              | 83       |
| Currency .....   | 133, 174 | Dot .....   | 56       |
| Customizing Operating Environment .....  | 56       | dse .....   | 41       |
| Custom-Spacing Grid .....  | 59       | DTD File .....  | 68, 90   |
|  |          | DTD File Conditions .....   | 263      |
|  |          | DTD File that can be Used .....                                       | 263      |

**D**

|   |              |
|---|--------------|
| Data Definitions .....                  | 18, 126      |
| Data Definitions (Report Header) .....  | 126          |
| Data File .....                         | 36           |
| Data Length .....                       | 72, 140      |
| Data List .....                         | 93           |
| Data Root .....                         | 262          |
| Data Root and Page Feed .....           | 284          |
| Data with the Field Name Label .....    | 36           |
| DataRoot .....                          | 38           |
| Date .....                              | 133, 174     |
| Date Field .....                        | 23           |
| DE (Details) .....                      | 26, 31       |
| Decimal places .....                    | 72, 140, 141 |
| Default Value Setting .....             | 57           |
| Default Value Setting for Fields .....  | 79           |
| Default Values and Unit of a Grid ..... | 247          |
| Defining Association Data .....         | 101, 109     |
| Defining the Form Output Methods .....  | 156          |

**E**

|  |        |
|--|--------|
| Edit Area .....  | 48, 52 |
| Editing Format Data File .....   | 115    |
| Editing the Form Style Descriptor (when Setting a Com-<br>bination-Form) ..... | 163    |
| Editing Toolbar .....  | 48     |
| Editor Specification Window .....  | 115    |
| Electronic Data .....  | 66     |
| Example of Printed Input Data Report .....                                     | 123    |
| Excel File (Input Data Reports) .....  | 122    |
| Extended Definitions .....   | 18     |
| Extracting to Data List .....  | 95     |

**F**

|                            |     |
|----------------------------|-----|
| Field .....                | 21  |
| Field Detail .....         | 133 |
| Field Details Report ..... | 205 |
| Field Entry List .....     | 51  |





|   |        |
|---|--------|
| Layout Preview Window .....               | 55     |
| Layout Print .....                        | 112    |
| Line Recognition .....                    | 240    |
| Line Spacing .....                        | 57     |
| Linking with a Database .....             | 39     |
| Linking with ERP .....                    | 40     |
| Linking with XBRL Data .....              | 40     |
| List Pane .....                           | 48, 49 |
| Listing Form Descriptor Information ..... | 198    |
| Literal .....                             | 174    |
| LPI .....                                 | 57     |

**M**

|   |          |
|---|----------|
| Mail .....                                    | 133      |
| Maintaining .....                             | 207      |
| Maintaining Form Descriptor Information ..... | 207      |
| Maintaining Forms .....                       | 207      |
| Management .....                              | 196      |
| Managing Form Descriptor Information .....    | 196      |
| Maximum Value (Custom-Spacing Grid) .....     | 247      |
| Media Data .....                              | 105, 108 |
| Media Data Type .....                         | 108      |
| Menubar .....                                 | 48       |
| Microsoft Translator .....                    | 181      |
| Minimum Value (Custom-Spacing Grid) .....     | 247      |
| mm .....                                      | 56       |
| Multipart Table Format .....                  | 62       |
| Multipart table format .....                  | 31       |

**N**

|  |               |
|--|---------------|
| Navigator Links Function .....                             | 39            |
| New .....  | 76, 199       |
| New Frame .....  | 153           |
| New Frame of the Form .....                                | 153           |
| New Page .....   | 153           |
| New Page of the Form .....                                 | 153           |
| New Window .....   | 76            |
| Not Use Overlay Conversion .....                           | 83            |
| Notes (Designing Input Data for<br>Combination-Form) ..... | 170           |
| Number .....   | 126           |
| Number of Repeats .....                                    | 126           |
| Numeric .....  | 132, 133, 174 |
| Numeric Field .....  | 21            |

**O**

|   |         |
|---|---------|
| Object .....  | 21      |
| OCR-B .....   | 132     |
| OCR-B Field .....   | 21      |
| Open .....  | 199     |
| Operating Environment for Form Designing .....              | 55      |
| Operation when Changing the Input Data Format ...           | 257     |
| Oracle E-Business Suite integration .....                   | 40      |
| Output Example of Test Print .....                          | 120     |
| Output Field .....  | 262     |
| Output Method .....   | 156     |
| Output Methods and Tab Settings .....                       | 156     |
| Output Settings Window .....                                | 232     |
| Output the List of Forms .....                              | 200     |
| Output the List of Forms and Existence of change ...        | 200     |
| OutputID .....  | 23      |
| Outputting the List of Form Descriptor<br>Information ..... | 200     |
| ovd .....   | 41      |
| Overlay .....   | 66, 225 |
| Overlay graphic .....                                       | 24      |
| Overlay Image .....   | 24      |
| Overlay text .....  | 24      |
| Overlay Toolbar .....                                       | 49      |
| Overlays .....  | 24      |
| OWF File Creation .....                                     | 156     |

**P**

|   |            |
|---|------------|
| Page Feed Timing (during Combination-Form<br>Output) .....    | 171        |
| Paper Size .....  | 230        |
| Partition .....   | 19, 20, 25 |
| Partition Details Report .....                                | 205        |
| Partition Graphic .....                                       | 24         |
| Partition Graphics Toolbar .....                              | 49         |
| Partition List .....  | 49         |
| Partition list .....  | 80         |
| Partition Rectangle .....                                     | 24         |
| Parts .....   | 21         |
| PDF File (Input Data Reports) .....                           | 122        |
| Per Inch .....  | 57         |
| PF (Page Footer) .....  | 32         |
| PH (Page Header) .....  | 31         |
| Pitch .....   | 57         |
| pmd .....   | 41         |
| Points for Designing Input Data for<br>Combination-Form ..... | 170        |
| Points to Note .....  | 239, 240   |
| Points to Note during Printing Forms .....                    | 239        |

|  |            |   |             |
|--|------------|---|-------------|
| Port Definition Window .....                         | 224        | Relationship Between Order of Fields in Form Layout |             |
| Post Code .....                                      | 132        | Data and Order of Input Data .....                  | 102         |
| Post Code Field .....                                | 23         | Rename .....  | 199         |
| Preparation .....                                    | 61         | Replace Information .....                           | 193         |
| Preparing DTD File .....                             | 74         | Replacement String .....                            | 117         |
| Preparing for Form Designing .....                   | 61         | Replacing Properties .....                          | 173         |
| Preparing the Files .....                            | 69         | Resolution .....                                    | 230         |
| Preparing XML File .....                             | 74         | Return Field to Undefined State .....               | 89          |
| Preview .....  | 113        | RH (Report Header) .....                            | 31          |
| Preview (Combination-Form) .....                     | 162        | Rulers .....  | 48, 52      |
| Preview window .....                                 | 117        |   |             |
| Previewing the Combination-Form .....                | 162        | <b>S</b>  |             |
| Previewing when Designing a Form .....               | 113        | SAP R/3 RDI integration .....                       | 40          |
| Previewing with Utility (Preview) .....              | 113        | Saving the Combination-Form Descriptor .....        | 163         |
| Prfmwiz.ini File .....                               | 51, 77, 78 | Schema File (schema.ini) .....                      | 67          |
| Print (Layout Print) .....                           | 112        | Select Editing Objects Tabs .....                   | 48, 53      |
| Print (Test Print) .....                             | 120        | Selecting Information to be Defined for Association |             |
| Print Area .....                                     | 48         | Data .....  | 104         |
| Print with Scales .....                              | 112        | Selection of Field Type .....                       | 70          |
| Printer Definition .....                             | 220        | Selection of Grid Types .....                       | 56          |
| Printing (Input Data Reports) .....                  | 122        | Separator .....                                     | 133         |
| Printing Reports of Form Style Information .....     | 205        | Separator Character .....                           | 37, 38, 126 |
| Process Flow .....                                   | 228        | Setting and Viewing Form Output Methods .....       | 199         |
| Properties that can be Replaced .....                | 174        | Setting Frame Links .....                           | 25          |
| Property .....                                       | 174        | Settings of Association Data .....                  | 109         |
| Property Identifier .....                            | 184        | Setup .....   | 220, 224    |
| Property List .....                                  | 48, 52     | Side View .....                                     | 48, 49, 52  |
| Property Replace File .....                          | 184        | Simple Table Format .....                           | 62          |
| psf .....  | 41         | Simple table format .....                           | 30          |
|  |            | Single Page Format .....                            | 29, 62      |
| <b>Q</b>   |            | Size to be Recognized .....                         | 239         |
| Quick Handle .....                                   | 48, 52     | Snap to Grid .....                                  | 59          |
|  |            | Sorting the Data .....                              | 141         |
| <b>R</b>   |            | Specifying the Form Format (for XML Data) .....     | 267         |
| Radio Button .....                                   | 21, 132    | Specifying the Form Format Responding to the XML    |             |
| Range of Input Data (per Page, per Detail) .....     | 266        | Data Configuration .....                            | 267         |
| Range of One Page or One Detail of the Input Data in |            | Specifying the Form Storage Directory .....         | 198         |
| XML Format .....                                     | 266        | SSIS link function .....                            | 39          |
| Range of One Page or One Detail of the Input Data in |            | Standard .....                                      | 133         |
| XML Format (Free Frame Format) .....                 | 266        | Standard Toolbar .....                              | 48          |
| Range of One Page/One Detail .....                   | 266        | Starting the [Form Descriptor List] .....           | 47          |
| Recognition of lines .....                           | 242        | Static Field .....                                  | 23          |
| Recognize Colors .....                               | 84         | Status Bar .....                                    | 48, 54      |
| Recognizing the Color Form .....                     | 242        | Sticky Label .....                                  | 34          |
| Recommended Properties for Different Languages ..    | 193        | Sticky Label Format .....                           | 62          |
| Record Information .....                             | 18         | Sticky label format .....                           | 34          |
| Record List .....                                    | 50         | String Operation .....                              | 105, 108    |
| Record list .....                                    | 79, 88     |   |             |

**T**

|  |          |
|--|----------|
| Tail (Block Repeat Format) .....         | 34       |
| Tail (Simple Table Format) .....         | 30       |
| Tail Area .....                          | 19       |
| Target page for overlay conversion ..... | 239      |
| Template .....                           | 66       |
| Template image file .....                | 239      |
| Test Print .....                         | 120      |
| Text .....                               | 132, 133 |
| Text Field .....                         | 21       |
| The Overlay Utility .....                | 212      |
| Time Field .....                         | 23       |
| Toolbar .....                            | 48       |
| Tools Toolbar .....                      | 48       |
| Type .....                               | 133      |
| Types of operators .....                 | 105      |

**U**

|  |     |
|--|-----|
| Understanding Data Configuration .....             | 124 |
| Understanding Data Format .....                    | 132 |
| Understanding Data Length of Each Field .....      | 140 |
| Understanding the Data Format for Each Field ..... | 132 |
| Uninstall .....                                    | 219 |
| Unit Settings .....                                | 56  |
| Use Field Info File .....                          | 86  |
| Use Navigator Links Function .....                 | 68  |
| Use Schema File (schema.ini) .....                 | 86  |
| Use XBRL Data Support Function .....               | 68  |
| Using DTD File .....                               | 89  |
| Using input data field information .....           | 67  |
| Using Item Information .....                       | 67  |
| Using Layout Data .....                            | 66  |
| Using Layout Data from a existing Form .....       | 66  |
| Using Layout Data from a Paper Form .....          | 66  |
| Using Layout Data from a Preprinted Paper .....    | 66  |
| Using Schema File (schema.ini) .....               | 73  |
| Using XML File .....                               | 89  |

**V**

|                                |     |
|--------------------------------|-----|
| Value .....                    | 133 |
| Value for selected state ..... | 72  |
| Variable length .....          | 37  |
| View Toolbar .....             | 48  |

**W**

|   |     |
|---|-----|
| What is Form Designing .....                                    | 16  |
| when Designing Forms for Multi-Form Output,<br>Cautions .....   | 252 |
| when Designing Forms to be Faxed, Cautions .....                | 251 |
| when Designing Forms to be saved as e-Record,<br>Cautions ..... | 251 |
| when Designing Forms to Output to Excel File,<br>Cautions ..... | 251 |
| when Designing Forms to Save as PDF, Cautions ...               | 251 |
| Windows Used for Form Designing .....                           | 47  |

**X**

|                                 |             |
|---------------------------------|-------------|
| XML File .....                  | 68, 91, 261 |
| XML File Conditions .....       | 263         |
| XML File that can be Used ..... | 263         |
| XML format .....                | 38          |
| XML Root .....                  | 262         |
| XMLRoot .....                   | 38          |