
Amgraf's

**OneForm Designer Plus
XML Workflow –
XML Import/Export Options**

*Reference Manual
Version 4.7.2.5*

OneForm™ Designer Plus

© 2009 by Amgraf, Inc. All rights reserved.

These specifications are subject to change without notice.

OneForm and MECCA are trademarks of Amgraf, Inc.

All other brand and product names are or may be trademarks used to identify products or services of their respective owners.



1501 Oak Street • Kansas City, MO 64108-1424

Phone: (816) 474-4797 • Fax: (816) 842-4477

E-Mail: info@amgraf.com • Internet: www.amgraf.com

TABLE OF CONTENTS

Chapter 1: Overview of the XML Workflow

System Requirements for the XML Options	2
Use of JAR Files	3
The XML Document Type Definitions	3
GUID's Identify Each Object/Component	3
Glossary of Terms Used	4
Sample A - Automated Composition	5
Form Title/Description Elementse (DTD Tags)	6
Question Group Elements (DTD Tags)	7
Question Group One	8
Question Group Two	9
Question Group Three	10
Question Group Four	11
Question Group Five	12
Question Group Six	13

Chapter 2: Using XML Interactive Import

XML Content Open/Processing Options	15
Open Options	16
Processing Options	16
Selecting a Stylesheet	17
XML Import Compose Options	17
Using the Import Options	18
Using Arrange Responses	19
Arrange Responses Example	20
Import Remainder	21
Saving the Graphic File	22
XML Save (Export) Option	22
Saving as a PDF	23

Chapter 3: Creating Base Drawing Templates

Steps for Creating a New Template	26
Moving an Existing Component to Absolute 0/0	27
Changing a Page Boundary to 0/0 Coordinate	27
Getting Coordinates for Stylesheets	27

Chapter 4: Interactive Stylesheet Editor

Stylesheet Components	29
Opening a Stylesheet File	30
Starting ISE	30
The Stylesheet ID Panel	31

The Columns and Blocks Panel	32
Working with Columns and Blocks	33
Modifying a Column/Block	34
Adding a New Column	35
Information about Blocks	36
Text Styles Tab	37
The Response Fields Tab	39
The Barcodes Tab	41
The Templates Tab	42
How to Make a New Stylesheet	43
[Add Columns to Base Page]	44
Modifying Columns	47
Adding a New Column	48
Change the P11 Text Style	49
Change the Comb Box Field	51
Save the New Stylesheet	52
Verifying Changes	53
Example of the XSL File	55

Chapter 5: Working with Graphic Files

Viewing GUID Components	63
No Fillable Fields View	65
Moving GUID Items	66

Chapter 6: Composition Model Option

Sample Survey Form	67
Samples of Usage for Composition Models	68
Using Composition Modeling	69
Quick Step-by-Step	70
Miscellaneous Notes	70

Chapter 7: Synchronizing Imported Objects

Opening XML Content for Edit Differences	73
Gray Boxes Identify Changes	74
Text Differences Log	75
Inserting New Questions/Instructions	76
Forcing Synchronization of XML Content	78

CHAPTER 1

Overview of the XML Workflow

The objective of the XML Workflow feature is to streamline the creation and revision of survey forms by separating the *Content* from the *Form Design*. Content refers to the instructions, question text, field captions, and other graphical objects that a user reads when filling out a form. Design refers to the page layout, column arrangements, typefaces, type sizes, form color, etc. that make the form visually appealing and easy to follow.

Diet and Exercise Magazine
P.O. Box 9863, Newport, MO 68474

2004
Customer
Survey 6-1

Please help us better serve our customers by filling out this quick and easy survey. Your time is greatly appreciated and will be rewarded with a FREE one year's subscription.

Please provide subscription information:

1. What is your name? (Print last name)

Last Name

First Name

2. What is your mailing address? (Print address below)

Address

City

State Zip Code

3. What is your telephone number? (Area Code + Number)

4. What is your sex? (Mark ONE box.)

Male
 Female

5. What is your age and date of birth? (Print numbers in boxes)

Age

Month Day Year of Birth

6. How many adults in your household are currently on diet?

Number of people:

7. I prepare most of my meals at home. (Mark ONE box.)

Yes
 No

8. My main goal is to eat more healthful.

Yes
 No

9. I am sure my progress on a scale.

Yes
 No

10. I eat more fast food than I should.

Yes
 No

11. I am good at planning meals ahead of time.

Yes
 No

12. I would like to increase energy.

Yes
 No

13. I want to lose weight for health reasons.


Yes
 No

14. Feel stressed when I am on a diet.

Yes
 No

15. I enjoy sports and other physical activities.

Yes
 No



NH-1

Figure 1-1: Questionnaire Form

Survey forms typically consist of instruction blocks followed by questions and related response fields. The question text strings (and other content elements) are often stored and maintained in a content database repository. By providing

a method to interpret XML-tagged content, OneForm Designer Plus makes it possible to establish a highly automated form production workflow, where survey authors extract XML-formatted content using database queries, and form composers flow the content into standard templated survey layouts. In addition, fillable response fields are automatically created during composition.

The style and look of the forms and composed data are controlled by XML Stylesheets and Template graphic files. There are many benefits of stylesheets:

- The printing characteristics (registration and plate marks, color separations, layers, etc.) can be encapsulated in the template and isolated from the XML content.
- Global changes to typefaces, text sizes, ink colors, etc. can be made by altering or replacing a stylesheet.
- The XML content can be reused with new or modified stylesheets to easily construct new graphical designs.

By building and maintaining a library of stylesheets, the content data is easily composed and standardized. Once composed in this manner, the form files can be utilized for conventional printing or re-purposed as both PDF and/or HTML Internet Forms with fillable response fields.

System Requirements for the XML Options

To use the XML Import/Export options, you will need to obtain and install the following Utility Software on your OneForm Designer Plus Windows PC's:

- The Java RunTime Environment from Sun is necessary for XSL stylesheets. This software is available at no charge from Sun Microsystems at:

<http://java.sun.com/getjava>

- OneForm Designer Plus uses the JAR File method to keep files together. The JAR Tool Binary is included in the Java 2 Platform, Standard Edition (J2SE) SDK and can be downloaded, extracted, and copied as needed. This is the only part of the SDK we currently need. The full SDK is available at no charge from Sun Microsystems at the following link:

<http://java.sun.com/j2se/1.4.2/download.html>

- For special character import and translations, install the open-source SED program. The default location selected by the install program (C:\Program Files\GnuWin32) is OK. The program is available at this link:

<http://easynews.dl.sourceforge.net/sourceforge/gnuwin32/sed-4.0.7-bin.exe>

- The Xalan XSL Processor (Java Version). This is included in the OneForm Designer Plus release software, and does not need to be downloaded.

Also, the following software requirement is necessary for saving directly to Acrobat PDF file format.

- Adobe Acrobat Distiller 5.0 or above is required. For best results, stylesheets should specify PostScript Fonts.

Use of JAR Files

OneForm Designer Plus uses the JAR File method to keep project files together. The XML Interactive Import can read a JAR file allowing the user to open XML content data files or any graphic files it contains. When opening a JAR file, its entire contents are extracted into a basename directory at that location.

During the Save process, you will also be allowed to optionally save the XML content data file, graphic file, and stylesheet referenced, for containment within a single JAR file. When a JAR file is saved, all files found in the basename folder are encapsulated in the JAR file.

The XML Document Type Definitions

Document Type Definition (DTD) files are XML support files. They declare what Document XML tags can be used during XML import. They are managed and distributed by Amgraf, Inc. with the OneForm Designer Plus software.

Current DTD Files available:

- OFDPBarcode.dtd
- OFDPBOCSurvey.dtd
- OFDPBOCSurveyEdits.dtd
- OFDPBOCSurveyPages.dtd
- OFDPDocument.dtd
- OFDPForm.dtd

GUID's Identify Each Object/Component

Each object/component of a page layout will be associated with a Globally Unique Identifier (GUID). This is a 128-bit number assigned to uniquely identify a component or object on a page layout. The GUID identifies each content object, and is preserved during import and export.

GUID's are generated by a program and are never repeated or reused. The GUID's for graphical components must be provided in the XML files imported into OneForm Designer Plus, and those GUID's will be stored with the composed forms.

GUID's can be assigned to objects/components that are "nested" within other objects that have a different GUID. In this manner, blocks of questions can be addressed as well as individual questions and responses. GUID requirements are found in the DTD files.

The Import XML File dialog "Show Detail" option will display GUID numbers (i.e., {BD29EA8A-4BD1-34B2-E034-080020CD8C15}) when checked.

OneForm Designer Plus utilizes the Microsoft CoCreateGuid function to generate GUID's on demand.

OneForm Designer Plus Profile

The OneForm Designer Plus Profile designates menu settings according to the type of forms needed: EForms/IForms, Paper Forms, and XML Workflow Forms are all needed for the display of the OFDP XML Import/Export menus. **Note:** If changing your profile, OneForm must be exited and restarted for the change to be applied.

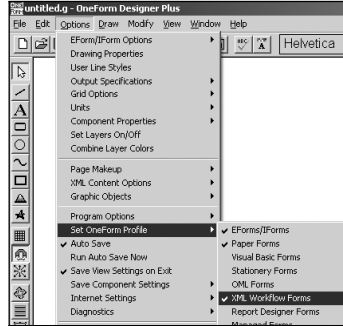


Figure 1-2: OneForm Profile Options

Glossary of Terms Used

DTD (Document Type Definition) – These XML support files have an extension of .DTD. They declare what Document XML tags can be used during XML import. They are managed and distributed by Amgraf, Inc. with the OneForm Designer Plus software.

GUID (Globally Unique Identifier) – A number assigned to uniquely identify each component or object imported. These numbers are generated by the program and are never repeated or reused.

ISE (Interactive Stylesheet Editor) – The OFDP XML QRM Stylesheet Editor option which simplifies the development of new XSL stylesheets.

JAR (Java Archive File Format) – A Java archive file has an extension .JAR and is used to store one or more files. They help to keep projects together such as the XML files, graphic files, stylesheet, etc. The "jar" command can be used to see the contents of a JAR file.

OFDP – OneForm Designer Plus Software by Amgraf, Inc.

PDF (Portable Document Format) – Adobe Acrobat Portable Document Format. OneForm's Save As XML dialog has a Save to PDF option.

QRM (Question/Response Metadata) – Data describing questions/responses.

XML (eXtensible Markup Language) – A coding method used to represent questionnaire data. These files have an extension of .XML.

XSD (eXtensible Schema Document) – The XML Interactive Stylesheet Editor reads an XSD file to identify stylesheet components.

XSL (eXtensible Stylesheet Language File) – The XML Stylesheet files have an extension of .XSL. These files customize the attributes of the XML tags used during XML import of the data and reference a base drawing page layout.

SAMPLE A

Automated Composition of Forms Using XML Objects

Diet and Exercise Magazine,
P.O. Box 9863, Newport, MO 68474

2004
Customer
Survey

8-1

Please help us better serve our customers by filling out this quick and easy survey. Your time is greatly appreciated and will be rewarded with a FREE one year's subscription.

Please provide subscription information:

1. What is your name? *Print name below.*
Last Name:
First Name: MI:
Address:
City:
State: Zip Code:
2. What is your mailing address? *Print address below.*
Address:
City:
State: Zip Code:
3. What is your telephone number?
Area Code + Number
4. What is your sex? *(Mark ONE box.)*
 Male
 Female
5. What is your age and date of birth?
Print numbers in boxes.
Age:
Month: Day: Year of Birth:
6. How many adults in your household are currently on diets?
Number of people:

7. I prepare most of my meals at home. *(Mark ONE box.)*
 Yes
 No
8. My main goal is to look more toned.
 Yes
 No
9. I measure my progress on a scale.
 Yes
 No
10. I eat more fast food than I should.
 Yes
 No
11. I am good at planning meals ahead of time.
 Yes
 No
12. I would like to have more energy.
 Yes
 No
13. I want to lose weight for health reasons.
 Yes
 No
14. I feel stressed when I am on a diet.
 Yes
 No
15. I enjoy sports and other physical activities.
 Yes
 No




Figure 1-2: Sample Questionnaire Form

Note 1: These XML File Examples do not include real GUID elements.

Note 2: Arrows are selectively used to show relationships between an XML tag and composed textual objects.

Form Title/Description Elements

Applicable DTD Tags:

```

<!ELEMENT FormInstruction (3CDATA | B | I | U | font-size)*>
<!-- General Instructions for entire Form -->
<!ELEMENT OMBExpirationDate (3CDATA)>
<!-- Form ID Number -->
<!ELEMENT OMBNumber (3CDATA)>
<!-- OMB assigned Form Number -->
<!ELEMENT OfficeAddress (3CDATA)>
<!-- Office/Department Address Block -->
<!ELEMENT FormNumber (3CDATA)>
<!-- Form ID Number -->
<!ELEMENT FormTitle (3CDATA | B | I | U | font-size)*>
<!-- Main Title for Form -->
<!ELEMENT Year (3CDATA)>
<!-- Year Number of this survey -->
<!ATTLIST QRM
    GUID CDATA #REQUIRED>
<!ELEMENT QRM (Year, FormTitle, FormNumber, OfficeAddress?, OMBNumber?,
    OMBExpirationDate?, FormFace+, FormInstruction*, InstructionItem*,
    GeneralComments*, QuestionGroup+)>
<!ELEMENT OFDPBOCSurvey (QRM)>
    
```

Graphical Representation after Import into OFDP:

XML Coding:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE OFDPBOCSurvey SYSTEM "OFDPBOCSurvey.dtd">
<!-- Generated by Oracle Reports version 3.0.2.0.1 -->
<OFDPBOCSurvey>
  <QRM GUID="{BD29EA8A-4BC3-34B2-E034-080020CD8C15}">
    <Year>2004</Year>
    <FormTitle>Customer Survey</FormTitle>
    <FormNumber>1</FormNumber>
    <OfficeAddress>Diet and Exercise Magazine, P.O.
      Box 9863, Newport, MO 68474</OfficeAddress>
    <OMBNumber/>
    <OMBExpirationDate/>
    <FormFace GUID="{BD29EA8A-4BC4-34B2-E034-080020CD8C15}"
      FaceNumber="1" Barcode="040401"/>
    <FormInstruction>
      <B><I>Please help us better serve our customers by
        filling out this quick and easy survey. Your time
        is greatly appreciated and will be rewarded with a
        FREE one year's subscription.</I></B>
    </FormInstruction>
  </QRM>
</OFDPBOCSurvey>
    
```

Question Group Elements

Applicable DTD Tags:

```
<!ELEMENT InstructionItemText (2CDATA | Number | IndentLevel | B | I | U |
font-size)*>
<!-- General Instruction Text String -->
<!ELEMENT QuestionNumber (2CDATA | B | I | U | font-size | IndentLevel)*>
<!-- Sequence Number for a Question -->
<!ELEMENT QuestionInstruction (2CDATA | B | I | U | font-size | IndentLevel)*>
<!-- Objects Allowed within Instructions -->
<!ATTLIST InstructionItem
    GUID CDATA #REQUIRED
    InstructionOrg (Vertical | Horizontal | Continued | NewColumn)
    #IMPLIED>
<!-- Indicator to Arrange Instructions Vertically or Horizontally -->
<!ELEMENT InstructionItem (InstructionItemText*)>
<!-- Instructions Style-->
<!ELEMENT GroupInstruction (2CDATA | B | I | U | font-size | IndentLevel)*>
<!-- Objects Allowed in a GroupInstruction -->
<!ATTLIST DataType
    Type (Any | Date | Currency | NumericFloat | NumericInteger | YesNo)
    #IMPLIED>
<!ELEMENT DataType (2CDATA)>
<!-- Fill Field Data Type -->
<!ELEMENT FieldName (2CDATA)>
<!-- Field Name for Response -->
<!-- Maximum number of characters allowed in field -->
<!-- Value Followed by Units, inch implied -->
<!ATTLIST ResponseType
    BoxHsize CDATA #IMPLIED
    BoxVsize CDATA #IMPLIED
    CharCount CDATA #IMPLIED
    Type (CheckBox | CombSpacedBox | TextBox) #REQUIRED>
<!ELEMENT ResponseType (2CDATA)>
<!ELEMENT CodeNumber (2CDATA)>
<!ELEMENT FieldBoxLabel (2CDATA | B | I | U)*>
<!-- Caption/Label for a Field -->
<!ATTLIST Response
    GUID CDATA #REQUIRED
    ResponseOrg (Vertical | Horizontal | NewColumn) #IMPLIED>
<!ELEMENT Response (CodeNumber, FieldName, DataType, ResponseType,
    FieldBoxLabel)>
<!-- Fill Field for Response -->
<!ELEMENT QuestionText (3CDATA | Number | B | I | U | font-size |
    IndentLevel)*>
<!ATTLIST Question
    GUID CDATA #REQUIRED
    AllowResponse (Multiple | Single) #IMPLIED>
<!ATTLIST Question
    ResponseOrg (Vertical | Horizontal) #IMPLIED>
<!ELEMENT Question (QuestionNumber, QuestionText, (GeneralComments |
    InstructionItem | QuestionInstruction | Response+)*)>
<!-- Objects Allowed in a Question -->
```

XML Coding for Question Group One:

```

<QuestionGroup GUID="{BD29EA8A-4BCD-34B2-E034-080020CD8C15}"
  GroupName="Subscriber">
  <Question GUID="{BD7D5C43-4978-3D7B-E034-080020CD8C15}"
    <InstructionItem GUID="{BD7D5C43-4979-3D7B-E034-080020CD8C15}"
      InstructionOrg="Vertical">
      <InstructionItemText>
        <B>Please provide subscription information:</B>
      </InstructionItemText>
    </InstructionItem>
  </Question>
  <Question GUID="{BD29EA8A-4BD2-34B2-E034-080020CD8C15}"
    AllowResponse="Multiple">
    <QuestionNumber>
      <B>1.</B>
    </QuestionNumber>
    <QuestionText>
      <B>What is your name?</B>
      <I> Print name below.</I>
    </QuestionText>
    <QuestionInstruction/>
  </Question>

```

```

<Response GUID="{BD29EA8A-4C35-34B2-E034-080020CD8C15}"
  <FieldName>P1_NAME_LAST </FieldName>
  <ResponseType Type="CombSpacedBox" CharCount="15"
    <FieldBoxLabel>
      <B>Last Name</B>
    </FieldBoxLabel>
  </Response>
  <Response GUID="{BD29EA8A-4C34-34B2-E034-080020CD8C15}"
    <FieldName>P1_NAME_FIRST </FieldName>
    <ResponseType Type="CombSpacedBox" CharCount="13"
      <FieldBoxLabel>
        <B>First Name</B>
      </FieldBoxLabel>
    </Response>
  <Response GUID="{BD29EA8A-4C36-34B2-E034-080020CD8C15}"
    <FieldName>P1_NAME_MI </FieldName>
    <ResponseType Type="CombSpacedBox" CharCount="1"
      <FieldBoxLabel>
        <B>MI</B>
      </FieldBoxLabel>
    </Response>
</Question>

```

XML Coding for Question Group Two:

```

<Question GUID="{BD29EA8A-4BD5-34B2-E034-080020CD8C15}"
  AllowResponse="Multiple">
  <QuestionNumber>
    <B>2.</B>
  </QuestionNumber>
  <QuestionText>
    <B>What is your mailing address?</B>
    <I>Print address below.</I>
  </QuestionText>
  <QuestionInstruction/>
  <Response GUID="{BD29EA8A-4D35-34B2-E034-080020CD8C15}"
    ResponseOrg="Vertical">
    <FieldName>P1_ADDRESS </FieldName>
    <DataType>Any </DataType>
    <ResponseType Type="CombSpacedBox" CharCount="15"
      BoxHsize=".2i" BoxVsize=".22i"/>
    <FieldBoxLabel>
      <B>Address</B>
    </FieldBoxLabel>
  </Response>
  <Response GUID="{BD29EA8A-4D34-34B2-E034-080020CD8C15}"
    ResponseOrg="Vertical">
    <FieldName>P1_CITY </FieldName>
    <ResponseType Type="CombSpacedBox" CharCount="15"
      BoxHsize=".2i" BoxVsize=".22i"/>
    <FieldBoxLabel>
      <B>City</B>
    </FieldBoxLabel>
  </Response>
  <Response GUID="{BD29EA8A-4D36-34B2-E034-080020CD8C15}"
    ResponseOrg="Horizontal">
    <CodeNumber/>
    <FieldName>P1_STATE </FieldName>
    <ResponseType Type="CombSpacedBox" CharCount="2"
      BoxHsize=".2i" BoxVsize=".22i"/>
    <FieldBoxLabel>
      <B>State</B>
    </FieldBoxLabel>
  </Response>
  <Response GUID="{BD29EA8A-4D37-34B2-E034-080020CD8C15}"
    <FieldName>P1_ZIPCODE </FieldName>
    <ResponseType Type="CombSpacedBox" CharCount="5"
      BoxHsize=".2i" BoxVsize=".22i"/>
    <FieldBoxLabel>
      <B>Zip Code</B>
    </FieldBoxLabel>
  </Response>
</Question>

```

XML Coding for Question Group Three:

```

<QuestionGroup GUID="{BD29EA8A-4BCC-34B2-E034-080020CD8C15}"
  GroupName="MOMB-HH">
  <Title/>
  <GroupInstruction/>
  <GeneralComments/>
  <Question GUID="{BD7D5C43-496D-3D7B-E034-080020CD8C15}"
    AllowResponse="Multiple">
    <QuestionNumber>
      <B>3.</B>
    </QuestionNumber>
    <QuestionText>
      <B>What is your telephone number?</B>
    </QuestionText>
    <QuestionInstruction/>
    <InstructionItem GUID="{BD7D5C43-496E-3D7B-E034-080020CD8C15}"
      InstructionOrg="Vertical">
      <InstructionItemText>Area Code + Number </InstructionItemText>
    </InstructionItem>
    <Response GUID="{BD7D5C43-4970-3D7B-E034-080020CD8C15}"
      ResponseOrg="Vertical">
      <CodeNumber/>
      <FieldName>HH_PHONE_AREA </FieldName>
      <DataType>NumericInteger </DataType>
      <ResponseType Type="CombSpacedBox" CharCount="3"
        BoxHsize=".2i" BoxVsize=".22i"/>
      <FieldBoxLabel> </FieldBoxLabel>
    </Response>
    <Response GUID="{BD7D5C43-4972-3D7B-E034-080020CD8C15}"
      ResponseOrg="Horizontal">
      <CodeNumber/>
      <FieldName>HH_PHONE_PREFIX </FieldName>
      <DataType>NumericInteger </DataType>
      <ResponseType Type="CombSpacedBox" CharCount="3"
        BoxHsize=".2i" BoxVsize=".22i"/>
      <FieldBoxLabel> </FieldBoxLabel>
    </Response>
    <Response GUID="{BD7D5C43-4974-3D7B-E034-080020CD8C15}"
      ResponseOrg="Horizontal">
      <CodeNumber/>
      <FieldName>HH_PHONE_SUFFIX </FieldName>
      <DataType>NumericInteger </DataType>
      <ResponseType Type="CombSpacedBox" CharCount="4"
        BoxHsize=".2i" BoxVsize=".22i"/>
      <FieldBoxLabel> </FieldBoxLabel>
    </Response>
  </Question>
</QuestionGroup>

```

3. What is your telephone number?
Area Code + Number

1 2 3

XML Coding for Question Group Four:

```
<QuestionGroup GUID="{BD29EA8A-4BDD-34B2-E034-080020CD8C15}"
  GroupName="MOMB-HM">
  <Title/>
  <GroupInstruction/>
  <GeneralComments/>
  <Question GUID="{BD29EA8A-4BD3-34B2-E034-080020CD8C15}"
    AllowResponse="Multiple">
    <QuestionNumber>
      <B>4.</B>
    </QuestionNumber>
    <QuestionText>
      <B>What is your sex?</B>
      <> Mark &XMark; ONE box.</>
    </QuestionText>
```

4. What is your sex? Mark ONE box.
 Male
 Female

```
<QuestionInstruction/>
<Response GUID="{BD29EA8A-4C00-34B2-E034-080020CD8C15}"
  ResponseOrg="Vertical">
  <CodeNumber/>
  <FieldName>P1_SEX_MALE </FieldName>
  <DataType>YesNo </DataType>
  <ResponseType Type="CheckBox" CharCount="1" BoxHsize=".125i"
    BoxVsize=".125i"/>
  <FieldBoxLabel>Male </FieldBoxLabel>
</Response>
<Response GUID="{BD29EA8A-4C01-34B2-E034-080020CD8C15}"
  ResponseOrg="Horizontal">
  <CodeNumber/>
  <FieldName>P1_SEX_FEMALE </FieldName>
  <DataType>YesNo </DataType>
  <ResponseType Type="CheckBox" CharCount="1" BoxHsize=".125i"
    BoxVsize=".125i"/>
  <FieldBoxLabel>Female </FieldBoxLabel>
</Response>
</Question>
```

XML Coding for Question Group Five:

```
<Question GUID="{BD29EA8A-4BD4-34B2-E034-080020CD8C15}"
  AllowResponse="Multiple">
  <QuestionNumber>
    <B>5.</B>
  </QuestionNumber>
  <QuestionText>
    <B>What is your age and date of birth?</B>
  </QuestionText>
```

XML Coding for Question Group Five (Continued):

```

<QuestionInstruction/>
<InstructionItem GUID="{C06324A8-5B53-64EC-E034-080020CD8C15}"
  InstructionOrg="Horizontal">
  <InstructionItemText>
    <l>Print numbers in boxes.</l>
  </InstructionItemText>
</InstructionItem>
<Response GUID="{C06324A8-5B51-64EC-E034-080020CD8C15}"
  ResponseOrg="Vertical">
  <FieldName>P1_AGE </FieldName>
  <DataType>NumericInteger </DataType>
  <ResponseType Type="CombSpacedBox" CharCount="3"
    BoxHsize=".2i" BoxVsize=".22i"/>
  <FieldBoxLabel>
    <B>Age</B> </FieldBoxLabel>
</Response>

```

5. What is your age and date of birth?
Print numbers in boxes.

Age

Month Day Year of Birth

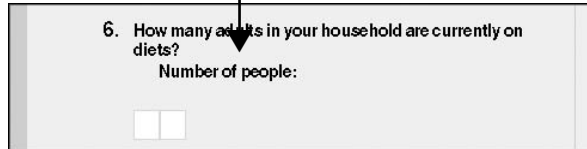
```

<Response GUID="{C06324A8-5B55-64EC-E034-080020CD8C15}"
  <FieldName>P1_DOB_MONTH </FieldName>
  <DataType>NumericInteger </DataType>
  <ResponseType Type="CombSpacedBox" CharCount="2"
    BoxHsize=".2i" BoxVsize=".22i"/>
  <FieldBoxLabel>
    <B>Month</B>
  </FieldBoxLabel>
</Response>
<Response GUID="{C06324A8-5B57-64EC-E034-080020CD8C15}"
  <FieldName>P1_DOB_DAY </FieldName>
  <DataType>NumericInteger </DataType>
  <ResponseType Type="CombSpacedBox" CharCount="2"
    BoxHsize=".2i" BoxVsize=".22i"/>
  <FieldBoxLabel>
    <B>Day</B>
  </FieldBoxLabel>
</Response>
<Response GUID="{C06324A8-5B59-64EC-E034-080020CD8C15}"
  <FieldName>P1_DOB_YEAR </FieldName>
  <DataType>NumericInteger </DataType>
  <ResponseType Type="CombSpacedBox" CharCount="4"
    BoxHsize=".2i" BoxVsize=".22i"/>
  <FieldBoxLabel>
    <B>Year of Birth</B>
  </FieldBoxLabel>

```


XML Coding for Question Group Six:

```
<Question GUID="{BD29EA8A-4BD1-34B2-E034-080020CD8C15}"
  AllowResponse="Multiple">
  <QuestionNumber>
    <B>6.</B>
  </QuestionNumber>
  <QuestionText>
    <B>How many adults in your household are currently
      on diets?</B>
  </QuestionText>
  <QuestionInstruction/>
  <InstructionItem GUID="{BD29EA8A-4BF3-34B2-E034-080020CD8C15}"
    InstructionOrg="Vertical">
    <InstructionItemText>
      <B>Number of people:</B>
    </InstructionItemText>
  </InstructionItem>
  <Response GUID="{BD29EA8A-4BF1-34B2-E034-080020CD8C15}"
    ResponseOrg="Horizontal">
    <CodeNumber/>
    <FieldName>H1_POP </FieldName>
    <DataType>NumericInteger </DataType>
    <ResponseType Type="CombSpacedBox" CharCount="2"
      BoxHsize=".2i" BoxVsize=".22i"/>
    <FieldBoxLabel> </FieldBoxLabel>
  </Response>
</Question>
```



The screenshot shows a question box with a light gray background. At the top, it displays the question number and text: "6. How many adults in your household are currently on diets?". Below this, the instruction "Number of people:" is shown. At the bottom of the box, there are two small, empty white input fields for the user to enter their answer.

(Questions 7 through 15 are very similar to Question 4.)

CHAPTER 2

Using XML Interactive Import

This option will import the selected XML Content based on the Stylesheet designated. It is able to import XML objects one at a time or automatically flow the XML objects into the columns chronologically, until the function has reached the end of the XML file or filled all available column space.

XML Interactive Import is found under **File, Import**. It is not necessary to start with a blank page, it will prompt if the existing file needs saved.

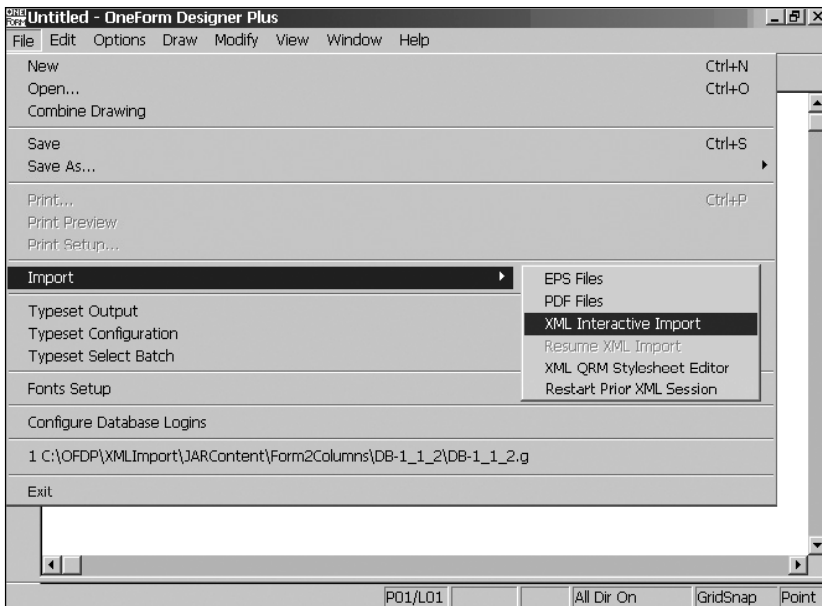


Figure 2-1: XML Interactive Import Menu

The first step is to locate the XML Content file and a Stylesheet.

Note: The display of the OFDP XML Import/Export menus are toggled on by the OneForm Designer Plus Profile. See *page 4* if the XML menus are not available.

XML Content Open/Processing Options

After selecting XML Interactive Import the following dialog will appear for selecting either an XML CONTENT file or a JAR file. OneForm Designer Plus uses the JAR File method to keep files of a project together. A JAR file may contain the XML Content data file, a composed graphic layout, or a revised XML Content data file containing edits.

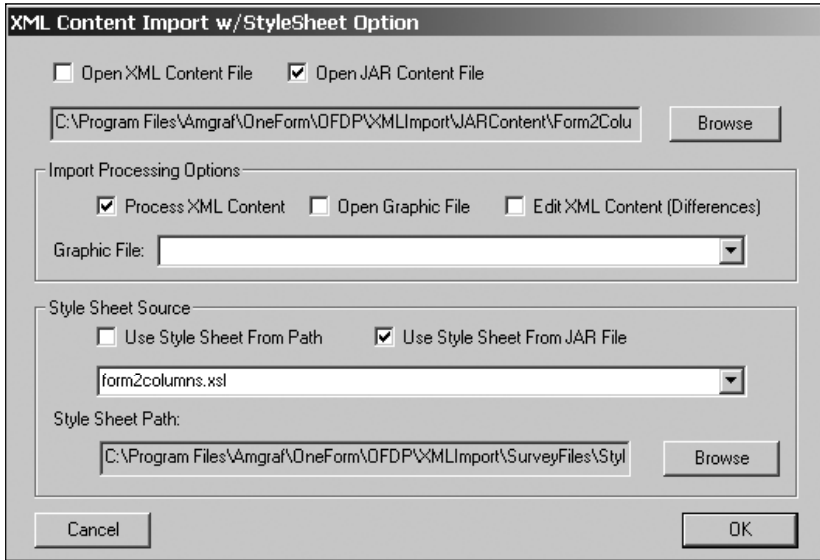


Figure 2-2: Opening XML Content from a JAR File

Figure 2-2 is a perfect example of selecting an XML Content file for processing that resides in a JAR file. The stylesheet source will be the one from the JAR file. Clicking [OK] will process the request and a brief processing window will appear. Then, the template indicated in the stylesheet will be loaded and the XML Content data will be prepared for importing and displayed within the XML Import File dialog. See Figure 2-3.

Open Options

If you *Open an XML Content File*, you are then able to process its contents for new layout with any desired stylesheet. This option is for creating new forms.

If you *Open a Jar File*, then you have three processing options available: Processing the XML Content File, Opening a Graphic File, or Editing XML Content Differences.

Processing Options

Processing the XML Content File: This option will load the CURRENT XML Content data file within the JAR file for processing. It will use the stylesheet

found within the JAR file unless directed otherwise. This option is for creating new forms.

Opening a Graphic File: This option will allow you to open a ".g" graphic file within the JAR file for editing.

Editing XML Content Differences: This option will compare the PREVIOUS and CURRENT XML data files within the JAR file and load only differences for processing. It will open the existing graphic file allowing the differences to be included.

Selecting a Stylesheet

The stylesheet controls the look of the document. There are two options for selecting a stylesheet. You can either [Browse] and *Use Stylesheet from Path* or have the program *Use the Stylesheet from the JAR File*.

Note: The current stylesheet XSL file is copied into the JAR file, every time you save a JAR file.

XML Import Compose Options

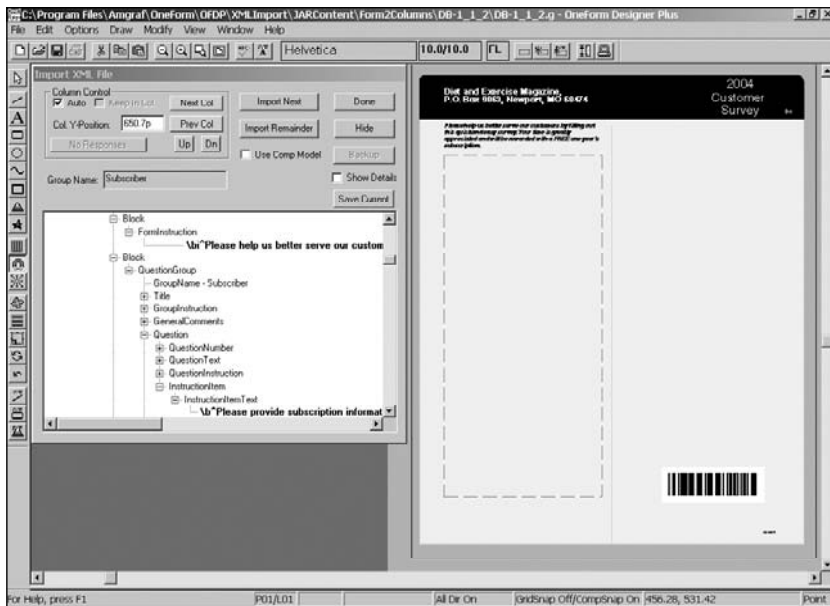


Figure 2-3: XML Data is Ready for Importing

Now you are ready to import and compose data into the columns.

Using the Import Options

The Import XML File dialog allows you to compose objects from the XML content data. The objects will be composed into the template according to the stylesheet specifications. You can import one GUID object at a time by clicking the **[Import Next]** button. All XML objects are processed chronologically. No object can be skipped.

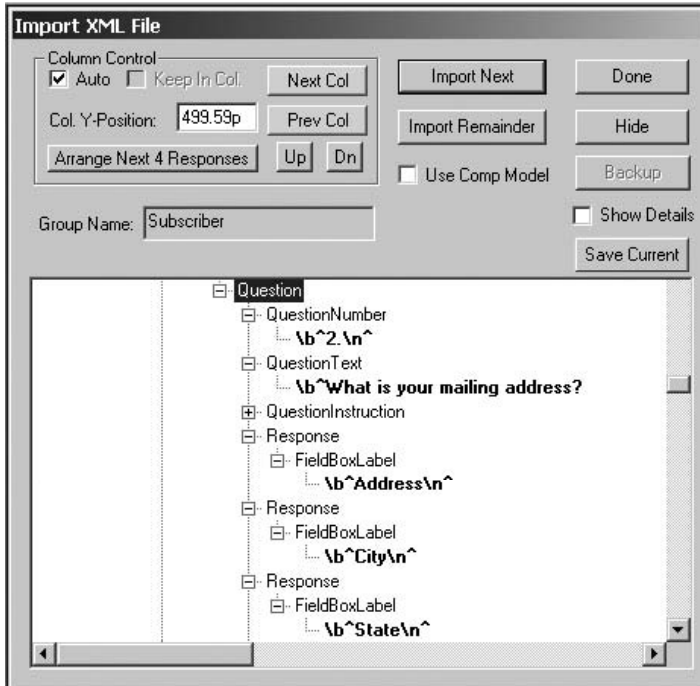


Figure 2-4: Use **[Import Next]** for One Object

The Column **[Up]** and **[Down]** buttons allow you to reposition the compose location on the page before importing the next set of components. You will see the top of the column boundary move to the new location.

[Import Remainder] will compose all remaining XML Content based on the stylesheet selected. All response fields will flow Top-to-Bottom, one on each row, without an opportunity to adjust their arrangement.

A GUID object usually consists of a question and its responses. If you do not like the arrangement of the response fields, you can **[Backup]** the entire item and rearrange them through the **[Arrange Next Responses]** dialog.

The "Show Detail" option will display stylesheet information and GUID numbers.

The [Hide] option closes the dialog temporarily. You can resume the import, through the File, Import, Resume XML Import. When you are finished importing, click [Done].

Using Arrange Responses

In the example below, the first question has three responses: Last Name, First Name, and Middle Initial. They will flow Top-to-Bottom, one on each line as a default.

If this is unacceptable, [Backup] and go to the [Arrange Responses] button and specify the rows and columns needed before importing again.

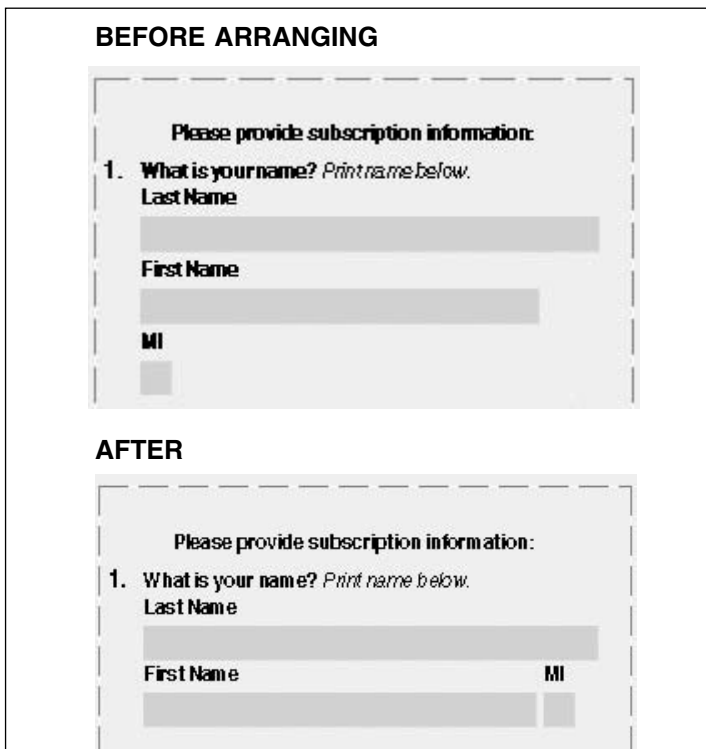


Figure 2-5: Arranging Responses

Anytime you want responses to be side-by-side, (as shown with the First Name and Middle Initial), they will need to be arranged. You are rearranging the direction of flow.

Note: The size of the response fields will not change. Arranging allows you to change their placement before importing, not their size.

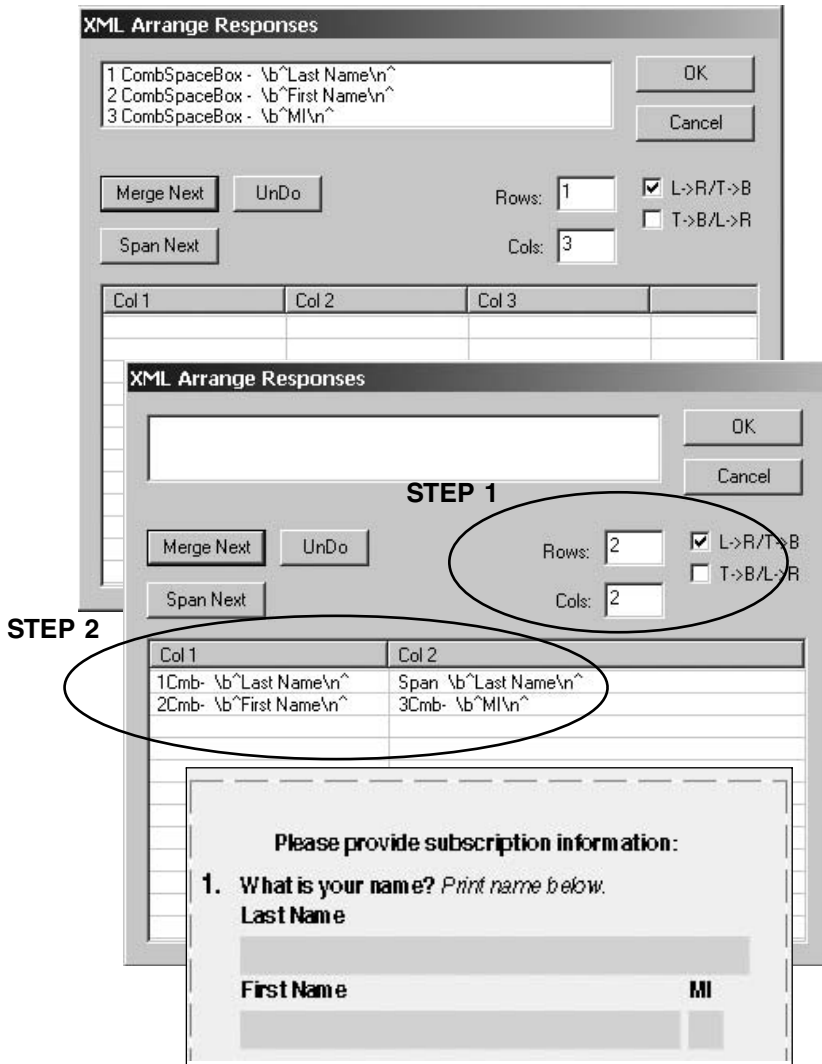


Figure 2-6: Arranged Data is Ready for Importing

Step 1: Above is an example of rearranging the responses to fall in 2 Rows and 2 Columns, Left-to-Right. **You must key in the Rows and Columns.**

Step 2: The [Merge] and [Span] buttons are used to place the responses into the rows and columns. Click [Merge] to place the Last Name field. Click [Span] to hold the location across the row. Now, pick [Merge] twice, to place the First Name and Middle Initial responses side-by-side. When you are satisfied with the row/column placement, click [OK] and [Import Next].

Import Remainder

This option will import and compose all remaining XML Content based on the stylesheet selected. The XML Content will flow into the designated columns chronologically (Auto Checked) until the function has reached the end of the XML file or filled all available column space.

All response fields will flow Top-to-Bottom, one on each row, without an opportunity to adjust their arrangement.

Diet and Exercise Magazine,
P.O. Box 9863, Newport, MO 68474

2004
Customer
Survey S-1

Please help us better serve our customers by filling out this quick and easy survey. Your time is greatly appreciated and will be rewarded with a FREE one year's subscription.

Please provide subscription information:

1. What is your name? *Print name below.*

Last Name

First Name

MI

2. What is your mailing address? *Print address below.*

Address

City

State

Zip Code

3. What is your telephone number?
Area Code + Number

4. What is your sex? *Mark ONE box.*

Male
 Female

5. What is your age and date of birth?
Print numbers in boxes.

Age

Month

Day

Year of Birth

6. How many adults in your household are currently on diets?
Number of people:

7. I prepare most of my meals at home. *Mark ONE box.*

Yes
 No

8. My main goal is to look more toned.

Yes
 No

9. I measure my progression on a scale.

Yes
 No

10. I eat more fast food than I should.

Yes
 No

11. I am good at planning meals ahead of time.

Yes
 No

12. I would like to have more energy.

Yes
 No

Figure 2-7: Import Remainder with Column Control Auto

Note: If the XML input file has not been exhausted, a warning message will appear stating you have "run out of columns." At this time, it is the responsibility of the user to manually edit the XML file (or modify the stylesheet), to re-import any remaining XML code into a second file.

Saving the Graphic File

When you are finished composing XML data, selecting [Done] will close the XML Import option. At this time you can Save the graphic file as normal.

The name of the graphic file comes from the basename of the XML file with an extension ".g": Form_Number.g

If you opened a JAR file in the beginning, it will prompt you to save the JAR file. This will copy the new graphic file into the JAR file. Otherwise, you are just saving the graphic file into the basename directory that the JAR file created.

[Save Current] During Import

As you are working, you can save the current state by clicking the [Save Current] button. This will track the question you last imported and save the graphic file as it currently stands. You can restart the importing session at any time by using the File, Import, Restart Prior XML Session menu. *Note:* Each time you click the button it will overwrite the previous session's tracking.

XML Save (Export) Option

Exporting to XML data is found under the **File, Save As, Save As XML**.

The XML Save Dialog has four distinct sections. Each is described below:

Component Options: Select the type of components to be saved. Default is to save All Components. The other three options save their respective component types exclusively.

EForm Field Options: This allows you to save Full Definitions of the fields or just certain properties, such as position and size.

File Options: When saving to XML, there is an ability to save three separate XML files containing: Content, Layout, and/or Fields. The user will also have the opportunity to save an Acrobat PDF file at the same time.

The files will be named automatically in the following manner:

Content File: Form_Number_CURRENT_content.xml

Layout File: Form_Number_layout.xml

Fields w/Coordinates: Form_Number_fields.xml

Save As Jar File: For ease of use, compression, and to keep all associated files in a single container, the XML files can be optionally encapsulated and saved within a single JAR file. When this option is checked, the file name will use the following format:

Jar File: Form_Number.jar.

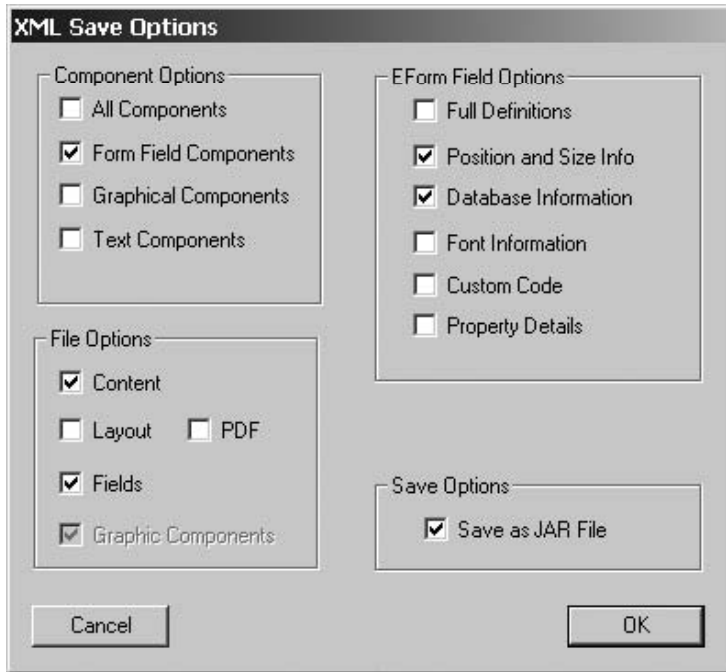


Figure 2-8: Saving to XML and PDF

Saving as a PDF

The PDF option when checked will automatically create two files which will reside in the current directory:

PDF File: Form_Number_proof.pdf

PostScript File: Form_Number_.ps

This save option will then automatically open Acrobat Distiller and create a PDF file using Amgraf's XMLOutput.joboptions.

By default, XML Import produces single-part forms. The Output Specification dialog will only appear if more than one Part is found in the file. For forms with just a Part 1, the default specification is automatically used. **Check a Part for Output** only if the dialog appears.

An Output dialog with *Adobe Acrobat* as the Printer Name will appear, **click [OK]**.

The screen will display two black processing windows and then Acrobat Distiller. When the PDF file is finished, Acrobat Distiller will close automatically.

Save As, Save as Jar Option

This option is only available if the XML Import has been used to open a .jar file. This will then save all of the XML file variations possible according to the options selected when the last "Save as XML" was invoked.

It will automatically run through the save process without presenting any dialogs or asking for any file names.

CHAPTER 3

Creating Base Drawing Templates

The imported XML data is always brought into a base drawing. These are pre-designed graphic template files stored within the *BaseDrawing* directory and are referenced in each Stylesheet XSL file. Following are the default settings:

- The template file's lower-left paper trim must be at the X and Y coordinate of 0/0. This is a requirement for proper printing and scanning of electronic field coordinates.
- Printing is controlled by a required Part's Output Specification.
- It is recommended that Layer 22 not be used on a base drawing. The XML Synchronization Option uses it to highlight changes made to the data.

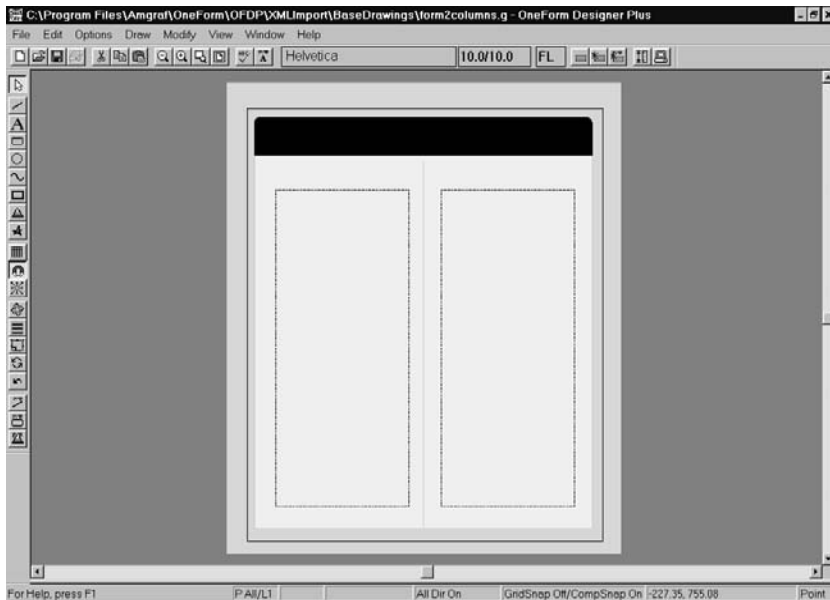


Figure 3-1: Base Drawing for Two Columns

On the next page, you will find step-by-step instructions for creating a new template.

Steps for Creating a New Template

Following you will find the basic procedures for creating a base drawing template. Our example will create a template that holds a standard letter size page (8.5i x 11i) that has .25i outside margins to hold trim marks, file comments, etc.

1. Start a new drawing and setup new drawing properties with the following dimensions:

Form Width = 9i

Form Depth = 11.5i

Form X Origin = -.25i

Form Y Origin = -.25i

This will create a page that has the 0/0 coordinates a quarter inch up and over, allowing room for outside information.

2. Create a 0-weight Rectangle for the boundary that is to show the letter size image area.
 - a. Set the **Line Properties tool** to have a **0-weight**, also the layer should be changed to some other than 1. Possibly, the same layer as the trim marks.
 - b. Click the **Rectangle tool** and press **Shift-[F12]** for absolute position.
 - c. This will bring up the absolute position dialog, type in **0** for both of the XPos and YPos fields. Click **[OK]**. When moving the mouse, the first corner of the rectangle will be locked at 0/0.
 - d. Press **[F12]** for a relative location. This will allow you to type in the rectangle's width and height. Type in **8.5i [right arrow]** and **11i [up arrow]** then press **[Enter]**.

The rectangle should appear, exactly a quarter inch from each side of the page boundary.

3. Add any graphical elements needed on the base drawing, background screens, ruling, logos, etc.
4. Define the Parts Output Specification according to what layers should print on each film. (See Typeset Output Specifications for more information.)
5. Use **Save As, Save All Components**, to save the template into the *BaseDrawing* directory with a filename that has an extension of ".g" which means it is a graphic file.

This base drawing template is now ready to be referenced within the XSL Stylesheet.

Moving an Existing Component to Absolute 0/0

There are times when you want to use an existing file or move an existing component to an absolute coordinate. Use the Shift-[F12] to do this.

Since our base drawing templates need to have the trim marks at the 0/0 coordinate, you can group and move the entire form using the Shift-[F12] function for moving to an absolute position.

1. Select the **Move tool**
2. **Pick the component** to move.
3. **Pick the handle** on the component.
4. **Press Shift-[F12]**. This will bring up the absolute position dialog, type in **0** for both of the XPos and YPos fields. Click **[OK]**.

The component's handle that is picked will move to the 0/0 location. If it is part of a group, the entire group moves.

Changing a Page Boundary 0/0 Coordinate

Also remember, that OneForm Designer when creating a new page always starts the lower-left corner of the page at 0/0. If you need room outside the trim for comments, go to *Options, Drawing Properties* and change the *Form's X and Y Origin* to be a negative value "margin" amount. Then increase the *Form's Width and Depth* to add that "margin" amount.

Getting Coordinates for Stylesheets

Stylesheets work from X/Y coordinates of columns and blocks. The ISE option reports these coordinates. If you need other coordinates for design purposes, get them from the base drawing by using the Box Tool to draw invisible boundaries representing the location needed.

Each Box component will have its X/Y coordinates in the pop-up menu Location Properties. Use the Select cursor to pick the Box, then right-mouse for the pop-up menu. Select Location Properties to get the coordinates.

CHAPTER 4

Interactive Stylesheet Editor

The style of the imported composed content data is controlled by XML Stylesheets. They define attributes such as typefaces, text sizes, ink colors, column and fixed block positions, as well as the base template used for the column-by-column data flow.

These controls are defined in the Stylesheet XSL file with XML commands and tags. The Interactive Stylesheet Editor (ISE) within OFDP simplifies the development of new XSL stylesheets and allows maintenance of XSL files through easy to follow dialogs without a need to understand the nuances of the XML language syntax.

The Stylesheet library resides at *C:/Program Files /Amgraf /OneForm /OFDP /XMLImport/SurveyFiles/Stylesheets*. To make a new stylesheet, you must open an existing one, then modify it, instead of starting from scratch. The XML QRM Stylesheet Editor has a Save As option allowing you to save the changes into a new stylesheet file.

By building and maintaining a library of stylesheets, content data is easily composed and standardized. Once composed in this manner, the form files can be utilized for conventional printing or re-purposed as both PDF and/or HTML Internet Forms with fillable response fields.

Stylesheet Components

The XSL stylesheet files consist of instructions for interpreting and processing the XML content. The ISE provides a graphical user interface to specify the transformation values for these specific XSL elements:

Blocks with Text:

- Year (and FormTitle)
- FormNumber
- OfficeAddress
- OMBNumber (and Date)
- FormInstruction

Barcode Block:

- FormFace

Instruction, Questions, & Label Text:

- InstructionItem
- QuestionNumber
- QuestionText
- QuestionInstruction
- FieldBoxLabel

Response Fields:

- CheckBox
- CombSpacedBox
- TextBox
- ResponseType
- FaintText

Opening a Stylesheet File

The ISE user interface is opened by starting OFDP, clicking the File, Import menu, and then the XML QRM Interactive Stylesheet Editor option.

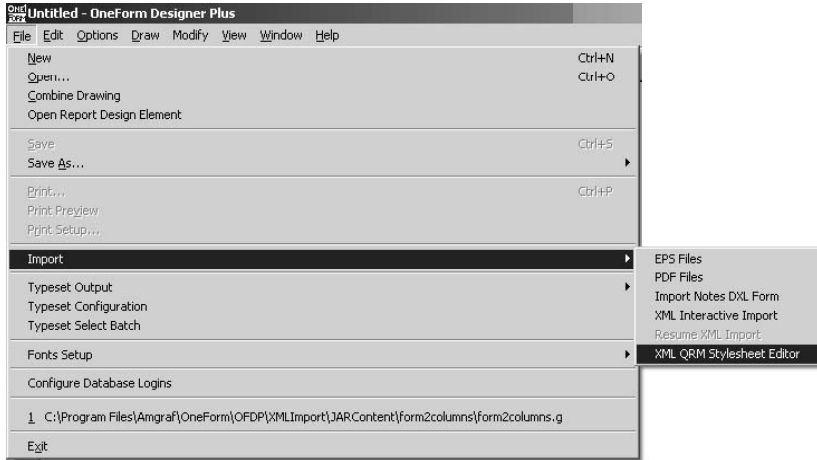


Figure 4-1: Starting the Interactive Stylesheet Editor

The ISE user interface dialog panels will be displayed as shown on the following pages.

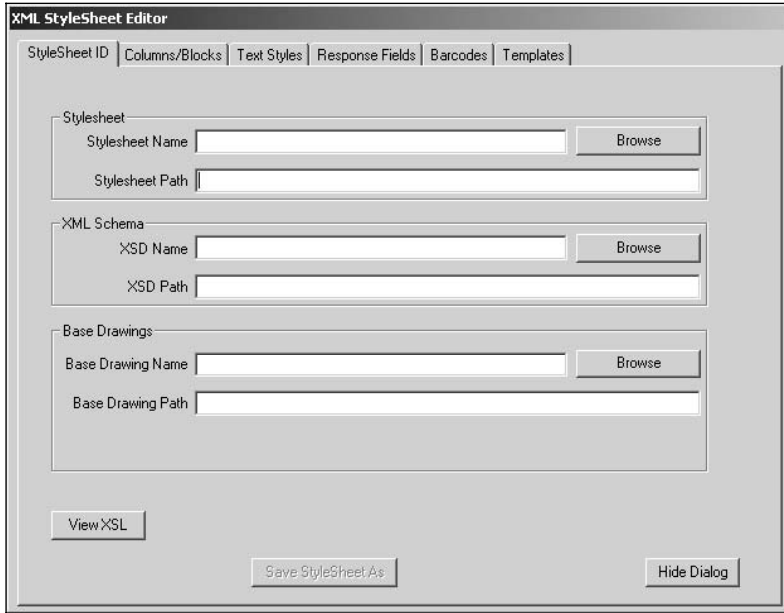


Figure 4-2: The Stylesheet ID Panel

The first tabbed panel called "Stylesheet ID" is used to specify the name and location of an existing stylesheet. This panel also includes the locations of the supporting XML Schema and Base Drawing template files.

When a stylesheet is opened, the style attributes of each component (i.e. typeface) are interpreted and the actual values are displayed in the appropriate dialog panel. In addition, the Base Drawing referenced in the stylesheet will be opened automatically.

This tab also holds buttons for saving or closing the XSL stylesheet files. All saves are a Save As.

Since the dialog is rather large there is a [Hide Dialog] button which will hide it until needed again. At that time you would go back to the View menu and toggle the XSL Stylesheet Editor back on. This will resume exactly where you left off.

The [View XSL] button shows the XSL command hierarchy of the file.

The Columns/Blocks Tab

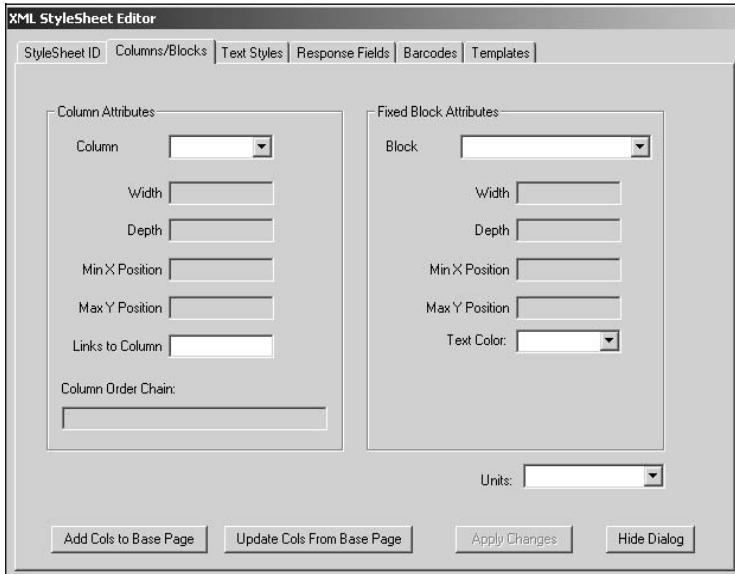


Figure 4-3: The Columns and Blocks Panel

The panel called "Columns/Blocks" shows the current coordinates and sizes of the rectangular areas for flowing content data. The values in this panel are updated as columns are drawn or blocks are moved by selecting the [Update Cols From Base Page] button.

Columns allow the text content to flow from one into another sequentially. Their order is shown in the "Column Order Chain". New Columns can be defined through the OFDP Page Makeup option described later in this document.

Blocks are fixed position areas linked to standardized content (i.e. Form Title) and cannot be created, but can be Moved and Resized.

[Add Columns to Base Page], [Update Columns From Base Page], [Apply Changes]

These buttons are used to transfer the columns and blocks positions from the XSL file to the graphic, and then from the graphic back into the XSL file. When you first open the XSL file click [Add Cols to Base Page] to transfer it from the XSL file to a visual on the graphic.

If you Move or Define a New Column graphically, click the [Update Columns from Base Page] to pull the attributes back into the ISE Dialog and the XSL file.

Working with Columns and Blocks

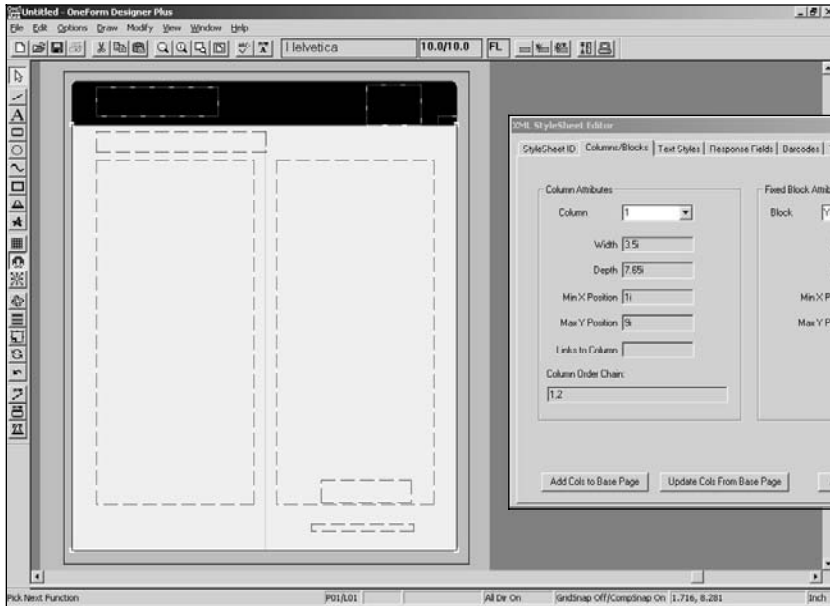


Figure 4-4: Column and Block Boundaries are Overlaid on the Base Page

To see a visual of these columns and blocks, click the button labeled [Add Cols to Base Page]. This will overlay the column and block boundaries on the base page template.

To see the base page more clearly, move the ISE dialog panel out of the window as shown.

Column boundaries are displayed as purple dashed lines. Block boundaries are displayed as red dashed lines. To identify the name/number of a column, use the select cursor to click the boundary, then right-button click and go to the "Column Properties".

On the tab, click the dropdown list titled "Column" to see how many columns are defined and their values. The dialog reports the X/Y upper-left corner coordinate and the width and depth. If you modify the column position or size on the page, click [Update Columns From Base Page] to transfer it from the visual on the graphic to the XSL file.

Modifying a Column or Block

Right button click on the selected column and choose the "Modify Column Size" function. Click on the side of the column you wish to modify and the new location. (This option works just like the "Modify Box" option.)

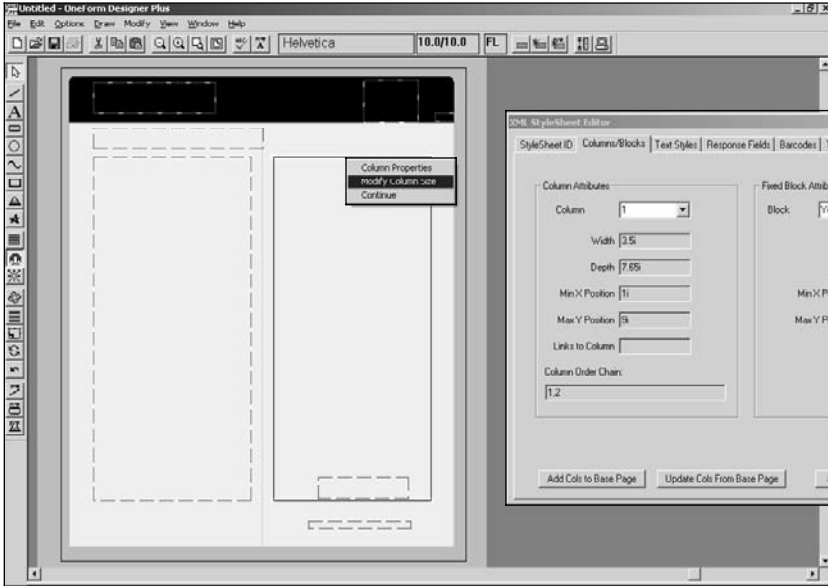


Figure 4-5: Using the Modify Column Size Function

Click the ISE button labeled [Update Cols from Base Page] to store the modified column values into the stylesheet.

The block components also have a "Modify Size" option.

Adding a New Column

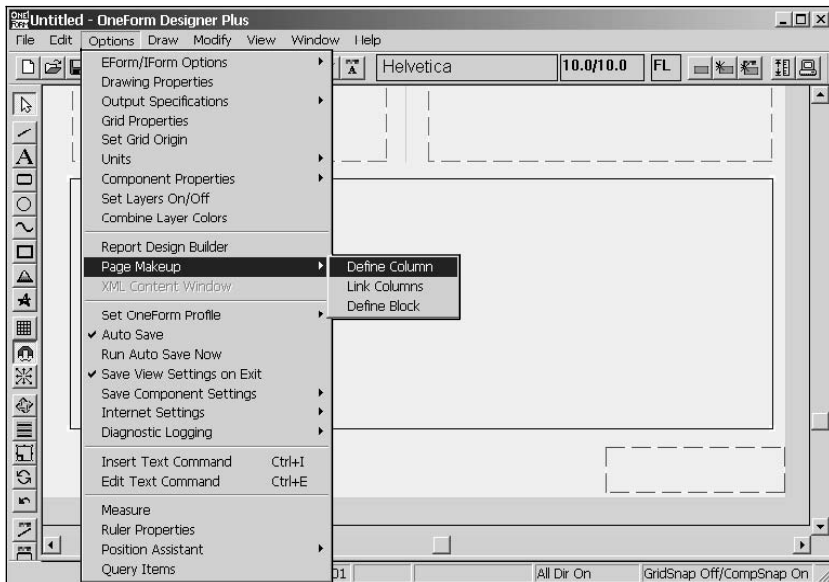


Figure 4-6: Page Makeup Option

A new column can be drawn on the base drawing through the Options, Page Makeup, Define Column. Here you will draw where the new designated column area should be. It will automatically name the new column with the next available number. Using the Select Arrow, you can identify it and modify its size or look at its properties.

The OFDP Copy and Move Tools can be used with columns. When a column is copied the column is named and linked consecutively from the last column defined.

Then, click the ISE button labeled [Update Cols from Base Page] to store the new column values into the stylesheet.

Linking Columns: Columns flow from one to another and need to be linked together. There are several ways to link columns. The Column/Blocks Tab has a "Link To Column" field. Just type in the column number and [Apply Changes].

The Page Makeup option has a "Link Column" function. Use the Select Arrow to identify the first column in the order and then the next column in the order.

Within a Column's properties, you will see the "Column Links To" attribute. For example, the properties of Column 1 will show it linked to Column 2, and the properties of Column 2 will link to Column 3, etc.

Information about Blocks

Block components are pre-defined through the original DTD support files. Only existing Blocks found in the dropdown list can be Moved or Resized. New Blocks cannot be defined through OneForm Designer Plus tools for XML QRM use. It requires DTD programming changes.

There are six defined Blocks which can be Moved, Resized, and Applied to a Text Style. They are:

- FormFace (holds the Barcode)
- FormInstruction
- FormNumber
- OfficeAddress
- OMBNumber (holds OMB Number and Expiration Date)
- Year (holds Year and Form Title)

Note: The *QuestionGroup* Block should not be changed. It is a default Block used by the program. It's attributes will appear as 0, since it is dynamically set during placement of Question/Responses.

Specifying Block Positions

Create a dummy piece of Text to represent the data flowing in the block. Be sure it is in the correct position.. Move the appropriate Block component to sit Vertically Top-to-Top with the text.

Horizontally, position it correctly for the composition mode. Left edges touching for Flush Left, Centered horizontally on each other for Centered, and Right edges touching for Flush Right.

Each Block holds a corresponding Text component. Use the Text Styles Dialog to specify the text attributes for these components.

Block Related Text Styles

Each Block holds a corresponding Text component. Use the Text Styles Tab to "Apply" a new component or change the text attributes.

Block related text styles found in the *form2columns.xml* file:

- Form Title (Year) Text Style = HD1
- Office Address Text Style = HD2
- Form Instruction Text Style = HD3
- Form Number Text Style = P3
- OMB Number (OMB Expiration Date) Small Text Style = P5

Text Styles Tab

Figure 4-7: The Text Styles Panel

The panel called "Text Styles" is used to specify the typographic attributes of each text element in the XML content file.

Use the [Apply Changes] button to save any attribute changes.

The dialog is divided into three sections. The top allows you to pick a *Component* and the bottom two sections hold the *Paragraph* and *Enumeration Attribute* values.

Text styles are controlled by component tags P1, P2, etc. Selecting a tag will display the English Text Style Name and the XSL elements associated.

Paragraph Attributes control a paragraph's body text.

Font – select the font family from the dropdown list

Style – select the default style (Normal, Bold, Italic, Bold-Italic)

Horizontal Size – enter the horizontal point size of the text

Vertical Size – enter the vertical point size of the text

Leading – baseline-to-baseline leading between text lines

Lead Before – vertical space before the text paragraph

Justification – horizontal composition mode (Left, Center, Right, Justified)

Left Margin – the left edge of the text measuring from left of column

Right Margin – the width of the text measuring from left edge of column
(a value of 0 means to pick up the column width)

Layer (Color) – layer number (1-32)

Hyphenation – yes or no

Letterspacing – letterspacing value

Kerning – yes or no

Enumeration Attributes are the hanging numbers or letters which make up an outline style. Check the text component as being *Enumerated* and then assign the values.

Enum Font – select the font family from the dropdown list

Enum Style – select the default style (Normal, Bold, Italic, Bold-Italic)

Enum Hsize – enter the horizontal point size of the text

Enum Vsize – enter the vertical point size of the text

Enum Leading – baseline-to-baseline leading between text lines

Enum LMargin – the left edge of the text measuring from left of column

Enum RMargin – the width of the enumerated portion of text measuring from the left edge of the column

Enum Just – horizontal composition mode (Left, Center, Right, Justified)

Use the [Apply Changes] button to save any attribute changes.

The [New Text Style] button will copy the attributes from the selected component tag into a new tag. Edit the new tag component and associate it to an XSL element by highlighting one in the "Not Used" list and pushing it to the "Used For" list. Next, change the Text Style description.

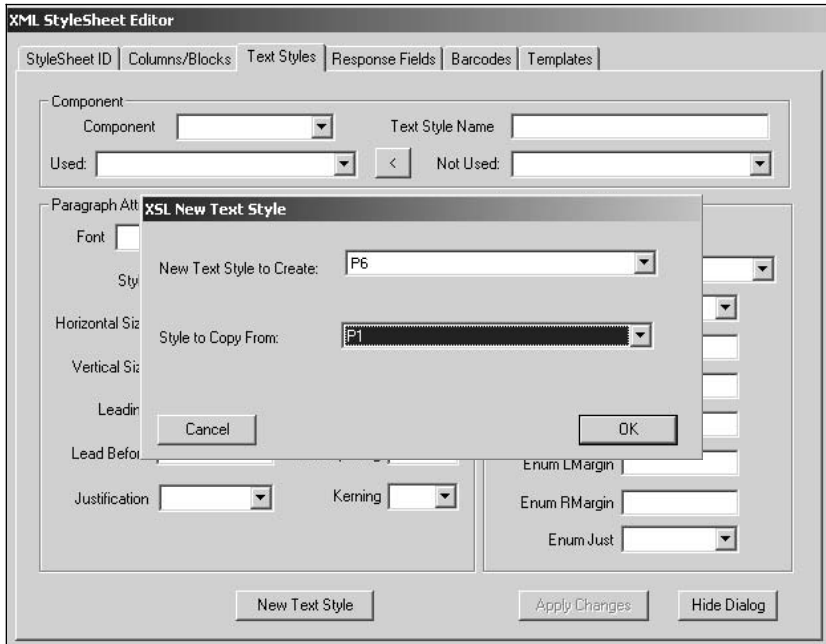


Figure 4-8: Creating a New Text Style Tag

The Response Fields Tab

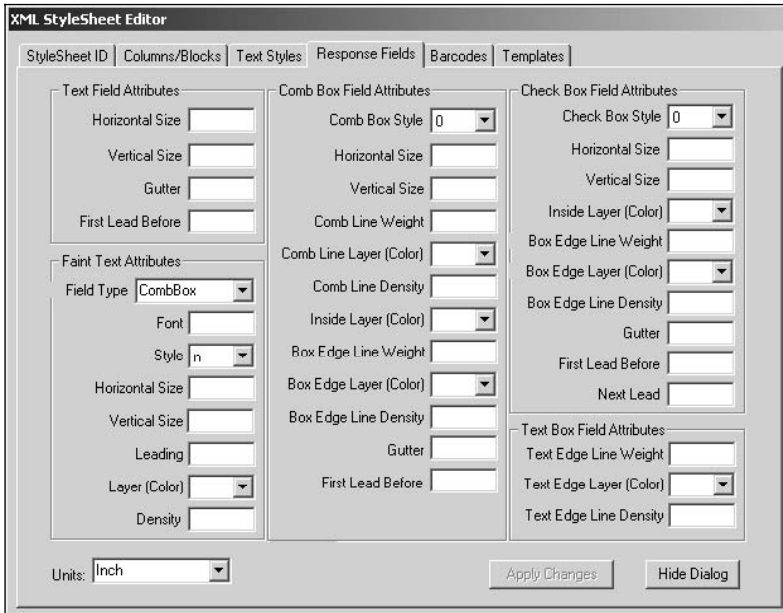


Figure 4-9: The Response Field Panel

The panel labeled "Response Fields" is used to display the specifications for the three types of response fields. If you make any changes to these attributes, select the [Apply Changes] button afterwards. It will respond with "Save Text Changes?" if you forget.

The dialog is broken into sections according to the response field type:

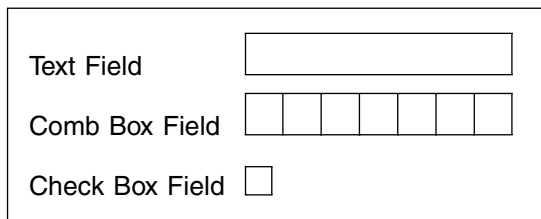


Figure 4-10: Response Field Types

The majority of the attributes for the fields are the same and are listed below:

Horizontal Size – width of each character in the field

Vertical Size – depth of the field

Gutter – horizontal spacing between fields

First Lead Before – vertical space between first field and its caption

Next Lead Before – vertical space between two fields

Comb Box Fields have additional attributes which describe the outside of the box versus the inside of the box, and then the vertical lines in-between. Each style of Comb Box can hold its own line weight, layer color, and density.

Comb Box Style – choose from the dropdown list between four styles (0-4)

Comb Line Weight – weight of vertical line between combs

Comb Layer (Color) – layer number (1-32) applied to vertical line

Comb Density – density (0-100) of color applied to comb line weight

Inside Layer (Color) – layer number (1-32) applied to the inside (normally white) area of the Comb Box

Box Edge Line Weight – line weight of the Comb Box Edge

Box Edge Layer (Color) – layer number (1-32) applied to Comb Edge

Box Edge Line Density – density (0-100) of color applied to Comb Edge

Check Box Fields also have four styles. Each style of Checkbox can hold its own line weight, layer color, and density. Their attributes are very similar to the Comb Box Fields in which they describe the outside of the box versus the inside of the box. (See Comb Box descriptions above).

Faint Text Attributes control the look of the shaded text placed inside of Combs or Check Boxes allowing you to show input examples. Density is the gray-scale percentage needed.

Field Type – choose from the dropdown list between Check or Combo Box

Font – enter your font mnemonic for the text you want to use

Style – enter the style from the dropdown list of the font you want to use

Leading – this is the amount of white space between text entries

Layer – layer number (1-32) applied to the faint text attributes

Density – density (0-100) of color applied to the faint text attributes

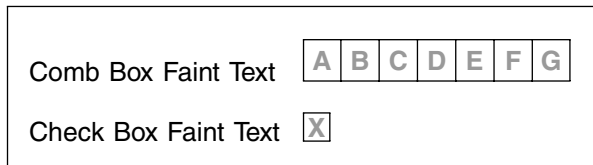
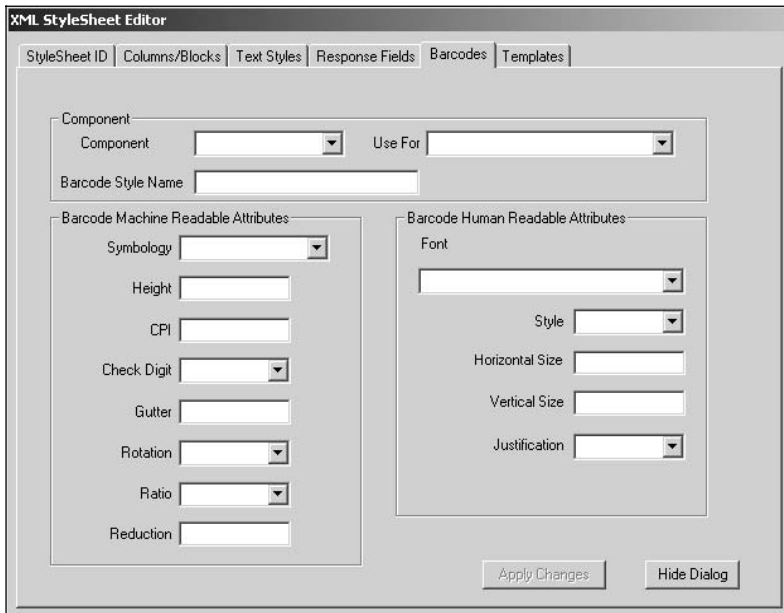


Figure 4-11: Faint Text Samples

The Barcodes Tab**Figure 4-12: The Barcodes Panel**

The panel called "Barcodes" is used to specify the barcode for the FormFace Block.

The *Barcode Machine Readable Attributes* reports the actual barcode type and its attributes.

Barcode Human Readable Attributes are the font attributes of the readable numbers supplied with the barcode.

The Templates Tab

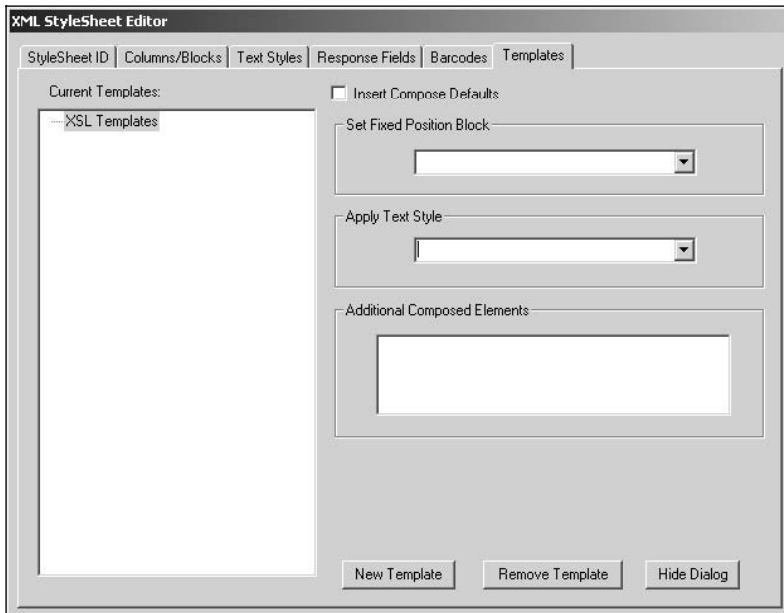


Figure 4-13: The Templates Panel

The "Templates" panel is used to insert special programming into a stylesheet. It reads the objects and lists their hierarchy.

How to Make a New Stylesheet

Follow these instructions to make a new stylesheet from an existing stylesheet. Start the ISE and click on the first "Browse" button to open the stylesheet file. We have used "form2columns.xml".

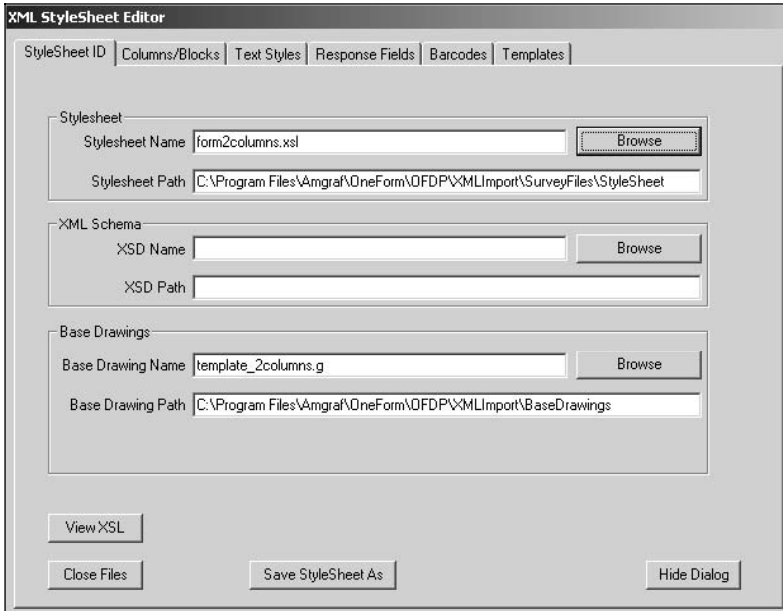


Figure 4-14: Opening an Existing Stylesheet

There is no need to open a Schema at this time.

Notice that the Base Drawing name is automatically extracted from the stylesheet and opened. There is no need to browse for a base drawing unless a new base drawing is desired.

Now click on the "Column/Blocks" tab.

[Add Columns to Base Page]

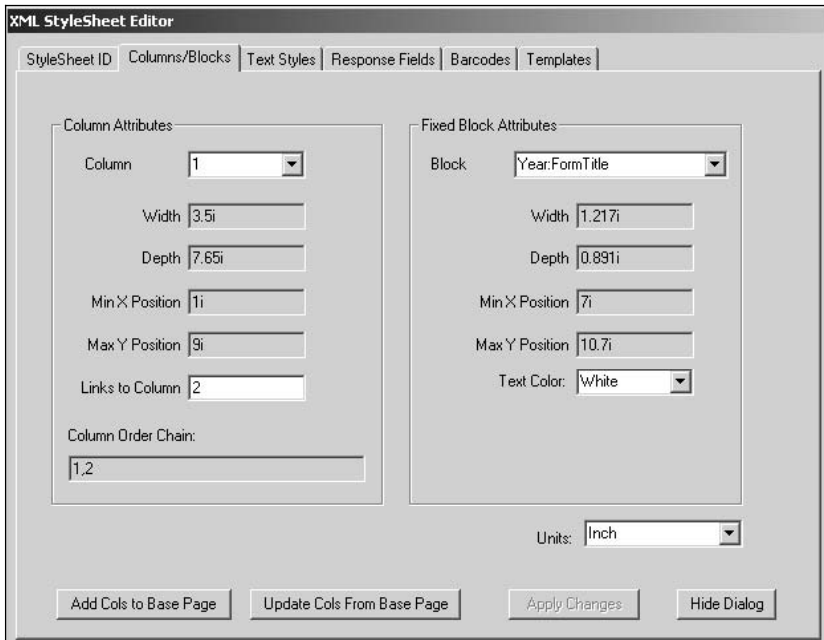


Figure 4-15: Examining the Column and Block Attributes

Click the dropdown list titled "Column Number" to see the attributes for each column definition in the stylesheet. Click the dropdown list titled "Block Name" to see the fixed blocks defined in the stylesheet.

To see a visual of these columns and blocks, click the button labeled [Add Cols to Base Page]. This will overlay the column and block boundaries on the base page template.

To see the base page more clearly, move the ISE dialog panel out of the window as shown.

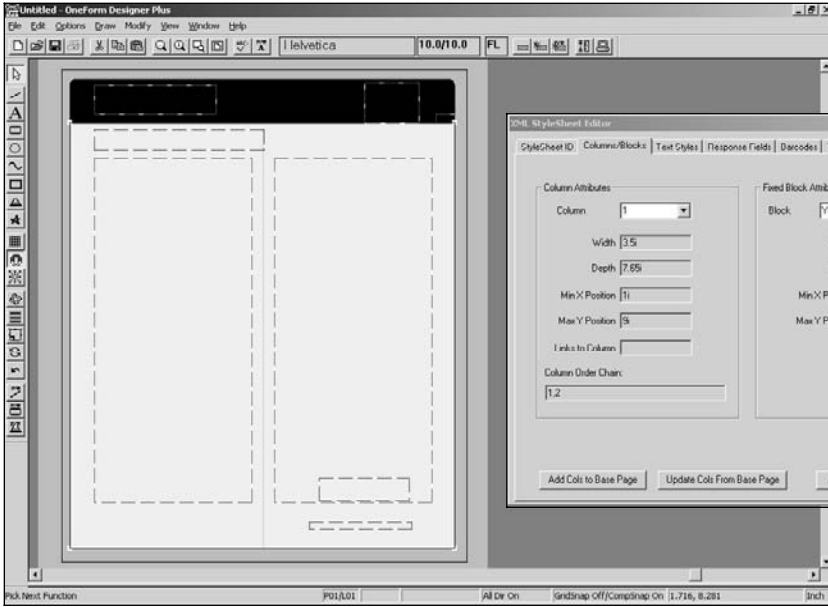


Figure 4-16: Column and Block Boundaries are Overlaid on the Base Page

Column boundaries are displayed as purple dashed lines. Block boundaries are displayed as red dashed lines.

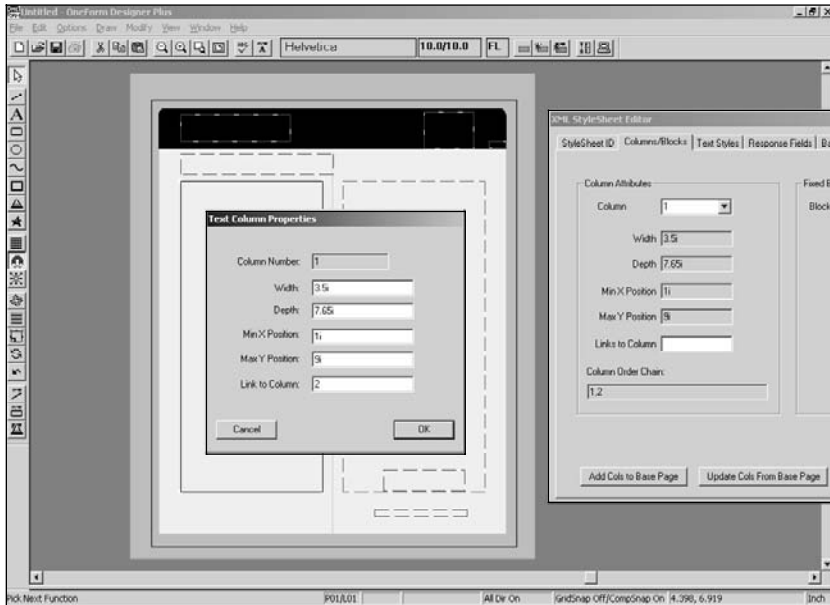


Figure 4-17: Examining Data Values for a Column

Zoom up so that the first page is more clearly shown.

To identify the name/number of a block or column, use the Select Arrow tool and click on a boundary, then right-button click to see the "Column Properties".

Do this now with the second column on the base drawing. Note that the selected column is displayed as a solid black rectangle.

Modifying Columns

Now right button click on the selected column and choose the "Modify Column Size" function. Click on the bottom of the column and move it up to make its depth shorter.

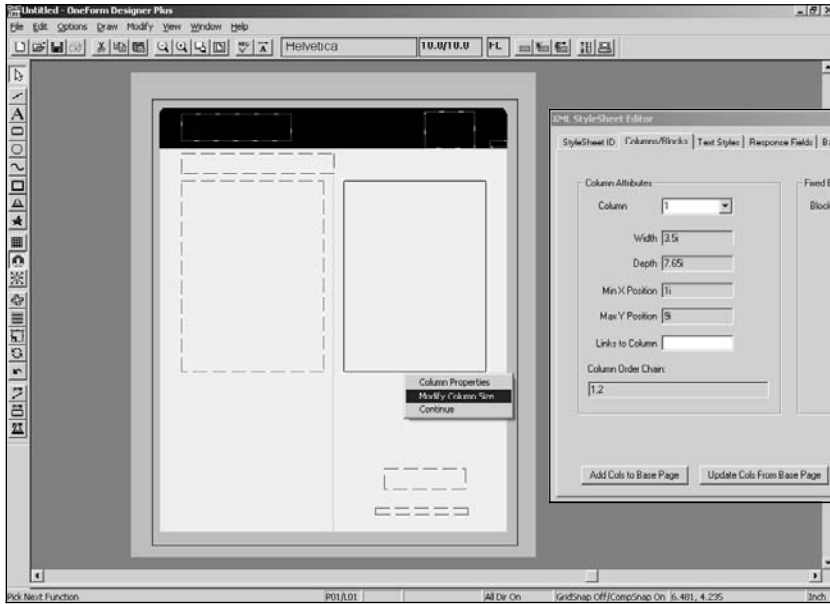


Figure 4-18: Using the Modify Column Size Function

Modify the depth of each column and select the [Update Cols From Base Page] button to pull the changes back into the stylesheet XSL file.

Adding a New Column

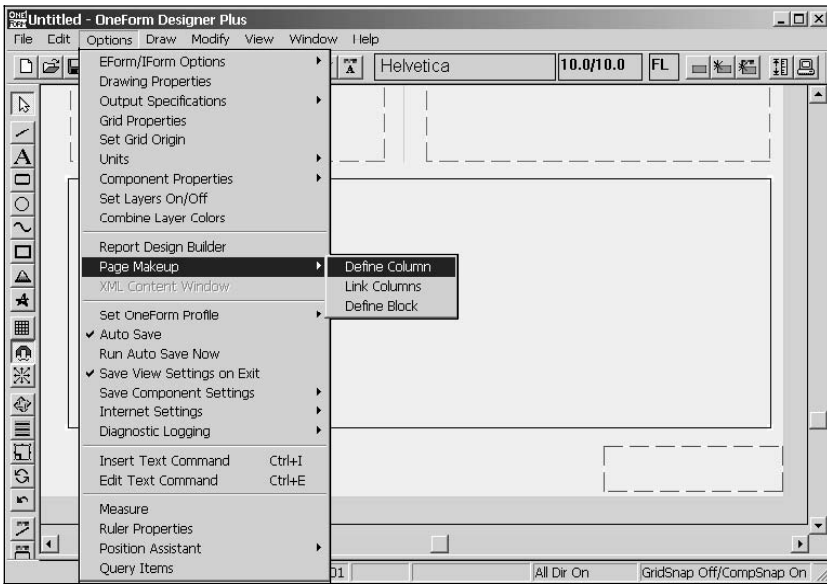


Figure 4-19: Page Makeup Option

A new column can be drawn on the base drawing through the Options, Page Makeup, Define Column. Draw a new column across the bottom of the page. It will automatically name the new column with the next available number.

Using the Select Arrow, you can identify it and modify its size or look at its properties.

The Link Column option allows you to apply the flow order. Click the Select Arrow, then go to the Options, Page Makeup, Link Columns function. Use the Select Arrow to pick Column 2 and then Column 3.

Be sure to click the ISE button labeled [Update Cols from Base Page] to store the new column values into the stylesheet. The "Column Order Chain" will show 1,2,3.

Now lets go to the "Text Style" Tab.

Change the P11 Text Style

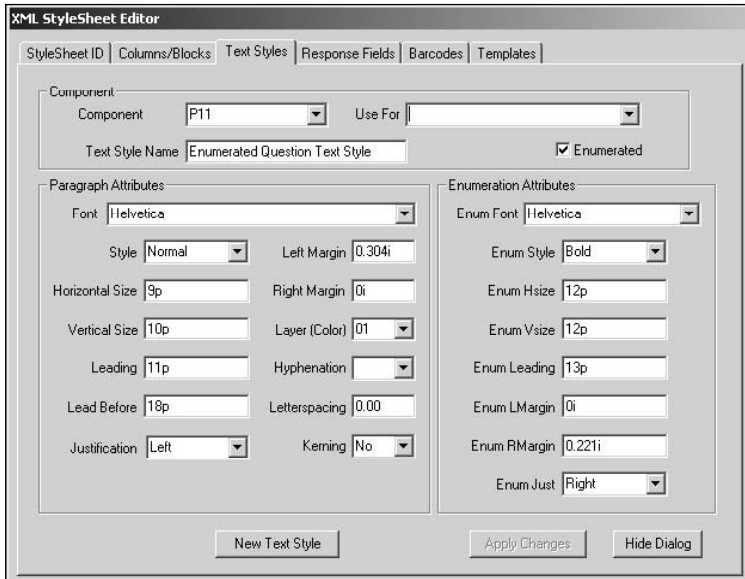


Figure 4-20: The Values for Text Styles are Displayed

The stylesheet has text style attributes for each kind of component used in a survey form. Click on the "Component Name" dropdown list and select P11 to see the typographical values for Enumerated Questions. In our exercise, we are going to change the attributes for this component as shown on the next page.

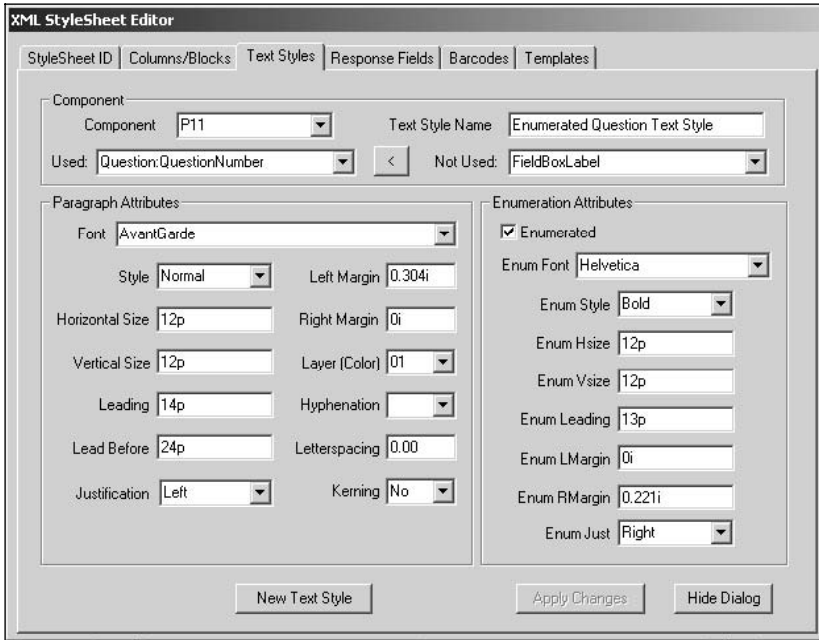


Figure 4-21: Changing Text Styles

As you can see above, we have changed the font from Helvetica to AvantGarde. We also changed the horizontal and vertical sizes to 12p, and the leading value to 14p.

Click the [Apply Changes] button to save the new values.

Now click the tab labeled "Response Fields" to continue.

Change the Comb Box Field

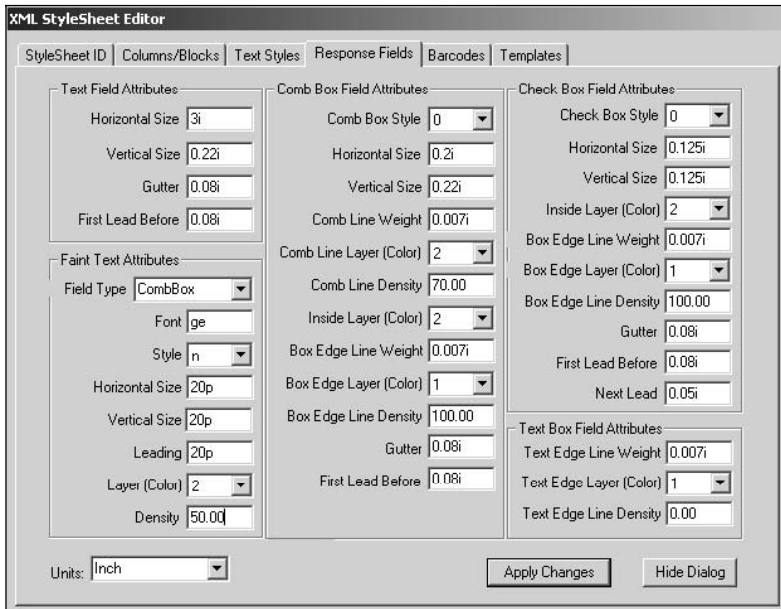


Figure 4-22: The Values for Response Fields are Displayed

Here we can see the attributes within the stylesheet for the response fields. Change the Comb Box Field "Box Edge Layer (Color)" from a "1" to a "2". This will make the edge the same as our background instead of black. Click the [Apply Changes] button to save the new value.

Save the New Stylesheet

Save the stylesheet by a new name by clicking the tab labeled "Stylesheet ID" and then clicking the button labeled "Save Stylesheet As" as shown and giving it the new name "form3columns.xml".

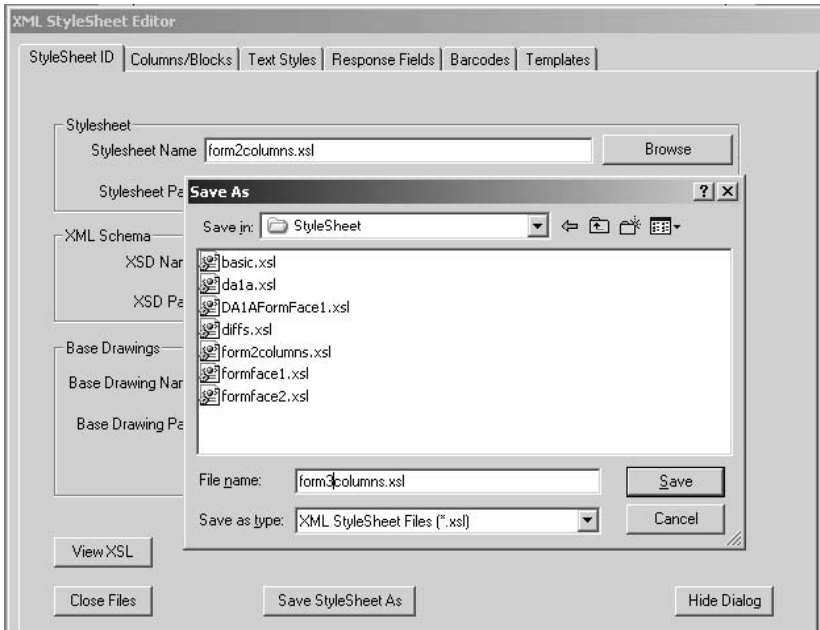


Figure 4-23: Saving a New Stylesheet

Verify that the Changes are in the New Stylesheet

A simple way to verify that the stylesheet changes were applied is to import an XML content file using the XML Workflow and specify the new stylesheet name during the import process.

Another way is to look at the XSL file to compare the original and new stylesheets. Following are screenshots showing the stylesheet source files before and after the changes described.

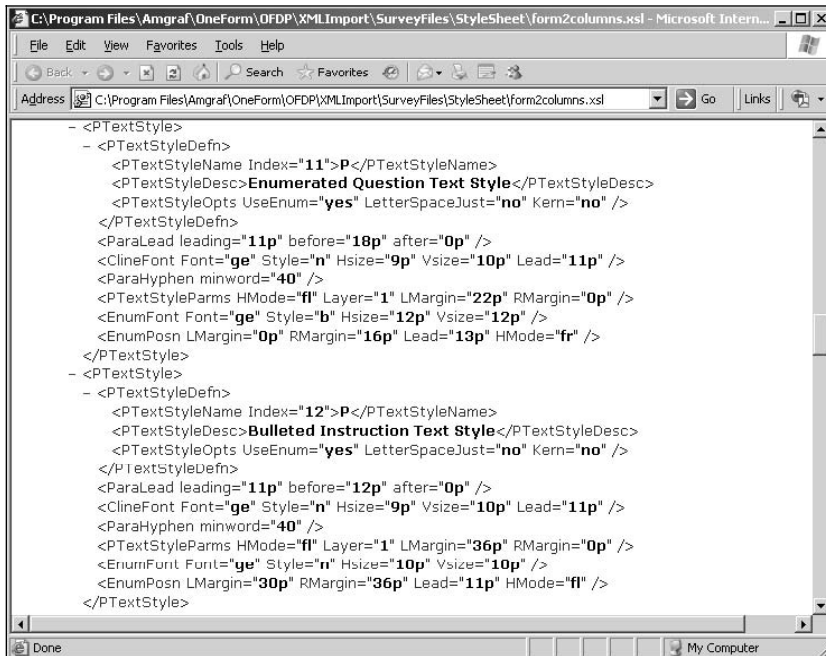


Figure 4-24: The Original P11 Text Style Values

The file was opened in Windows Explorer, but you could use an XML editor such as XMLSpy or PSPad.

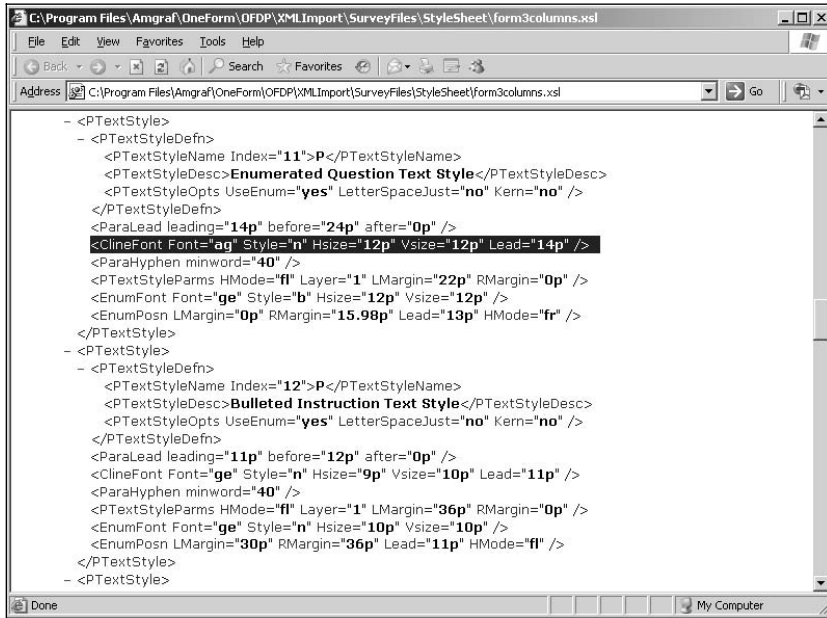


Figure 4-25: The New P11 Text Style Values

The highlighted area in the above illustration shows the typeface changes made using the ISE panels.

Example of the XSL File

Following is an example of the *Two Column Form XSL* file. The stylesheets can be maintained and modified by using a programmer's XML Editor. This is not only inconvenient, but is also risky because it is possible to introduce XML errors. As you can see the XSL file is a fairly complicated file, but can be read with practice.

The ISE enhancement to OFDP simplifies the development of new XSL files and makes the appropriate changes within the XSL file through easy to use dialogs.

The bolded items within the file are referenced below:

PageDefn: Describes the page size.

OpenPage: This is the template graphic ".g" file to use.

ColumnDefn: Describes each column sizes and their starting position.

UseDefn: Lets you decide which Columns are to be used.

PTextStyle: Each Paragraph style can have its own attributes, such as font, pointsize, leading, layer, etc.

form2columns.xsl

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="xml"/>
  <xsl:template match="/">
    <xsl:text disable-output-escaping="yes">&lt;?xml
      version="1.0" encoding="iso-8859-1"
      standalone="no"?&gt;</xsl:text>
    <xsl:text disable-output-escaping="yes">&lt;!DOCTYPE
      OFDPBOCSurveyPages SYSTEM "c:/Program
      Files/Amgraf/OneForm/OFDP/XMLImport/DTD/
      OFDPBOCSurveyPages.dtd"&gt;</xsl:text>
    <xsl:text>
    </xsl:text>
    <xsl:apply-templates select="@*|node()"/>
  </xsl:template>
  <!--Stylesheet Processor for Amgraf XML code-->
  <xsl:template match="OFDPBOCSurvey">
    <!--NOTE: xsl:text pair causes a Line Break
      in the output file-->
    <xsl:text>
    </xsl:text>
    <!--Refer to New DTD -->
    <OFDPBOCSurveyPages>
      <PageDefn Width="10i" Depth="12i" Lm="0i"
        Rm="0i" Duplex="no" OutputPageNumber="1"
        OpenPage="c:/Program Files/Amgraf/OneForm/
        OFDP/XMLImport/BaseDrawings/template_2columns.g"/>
      <xsl:apply-templates select="@*|node()"/>
    </OFDPBOCSurveyPages>
  </xsl:template>
</xsl:stylesheet>
```

```
</xsl:template>
<!--Now Process Columns of Form -->
<xsl:template match="QRM">
  <QRM GUID="{@GUID}">
    <ResponseFieldDefaults>
      <ComboBoxDefaults Style="0" Hsize=".20i" Vsize=".22i"
        CombLineWeight=".5p" CombLineLayer="1" CombLineDensity="70"
        EdgeLineWeight=".5p" EdgeLineLayer="1" EdgeLineDensity="70"
        CombInsideLayer="2" Gutter=".08i"
        FirstLB=".08i"/>
      <CheckBoxDefaults Style="0" Hsize=".125i" Vsize=".125i"
        BoxLineWeight=".5p" BoxLineLayer="1" BoxLineDensity="100"
        BoxInsideLayer="2" Gutter=".08i" FirstLB=".08i"
        Gutter=".08i" FirstLB=".08i" NextLB=".05i"/>
      <TextBoxDefaults Hsize="3i" Vsize=".22i"
        Gutter=".08i" FirstLB=".08i"/>
    </ResponseFieldDefaults>
    <PTextStyle>
      <PTextStyleDefn>
        <PTextStyleName Index="1">P</PTextStyleName>
        <PTextStyleDesc>Standard Caption Text
          Style</PTextStyleDesc>
        <PTextStyleOpts UseEnum="no" LetterSpaceJust="no"
          Kern="no"/>
      </PTextStyleDefn>
      <ParaLead leading="10p" before="12p" after="0p"/>
      <ClineFont Font="ge" Style="n" Hsize="9p"
        Vsize="9p" Lead="10p"/>
      <ParaHyphen minword="40"/>
      <PTextStyleParms HMode="fl" Layer="1" LMargin="0"
        RMargin="0"/>
    </PTextStyle>
    <PTextStyle>
      <PTextStyleDefn>
        <PTextStyleName Index="2">P</PTextStyleName>
        <PTextStyleDesc>Condensed Caption Text
          Style</PTextStyleDesc>
        <PTextStyleOpts UseEnum="no" LetterSpaceJust="no"
          Kern="no"/>
      </PTextStyleDefn>
      <ParaLead leading="10p" before="12p" after="0p"/>
      <ClineFont Font="ge" Style="n" Hsize="8p"
        Vsize="9p" Lead="10p"/>
      <ParaHyphen minword="40"/>
      <PTextStyleParms HMode="fl" Layer="1" LMargin="0"
        RMargin="0"/>
    </PTextStyle>
    <PTextStyle>
      <PTextStyleDefn>
        <PTextStyleName Index="3">P</PTextStyleName>
        <PTextStyleDesc>Bold Instruction Text
          Style</PTextStyleDesc>
        <PTextStyleOpts UseEnum="no" LetterSpaceJust="no"
          Kern="no"/>
      </PTextStyleDefn>
```

```
<ParaLead leading="10.5p" before="18p" after="0p"/>
<ClineFont Font="ge" Style="n" Hsize="9.5p"
  Vsize="9.5p" Lead="10.5p"/>
<ParaHyphen minword="40"/>
<PTextStyleParms HMode="fl" Layer="1" LMargin="0"
  RMargin="0"/>
</PTextStyle>
<PTextStyle>
  <PTextStyleDefn>
    <PTextStyleName Index="4">P</PTextStyleName>
    <PTextStyleDesc>Bold Italic Instruction
      Text Style</PTextStyleDesc>
    <PTextStyleOpts UseEnum="no" LetterSpaceJust="no"
      Kern="no"/>
  </PTextStyleDefn>
  <ParaLead leading="10.5p" before="18p" after="0p"/>
  <ClineFont Font="ge" Style="n" Hsize="9.5p"
    Vsize="9.5p" Lead="10.5p"/>
  <ParaHyphen minword="40"/>
  <PTextStyleParms HMode="fl" Layer="1" LMargin="0"
    RMargin="0"/>
</PTextStyle>
<PTextStyle>
  <PTextStyleDefn>
    <PTextStyleName Index="5">P</PTextStyleName>
    <PTextStyleDesc>OMB Number Small Text
      Style</PTextStyleDesc>
    <PTextStyleOpts UseEnum="no" LetterSpaceJust="no"
      Kern="no"/>
  </PTextStyleDefn>
  <ParaLead leading="7p" before="12p" after="0p"/>
  <ClineFont Font="ge" Style="n" Hsize="6p"
    Vsize="6p" Lead="7p"/>
  <ParaHyphen minword="40"/>
  <PTextStyleParms HMode="fl" Layer="1" LMargin="0"
    RMargin="0"/>
</PTextStyle>
<PTextStyle>
  <PTextStyleDefn>
    <PTextStyleName Index="11">P</PTextStyleName>
    <PTextStyleDesc>Enumerated Question Text
      Style</PTextStyleDesc>
    <PTextStyleOpts UseEnum="yes"/>
  </PTextStyleDefn>
  <ParaLead leading="11p" before="18p" after="0p"/>
  <ClineFont Font="ge" Style="n" Hsize="9p"
    Vsize="10p" Lead="11p"/>
  <ParaHyphen minword="40"/>
  <PTextStyleParms HMode="fl" Layer="1" LMargin="22p"
    RMargin="0"/>
  <EnumFont Font="ge" Style="n" Hsize="12p"
    Vsize="12p"/>
  <EnumPosn LMargin="0p" RMargin="16p" Lead="13p"
    HMode="fr"/>
</PTextStyle>
```

```
<PTextStyle>
  <PTextStyleDefn>
    <PTextStyleName Index="12">P</PTextStyleName>
    <PTextStyleDesc>Bulleted Instruction Text
      Style</PTextStyleDesc>
    <PTextStyleOpts UseEnum="yes"/>
  </PTextStyleDefn>
  <ParaLead leading="11p" before="12p" after="0p"/>
  <ClineFont Font="ge" Style="n" Hsize="9p"
    Vsize="10p" Lead="11p"/>
  <ParaHyphen minword="40"/>
  <PTextStyleParms HMode="fl" Layer="1" LMargin="36p"
    RMargin="0"/>
  <EnumFont Font="ge" Style="n" Hsize="10p"
    Vsize="10p"/>
  <EnumPosn LMargin="30p" RMargin="36p" Lead="11p"
    HMode="fl"/>
</PTextStyle>
<PTextStyle>
  <PTextStyleDefn>
    <PTextStyleName Index="1">HD</PTextStyleName>
    <PTextStyleDesc>Form Title Text
      Style</PTextStyleDesc>
    <PTextStyleOpts UseEnum="no"/>
  </PTextStyleDefn>
  <ParaLead leading="22p" before="36p" after="0p"/>
  <ClineFont Font="ge" Style="n" Hsize="20p"
    Vsize="20p" Lead="22p"/>
  <ParaHyphen minword="40"/>
  <PTextStyleParms HMode="ce" Layer="1" LMargin="0"
    RMargin="88p"/>
</PTextStyle>
<PTextStyle>
  <PTextStyleDefn>
    <PTextStyleName Index="2">HD</PTextStyleName>
    <PTextStyleDesc>Office Address Text
      Style</PTextStyleDesc>
    <PTextStyleOpts UseEnum="no"/>
  </PTextStyleDefn>
  <ParaLead leading="14p" before="14p" after="0p"/>
  <ClineFont Font="ge" Style="b" Hsize="14p"
    Vsize="14p" Lead="14p"/>
  <ParaHyphen minword="40"/>
  <PTextStyleParms HMode="fl" Layer="1" LMargin="0"
    RMargin="240p"/>
</PTextStyle>
<PTextStyle>
  <PTextStyleDefn>
    <PTextStyleName Index="3">HD</PTextStyleName>
    <PTextStyleDesc>Form Instruction Text
      Style</PTextStyleDesc>
    <PTextStyleOpts UseEnum="no"/>
  </PTextStyleDefn>
  <ParaLead leading="8p" before="8p" after="0p"/>
  <ClineFont Font="ge" Style="n" Hsize="8p"
    Vsize="8p" Lead="8p"/>
  <ParaHyphen minword="40"/>
  <PTextStyleParms HMode="fl" Layer="1" LMargin="0"
    RMargin="26p"/>
</PTextStyle>
```

```

<BarcodeStyle>
  <BarcodeStyleDefn>
    <BarcodeStyleName Index="1">B</BarcodeStyleName>
    <BarcodeStyleDesc>Interleaved 2 of 5 Barcode
    </BarcodeStyleDesc>
    <BarcodeSymbology Type="INT25"/>
    <BarcodeMRStyleOpts Height=".5i" CPI="7.5"
      CheckDigit="no" Gutter="10p"
      Rotation="0" Ratio="2.5" Reduction=".001i"/>
    <BarcodeHRStyleOpts Location="Left" Font="ge"
      Style="n" Hsize="7p"
      Vsize="7p" HMode="fr"/>
    </BarcodeStyleDefn>
  </BarcodeStyle>
<ColumnDefn Number="1" Width="3.532i" Depth="7.65i"
  Minx="1i" Maxy="9i"/>
<ColumnDefn Number="2" Width="4.181i" Depth="7.65i"
  Minx="5i" Maxy="9i"/>
  <TextStyleAttach Tag="Question:QuestionText"
    Use="P11" Section="Body"/>
  <TextStyleAttach Tag="Question:QuestionNumber"
    Use="P11" Section="Enum"/>
  <TextStyleAttach Tag="Question:QuestionInstruction"
    Use="P4" Section="Body"/>
  <TextStyleAttach Tag="Question:InstructionItem:
    InstructionItemText"
    Use="P12" Section="Body"/>
  <TextStyleAttach Tag="Question:InstructionItem:
    InstructionItemText:Number"
    Use="P12" Section="Enum"/>
  <TextStyleAttach Tag="FieldBoxLabel" Use="P1"
    Section="Body"/>
  <TextStyleAttach Tag="GroupInstruction" Use="P3"
    Section="Body"/>
  <Column ForceNewPage="no" UseDefn="1,2" Position="fixed">
    <xsl:text>
    </xsl:text>
    <xsl:apply-templates select="@*|node()"/>
  </Column>
</QRM>
</xsl:template>
<xsl:template match="Year">
  <Block PositionType="fixed" PageX="7i" PageY="10.7i"
    Width="1.217i" Depth="1.071i">
    <Textstyle Use="HD1" color="White"/>
    <FormTitle>
      <xsl:apply-templates select="@*|node()"/>
      <xsl:text>
      </xsl:text>
      <!--Concatenate the Year and Form Title into 1 tag-->
      <xsl:value-of select="../FormTitle"/>
    </FormTitle>
  </Block>
</xsl:template>

```

```
<!--Now Discard FormTitle -->
<xsl:template match="FormTitle"/>
<!--Process Margin Text outside of Columns -->
<xsl:template match="FormNumber">
  <Block PositionType="fixed" PageX="8.6i" PageY="10i"
    Width="0.996i" Depth="0.18i">
    <Textstyle Use="HD3" color="White"/>
    <FormNumber>
      <xsl:apply-templates select="@*|node()"/>
    </FormNumber>
  </Block>
</xsl:template>
<!-- Extract OMB Number and Expiration Date-->
<xsl:template match="OMBNumber">
  <Block PositionType="fixed" PageX="17.054i"
    PageY="0.749i" Width="2.279i" Depth="0.07">
    <Textstyle Use="P5" color="Black"/>
    <Number>
      <xsl:apply-templates select="@*|node()"/>
      <xsl:text>
        </xsl:text>
      <!-- Concatenate the OMB Number and Expiration
        Date into 1 tag-->
      <xsl:value-of select="../OMBExpirationDate"/>
    </Number>
  </Block>
</xsl:template>
<!-- Now Discard Expire Date -->
<xsl:template match="OMBExpirationDate"/>
<!-- Insert Barcode Human and Machine Readable
Codes -->
<xsl:template match="FormFace">
  <Block PositionType="fixed" PageX="8.115i"
    PageY="0.582i" Width=".4i" Depth="0.07i">
    <Textstyle Use="P5" color="Black"/>
    <BarcodeHR>
      <xsl:value-of select="@Barcode"/>
    </BarcodeHR>
  </Block>
  <Block PositionType="fixed" PageX="7i" PageY="0.88i"
    Width="2.0i" Depth="0.5i">
    <Textstyle Use="B1" color="Black"/>
    <BarcodeMR>
      <xsl:value-of select="@Barcode"/>
    </BarcodeMR>
  </Block>
</xsl:template>
```



```
<!-- Extract Office Address -->
<xsl:template match="OfficeAddress">
  <Block PositionType="fixed" PageX="1i" PageY="10.5i"
    Width="4.0i" Depth="0.5i">
    <Textstyle Use="HD2" color="White"/>
    <OfficeAddress>
      <xsl:apply-templates select="@*|node()"/>
    </OfficeAddress>
  </Block>
</xsl:template>
<!-- Extract Form Instructions -->
<xsl:template match="FormInstruction">
  <Block PositionType="fixed" PageX="1i" PageY="9.7ii"
    Width="5.546i" Depth="0.474i">
    <Textstyle Use="P4" color="Black"/>
    <FormInstruction>
      <xsl:apply-templates select="@*|node()"/>
    </FormInstruction>
  </Block>
</xsl:template>
  Position="fixed"> -->
<xsl:template match="QuestionGroup">
  <Block PositionType="run">
    <Textstyle Group="P11" color="Black"/>
    <QuestionGroup>
      <xsl:apply-templates select="@*|node()"/>
    </QuestionGroup>
  </Block>
</xsl:template>
<!-- Copy all the remaining source tree nodes -->
<xsl:template match="@*|node()">
  <xsl:copy>
    <xsl:apply-templates select="@*|node()"/>
  </xsl:copy>
</xsl:template>
</xsl:stylesheet>
```


CHAPTER 5

Working with the Graphic Files

There are two options specifically designed to work with GUID objects on the composed graphic files: View and Move GUID Objects.

Viewing GUID Components

Under the View menu is a toggle for viewing GUID objects. The default is to view all GUID objects on the form.

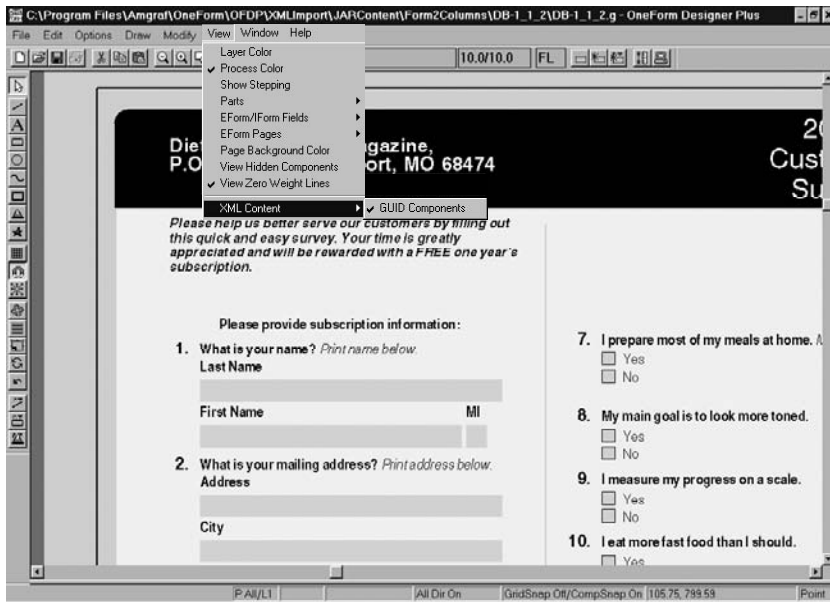


Figure 5-1: Viewing GUID Components

If you want to look **ONLY** at the base template or any other static data which has been placed on the form, toggle off the GUID view.

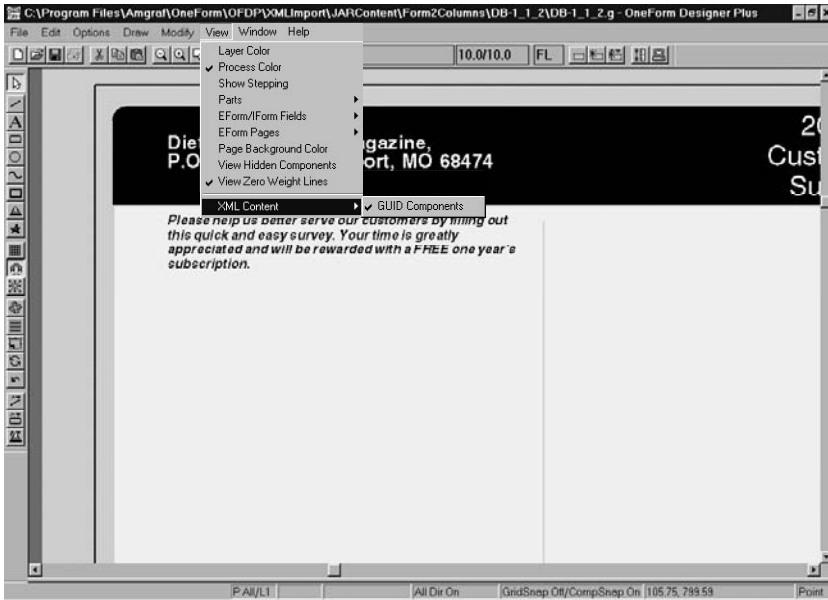


Figure 5-2: Turning Off the GUID View

No Fillable Fields View

There is also a View option for turning off the response fillable fields. **View, Eform/Iform Fields, No Fields** will turn off the display of fillable fields and let you see the conventional print version of the form.

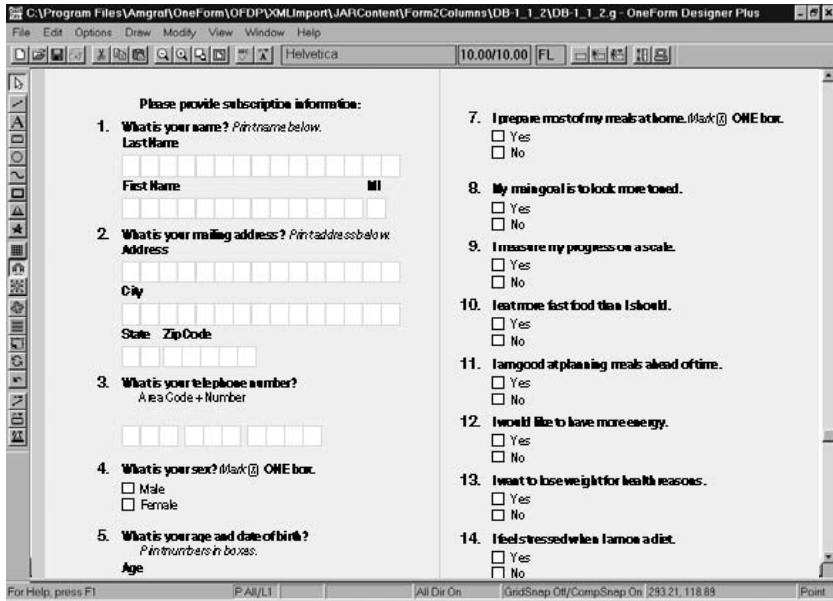


Figure 5-3: Turning Off the Fillable Fields View

Use the View, **Eform/Iform Fields, Normal Fields** option to turn on the display of fillable fields.

Moving GUID Items

There are times when GUID components need to be moved, usually to insert a new question. You will find that GUID Response components are combined together; the label, fillable field, and underlying comb boxes are linked as one group.

The Move tool menu has an option for moving GUID components individually. This option will unlink the components temporarily for that one move step.

For example, you want to keep the comb boxes and the fillable field components together during the Move process. Use the "Group Inside" function under Edit to identify the items which need to be kept together. Use this Grouping function, before going to the Move's "Allow GUID Item Move" option. This will allow the grouped items to stay together, but unlink themselves from the label text components.

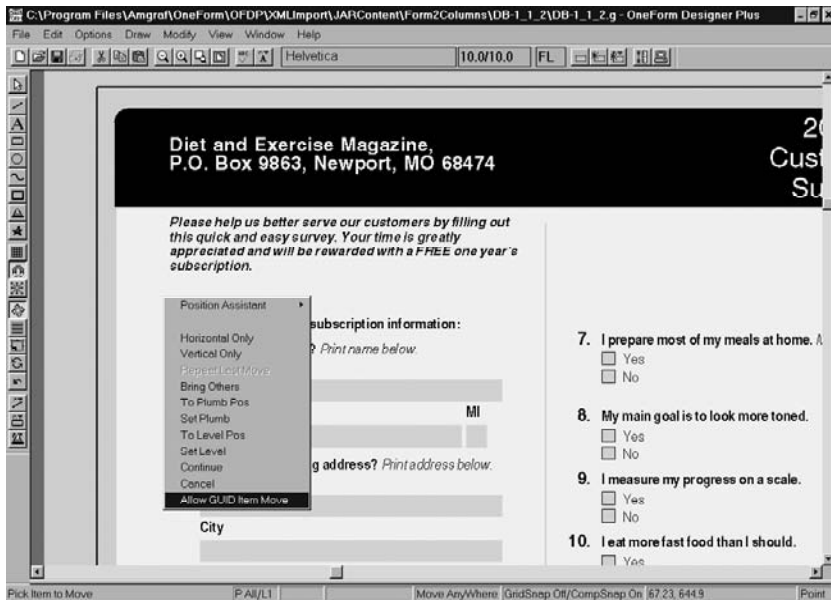


Figure 5-4: Move, Allow GUID Item Move

CHAPTER 6

Composition Model Option

Frequently there are sections on survey forms that have a great deal of graphical similarity, especially those questions that are repeated for multiple persons residing in the same household. To streamline this operation the Composition Modeling option was developed.

Before the introduction of XML Workflow, the forms composers could easily layout the section for one person and then copy it multiple times for the other persons, and with minimal text editing the form layout would then be completed.

Within the XML Workflow, copying of composed sections is not allowed because response fields are now part of the layout. The response fields, and almost all other textual components have unique names and GUIDs that cannot be copied. The result is that each "person section" in the form has to be individually composed through the XML Import software.

The objective of Composition Modeling is to reduce the overall composition time required to create forms with repeating questions. As shown in our example form, each column is nearly identical.

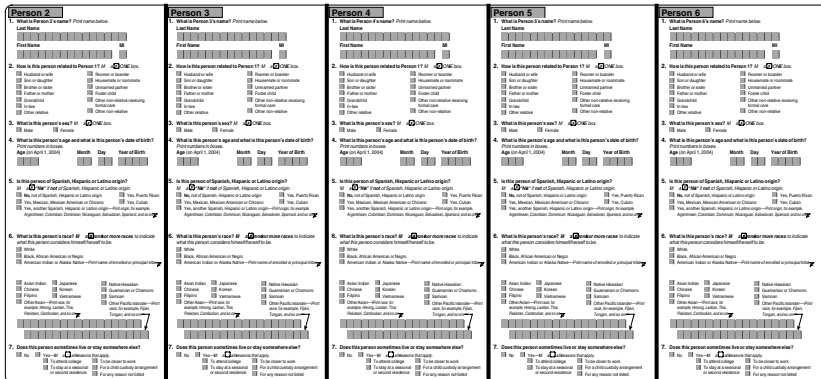


Figure 6-1: Sample Survey Form with Repeating Questions

Each of the columns has the same 7 questions with their related responses. The question/response content for Column 1 is carefully arranged of the OFDP operator into the desired layout one time, then by activating the Composition Model option within a New Column, the raw content will pick up the layout for previous questions of the same number and import them applying the graphical layouts stored.

The Composition Model function duplicates the graphical similarity and position of question/responses when that question is repeated in another column. The composition model reference is by question text number. For example: when Question 1 is composed for the first time its graphical layout

is stored automatically in the file as "Composition Model Q1". When Question 1 appears a second time, if Composition Model is invoked, it will automatically compose the second Question 1 using the graphical layout of the first appearance of Question 1.

Samples of Usage for Composition Models

Looking at the differences between the standard Imported XML Content and the Composed Questions/Responses you will see that placement of the text and fields are different. Composition Model allows for the graphical positioning to be reused on repeated questions within the same form.

Imported XML Content	Composed Questions/Responses
<p>1. What is Person 2's name? <i>Print name below.</i></p> <p>Last Name <input type="text"/></p> <p>First Name <input type="text"/></p> <p>MI <input type="text"/></p>	<p>1. What is Person 2's name? <i>Print name below.</i></p> <p>Last Name <input type="text"/></p> <p>First Name <input type="text"/> MI <input type="text"/></p> <p>Composition Model: Q1</p>
<p>2. How is this person related to Person 1? <i>Mark <input checked="" type="checkbox"/> ONE box.</i></p> <p><input type="checkbox"/> Husband or wife <input type="checkbox"/> Son or daughter <input type="checkbox"/> Brother or sister <input type="checkbox"/> Father or mother <input type="checkbox"/> Grandchild <input type="checkbox"/> In-law <input type="checkbox"/> Other relative <input type="checkbox"/> Roomer or boarder <input type="checkbox"/> Housemate or roommate <input type="checkbox"/> Unmarried partner <input type="checkbox"/> Foster child <input type="checkbox"/> Other non-relative under formal care <input type="checkbox"/> Other non-relative</p>	<p>2. How is this person related to Person 1? <i>Mark <input checked="" type="checkbox"/> ONE box.</i></p> <p><input type="checkbox"/> Husband or wife <input type="checkbox"/> Son or daughter <input type="checkbox"/> Brother or sister <input type="checkbox"/> Father or mother <input type="checkbox"/> Grandchild <input type="checkbox"/> In-law <input type="checkbox"/> Other relative <input type="checkbox"/> Roomer or boarder <input type="checkbox"/> Housemate or roommate <input type="checkbox"/> Unmarried partner <input type="checkbox"/> Foster child <input type="checkbox"/> Other non-relative under formal care <input type="checkbox"/> Other non-relative</p> <p>Composition Model: Q2</p>
<p>3. What is this person's sex? <i>Mark <input checked="" type="checkbox"/> ONE box.</i></p> <p><input type="checkbox"/> Male <input type="checkbox"/> Female</p>	<p>3. What is this person's sex? <i>Mark <input checked="" type="checkbox"/> ONE box.</i></p> <p><input type="checkbox"/> Male <input type="checkbox"/> Female</p> <p>Composition Model: Q3</p>
<p>4. What is this person's age and what is this person's date of birth? <i>Print numbers in boxes.</i></p> <p>Age (on April 1, 2004) <input type="text"/></p> <p>Month <input type="text"/></p> <p>Day <input type="text"/></p> <p>Year of Birth <input type="text"/></p>	<p>4. What is this person's age and what is this person's date of birth? <i>Print numbers in boxes.</i></p> <p>Age (on April 1, 2004) Month Day Year of Birth <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p> <p>Composition Model: Q4</p>
<p>5. Is this person of Spanish, Hispanic or Latino origin? <i>Mark <input checked="" type="checkbox"/> "No" if not of Spanish, Hispanic or Latino origin.</i></p> <p><input type="checkbox"/> No, not of Spanish, Hispanic or Latino origin <input type="checkbox"/> Yes, Mexican, Mexican American or Chicano <input type="checkbox"/> Yes, another Spanish, Hispanic or Latino origin — <i>Print origin, for example, Argentinean, Colombian, Dominican, Nicaraguan, Salvadoran, Spaniard and so on.</i></p> <p><input type="text"/></p> <p><input type="checkbox"/> Yes, Puerto Rican <input type="checkbox"/> Yes, Cuban</p>	<p>5. Is this person of Spanish, Hispanic or Latino origin? <i>Mark <input checked="" type="checkbox"/> "No" if not of Spanish, Hispanic or Latino origin.</i></p> <p><input type="checkbox"/> No, not of Spanish, Hispanic or Latino origin <input type="checkbox"/> Yes, Puerto Rican <input type="checkbox"/> Yes, Mexican, Mexican American or Chicano <input type="checkbox"/> Yes, Cuban <input type="checkbox"/> Yes, another Spanish, Hispanic or Latino origin — <i>Print origin, for example, Argentinean, Colombian, Dominican, Nicaraguan, Salvadoran, Spaniard and so on.</i></p> <p><input type="text"/></p> <p>Composition Model: Q5</p>

Figure 6-2: Composed Question Models can be Reused on the Same Form

Using Composition Modeling

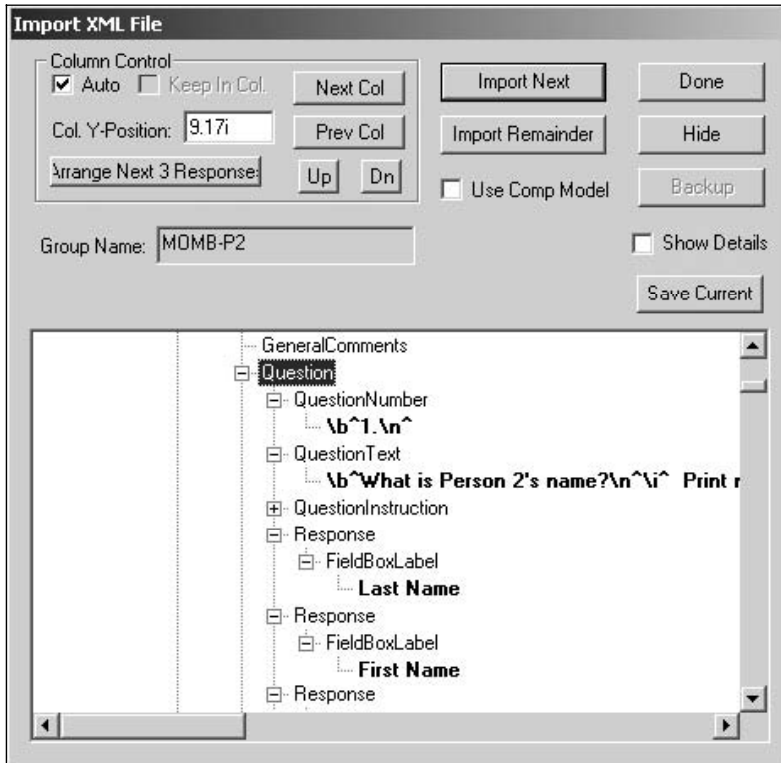


Figure 6-3: XML Import Dialog

During initial XML import for a new form, the OFDP operator uses the existing import and "Arrange Responses" tools, as well as interactive positioning, to create the first instance of the preferred layout.

- Toggle off "Auto" Column Control
- Question 1 is arranged horizontally and moved up.
- Question 2 is arranged into columns from top-to-bottom.
- Questions 3 and 4 are arranged horizontally and spaced appropriately.
- Questions 5, 6, and 7 are interactively moved into position and have point size adjustments.

All of these graphical layout properties are stored within the file as question number composition models, and are relative in position to the upper left corner of their originating column.

Using Composition Modeling is very easy. Upon clicking the [Next Column] button, the upper left corner of the new column becomes the reference point for composition model input. Then whenever there is a repeating question, just checking "Use Composition Model" before clicking the [Import Next] button will activate import of raw content using the stored composition model for that repeating question. Clicking [Import Next] five more times will import the new material for all 7 questions in column 2.

Extend this action for each column just by selecting the [Next Column], [Import Next], for the remaining 3 columns.

As you can see, this greatly accelerates the creation of new form designs with repeating questions.

Quick Step-by-Step

1. Toggle Off [Auto] column control.
2. Import all text for Column 1 using [Import Next], [Arrange Responses], and interactively moving, positioning, and sizing question/responses.
3. Click [Next Column] to display the position of Column 2.
4. Check "Use Composition Model" to activate the stored graphical layouts.
6. Click [Import Next] for each question in Column 2.
7. Click [Next Column] to display the position of Column 3.
8. Click [Import Next] for each question in Column 3.
9. Continue with Steps 7 and 8 for all remaining Columns.

Miscellaneous Notes

The position of Composition Models are relative to the upper-left corner of the originating column. So, the [Next Column] button is necessary to reposition the import. Turn off "Auto" Column Control so that you can control the column flow.

Each unique question number imported automatically invokes the creation of a new composition model. The models are referenced by their question text numbers. For example: Q1, Q2, Qa, Qb, Qi, Qii, etc.

Remember, if there is a repeating question save time by using composition model. This will import the new content data using the first graphical layout occurrence of that question number.

If you need to [Import Next] for new non-repeating questions after using composition models, the beginning vertical position of the data will need to be manually adjusted using the [New Column] or [Up] and [Down] buttons, before selecting [Import Next].

The naming of composition models, by default, is by question number. Each unique question number imported automatically invokes the creation of a new composition model. The models are referenced by their question text numbers.

For example: Q1, Q2, etc. If the question number text does not exist, then a "Group Name" Dialog will appear. Manually name your composition model by typing in the group name.

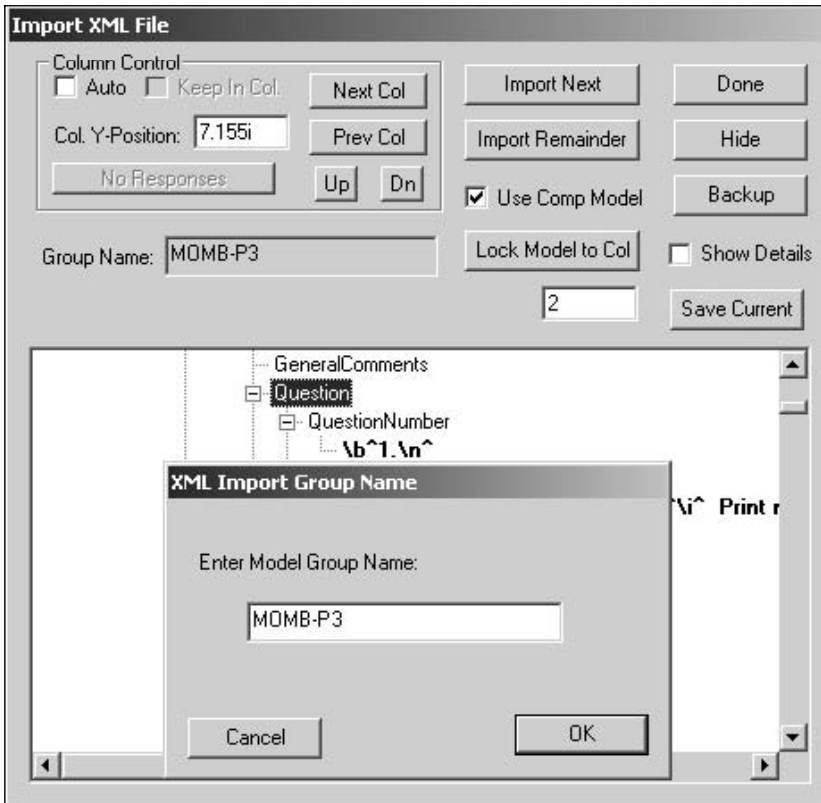


Figure 6-4: Naming Composition Model Groups

Using Lock Model to Column

Composition Model's "Lock Model to Column" option allows you to choose a column for the model style before importing. This gives the modeling option more flexibility. For example, question 1 can have two different looks between column 1 and 2. When it is time to import question 1 for column 3, you can choose between the look of column 1 or 2. If a "0" appears in the Lock Model field, no column has been chosen and the Composition Model feature will pick the first model found.

When you click on the [Lock Model to Col] button, you will be prompted to "pick item from Column for Lock Model to Use". When picking to identify the column which is to be the model, pick on any piece of text within that

column. You should notice that column number appear in the Lock Model field.

Clicking [Import Next] will automatically pull in the question using that column's model. If it asks for a model name, it wants the "Group Name" of the model you are wanting to use.

Quick Step-by-Step

1. Toggle Off [Auto] column control.
2. Import all text for Column 1 using [Import Next], [Arrange Responses], and interactively moving, positioning, and sizing question/responses.
3. Click [Next Column] to display the position of Column 2.
4. Import text for Question 1 using [Import Next], [Arrange Responses], and interactively moving, positioning, and sizing question/responses. It is a different look than in Column 1.
5. Check "Use Composition Model" to activate the stored graphical layouts.
6. Click [Import Next] for the remaining questions in Column 2.
7. Click [Next Column] to display the position of Column 3.
8. Click [Lock Model to Col] and pick on any piece of text within Column 2. You will see a "2" appear in the Lock Model to Column field.
9. Click [Import Next] for question 1 and you will see it look like Column 2. [Import Next] for the remaining questions.

CHAPTER 7

Synchronizing Imported Objects

"Synchronize" is the term used to describe the process of comparing the previous content.xml with the current (edited) content.xml, and making changes to the components of the form that are different. By default during synchronization, OFDP compares the two most recent content files and identifies content differences, and applies those changes to the ".g" file. By default, the graphic attributes of the synchronized object are always based on the ".g" file.

This function is provided so that minor changes can easily be made to composed forms through the Import XML interface while preserving the integrity of the XML data maintained within the repository.

Changes to question text, response captions, Edits to questions of responses, insertion of new responses, deleted questions or responses, will take place automatically.

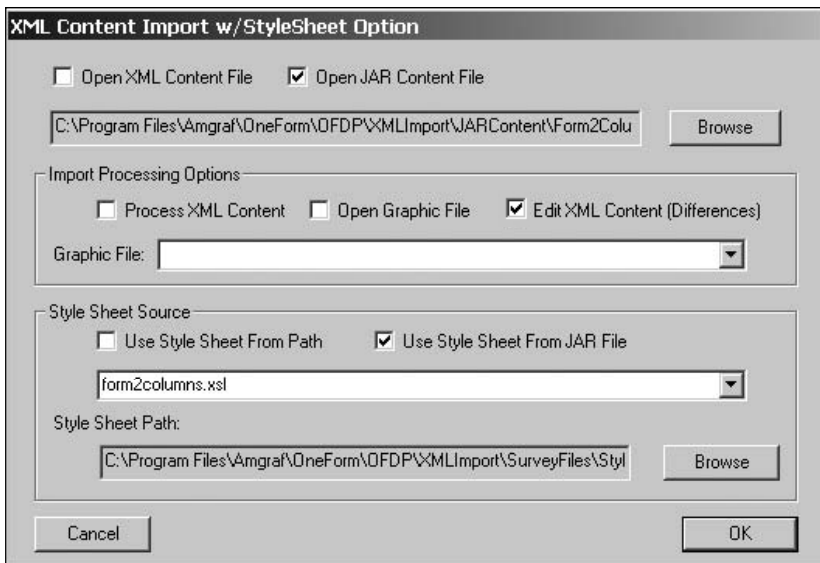


Figure 7-1: Opening XML Content for Edit Differences

The "Edits" option is only available when opening a JAR file that contains (at a minimum) three files. These files are required in order to synchronize updated content with the current graphic form design:

Graphic ".g" File Format: Form_Number.g

Previous Content file: Form_Number_PREVIOUS_content.xml

Current Content file: Form_Number_CURRENT_content.xml

Note: The sequences of GUID items in the XML "content" files are important and should be consistent from one version to the next. The form to be modified must be available as a graphic ".g" file in OFDP. The only Delete or Replace edits allowed are those that refer to GUID's within the targeted form. If GUID's, or a file, is missing, a warning message is displayed and the operation is terminated.

The two XML files (PREVIOUS and CURRENT) will be compared to determine the changes from version to version.

To help the operator see the object/component that has been changed, the background of the location where a change takes place (deleted, inserted, or replaced) will be highlighted in gray.

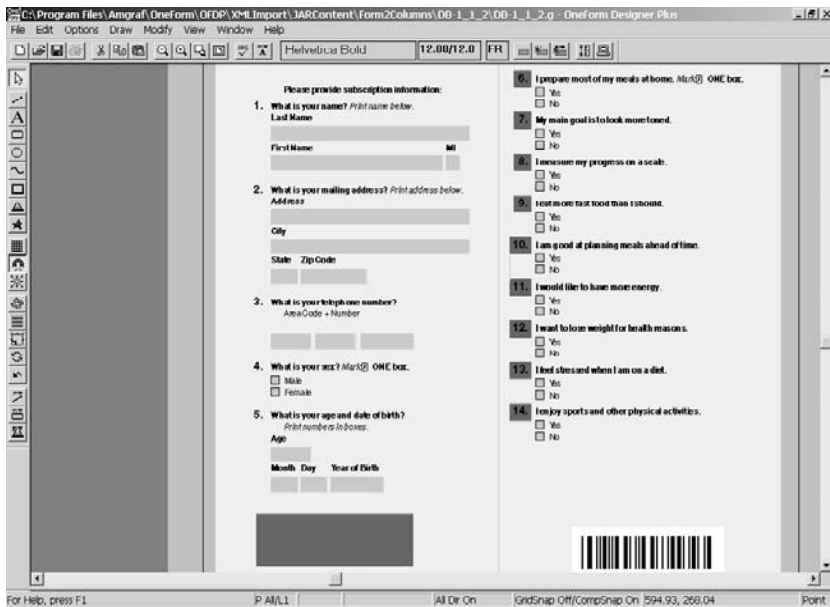


Figure 7-2: Gray Boxes Identify Changes

The gray boxes used to highlight changes are on the non-printing Layer 22.

Four kinds of variances will be automatically detected:

- GUIDDelete - Indicates that the specified GUID is to be Deleted and identified with a highlighted gray box.
- GUIDInsertFirst - Indicates that the specified GUID is to be Inserted as the First object/component on the form. If this is a New Question, the Insert Dialog will appear.
- GUIDInsertAfter - Indicates that the specified GUID is to be Inserted After the reference GUID (used for all other inserted objects/components). If this is a New Question, the Insert Dialog will appear.

- GUIDReplace - Indicates that the specified GUID is to be Replaced and identified with a highlighted gray box.

The changed GUID's will be detected and written to an "XML Edits" file. The program will then examine the MECCA ".g" file to insure that all the GUID's referenced for delete or replace are present on the targeted form. The synthesized "XML Edits" file will then be processed and displayed in the "Text Differences Log".

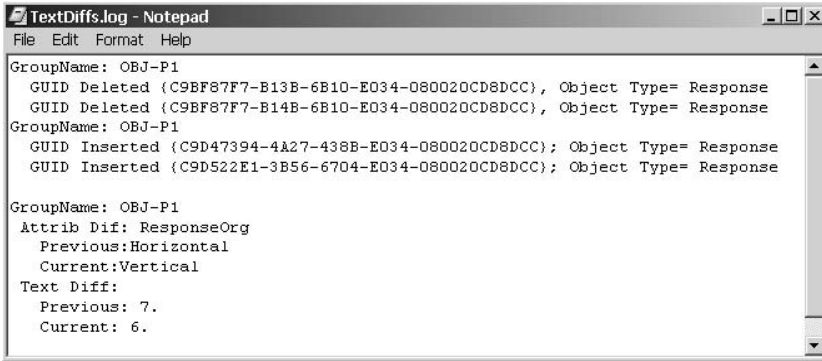
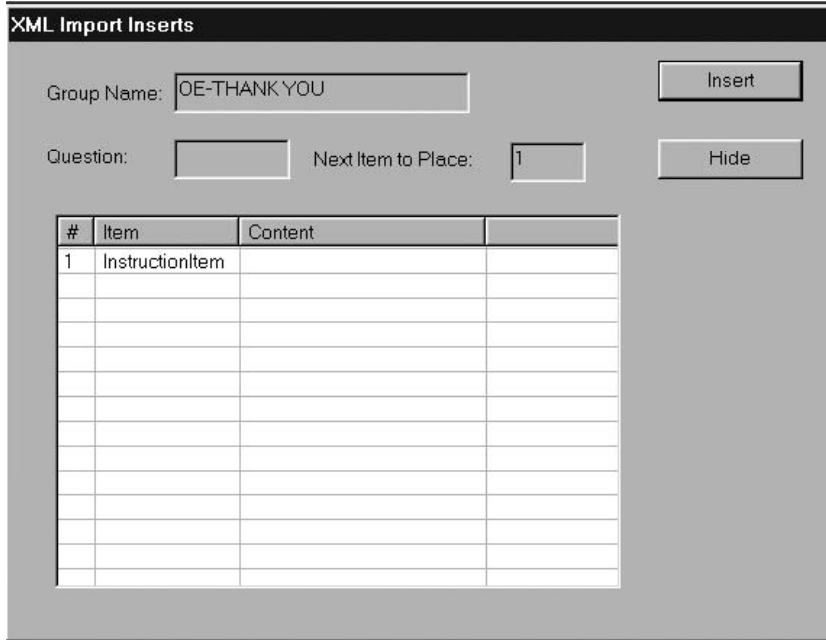


Figure 7-3: Text Differences Log

When the XML import for edit and composition processes are completed, the form can be exported as XML, and both the content and layout information will be preserved. See *XML Save (Export) Option*.

Inserting New Questions/Instructions

Upon opening the XML Edit file that contains New GUID Questions, the display changes to show the interactive window next to the form being composed. The New Question data will appear. The only button choices allowed are [Insert] and [Hide].



#	Item	Content
1	InstructionItem	

Figure 7-4: Insert New Dialog

Clicking [Insert] prompts you to "Pick new location for XML Insert". Pick **where** the new Question Instructions should be placed on the form.

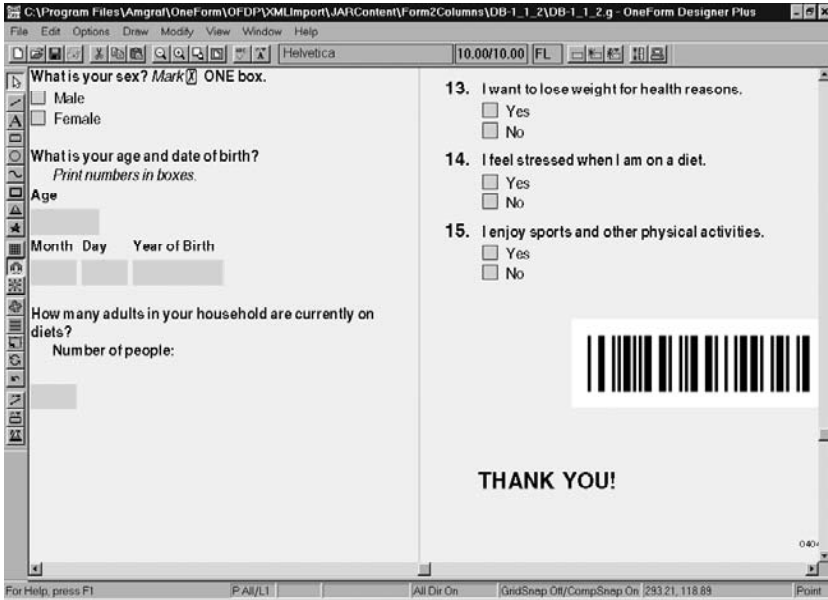


Figure 7-5: The New Data will be Composed

Forcing Synchronization of XML Content

By default during synchronization, OFDP compares the two most recent content files and identifies content differences, and applies those changes to the ".g" file. By default, the graphic attributes of the synchronized object are always based on the ".g" file.

Using the "Force Synchronization" tag within XML content, OFDP allows an IBEAM content writer to force the recomposition of certain GUID components on a survey form, and optionally indicate that a stylesheet is to be used for graphical attributes during recomposition.

Syntax: <SyncFlag="value">

Values:

C – Content Only, Force Synchronization using CURRENT Content and ".g" Graphical Attributes

B – Both Content and StyleSheet, Force Synchronization using CURRENT Content and Stylesheet Attributes

N – No Synchronization

The technique is to use the attribute tag called SyncFlag for an individual GUID. It will be examined first before the "content difference" processor is invoked.

The SyncFlag tag has been added to the DTD for use within the following elements: Instruction Item Text, Instruction Item, Question Text, Question Instruction, Question Number, and Response data types.

When the SyncFlag value for an individual GUID component is "C", OFDP will recompose the item using the graphical attributes found in the ".g" file. This value forces composition for those marked GUID components regardless of content difference. This can be used to recompose special character symbols as they change in design.

When the SyncFlag value for an individual GUID component is "B", OFDP will recompose the item using the graphical attributes found in the Stylesheet. This value forces composition for those GUID components regardless of content difference and can be used to recompose the GUIDs while using the attributes in another stylesheet. Just browse for the new stylesheet before processing XML content differences.

If you have GUID components that should never be changed, apply the SyncFlag with the "N" value for "No Synchronization".

Example of Each SyncFlag Tag Value in Use

C – Content Only, Force Synchronization using CURRENT Content and ".g" Graphical Attributes

Following is an example of the SyncFlag tag with the value "C" for the QuestionText and Response elements. This forces recomposition of the QuestionText regardless of text differences. In this example the PREVIOUS and CURRENT content QuestionText was the same, but the forced recompose action caused it to see a new font used for the XMark character. The Response labels are different and get recomposed using CURRENT content and ".g" graphical attributes.

```
<Question GUID="{BD29EA8A-4BD3-34B2-E034-080020CD8C15}" AllowResponse="Multiple">
<QuestionNumber> <B>4.</B> </QuestionNumber>
<QuestionText SyncFlag="C"><B>What is your sex?</B> <I> Mark &XMark; ONE
box.</I></QuestionText>
<Response GUID="{BD29EA8A-4C00-34B2-E034-080020CD8C15}" ResponseOrg="Vertical"
SyncFlag="C">
<CodeNumber/>
<FieldName>P1_SEX_MALE </FieldName>
<DataType>YesNo </DataType>
<ResponseType Type="CheckBox" CharCount="1" BoxHsize=".125i" BoxVsize=".125i"/>
<FieldBoxLabel>MALE </FieldBoxLabel>
</Response>
<Response GUID="{BD29EA8A-4C01-34B2-E034-080020CD8C15}" ResponseOrg="Horizontal"
SyncFlag="C">
<CodeNumber/>
<FieldName>P1_SEX_FEMALE </FieldName>
<DataType>YesNo </DataType>
<ResponseType Type="CheckBox" CharCount="1" BoxHsize=".125i" BoxVsize=".125i"/>
<FieldBoxLabel>FEMALE </FieldBoxLabel>
</Response>
</Question>
```

PREVIOUS	CURRENT
<p>4. What is your sex? Mark <input type="checkbox"/> ONE box.</p> <p><input type="checkbox"/> Male</p> <p><input type="checkbox"/> Female</p>	<p>4. What is your sex? Mark <input type="checkbox"/> ONE box.</p> <p><input type="checkbox"/> MALE</p> <p><input type="checkbox"/> FEMALE</p>

Figure 7-6: Example of SyncFlag="C" Output

B – Both Content and StyleSheet, Force Synchronization using CURRENT Content and Stylesheet Attributes

Following is an example of the SyncFlag tag with the value "B" for the QuestionNumber, QuestionText, and the Response elements. This forces recomposition of the QuestionNumber and QuestionText regardless of text differences, while also looking at the attributes of a stylesheet. In this example the PREVIOUS and CURRENT content are the same (except for the SyncFlag) but the stylesheet chosen on the "XML Import" Dialog was TimesRoman.xsl instead of the original Helvetica.xsl. The recompose action will see the font Times Roman as specified in the new stylesheet.

```
<Question GUID="{BD29EA8A-4BD3-34B2-E034-080020CD8C15}" AllowResponse="Multiple">
<QuestionNumber SyncFlag="B">
<B>4.</B> </QuestionNumber>
<QuestionText SyncFlag="B"><B>What is your sex?</B> <I> Mark &XMark; ONE
box.</I></QuestionText>
<Response GUID="{BD29EA8A-4C00-34B2-E034-080020CD8C15}" ResponseOrg="Vertical"
SyncFlag="B">
<CodeNumber/>
<FieldName>P1_SEX_MALE </FieldName>
<DataType>YesNo </DataType>
<ResponseType Type="CheckBox" CharCount="1" BoxHsize=".125i" BoxVsize=".125i"/>
<FieldBoxLabel>MALE </FieldBoxLabel>
</Response>
<Response GUID="{BD29EA8A-4C01-34B2-E034-080020CD8C15}" ResponseOrg="Horizontal"
SyncFlag="B">
<CodeNumber/>
<FieldName>P1_SEX_FEMALE </FieldName>
<DataType>YesNo </DataType>
<ResponseType Type="CheckBox" CharCount="1" BoxHsize=".125i" BoxVsize=".125i"/>
<FieldBoxLabel>FEMALE </FieldBoxLabel>
</Response>
</Question>
```

PREVIOUS	CURRENT
<p>4. What is your sex? Mark <input type="checkbox"/> ONE box.</p> <p><input type="checkbox"/> Male</p> <p><input type="checkbox"/> Female</p>	<p>4. What is your sex? Mark <input type="checkbox"/> ONE box.</p> <p><input type="checkbox"/> MALE</p> <p><input type="checkbox"/> FEMALE</p>

Figure 7-7: Example of SyncFlag="B" Output

N – No Synchronization

Following is an example of the SyncFlag tag with the value "N". This skips the item regardless of text differences. In this example the PREVIOUS and CURRENT content are not the same. The QuestionText and Responses have been capitalized, but you will notice the text comparison was skipped and no recomposition was performed because of the SyncFlags.

```
<Question GUID="{BD29EA8A-4BD3-34B2-E034-080020CD8C15}" AllowResponse="Multiple">
<QuestionNumber> <B>4.</B> </QuestionNumber>
<QuestionText SyncFlag="N"><B>WHAT IS YOUR SEX?</B> <I> Mark &XMark; ONE
box.</I></QuestionText>
<Response GUID="{BD29EA8A-4C00-34B2-E034-080020CD8C15}" ResponseOrg="Vertical"
SyncFlag="N">
<CodeNumber/>
<FieldName>P1_SEX_MALE </FieldName>
<DataType>YesNo </DataType>
<ResponseType Type="CheckBox" CharCount="1" BoxHsize=".125i" BoxVsize=".125i"/>
<FieldBoxLabel>MALE </FieldBoxLabel>
</Response>
<Response GUID="{BD29EA8A-4C01-34B2-E034-080020CD8C15}" ResponseOrg="Horizontal"
SyncFlag="N">
<CodeNumber/>
<FieldName>P1_SEX_FEMALE </FieldName>
<DataType>YesNo </DataType>
<ResponseType Type="CheckBox" CharCount="1" BoxHsize=".125i" BoxVsize=".125i"/>
<FieldBoxLabel>FEMALE </FieldBoxLabel>
</Response>
</Question>
```

PREVIOUS	CURRENT
<p>4. What is your sex? Mark <input type="checkbox"/> ONE box.</p> <p><input type="checkbox"/> Male</p> <p><input type="checkbox"/> Female</p>	<p>4. What is your sex? Mark <input type="checkbox"/> ONE box.</p> <p><input type="checkbox"/> Male</p> <p><input type="checkbox"/> Female</p>

Figure 7-8: Example of SyncFlag="N" Output

Example of Using SyncFlag Tags in CURRENT XML

The CURRENT content XML file was modified to show usage of SyncFlag tags. They are shown in **BOLD** print, along with any text changes for comparison. See example of output on page 8.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE OFDPBOCSurvey SYSTEM "OFDPBOCSurvey.dtd">
<!-- Generated by Oracle Reports version 9.0.2.0.1 -->
<OFDPBOCSurvey>
<QRM GUID="{BD29EA8A-4BC3-34B2-E034-080020CD8C15}">
<Year>2008</Year>
<FormTitle>Updated Survey</FormTitle>
<FormNumber>S-2</FormNumber>
<OfficeAddress>Diet and Exercise Magazine Survey, P.O. Box 9863 Newport, MO 68474
</OfficeAddress>
<OMBNumber>THANK YOU!</OMBNumber>
<OMBExpirationDate></OMBExpirationDate>
<FormFace GUID="{BD29EA8A-4BC4-34B2-E034-080020CD8C15}" FaceNumber="1"
Barcode="040401">
</FormFace>
<FormInstruction>Please help us better serve our customers by filling out this quick and
easy survey. Your time is greatly appreciated and will be rewarded with a FREE one year's
subscription. </FormInstruction>
<QuestionGroup GUID="{BD29EA8A-4BCC-34B2-E034-080020CD8C15}" GroupName="MOMB-P1">
<Question GUID="{BD29EA8A-4BD1-34B2-E034-080020CD8C15}" AllowResponse="Multiple">
<QuestionNumber></QuestionNumber>
<QuestionText></QuestionText>
<InstructionItem GUID="{E91692D0-6086-4737-E034-080020CD8C15}"
InstructionOrg="Vertical"
SyncFlag="N">
<InstructionItemText><B>Provide subscription information be-
low:</B></InstructionItemText>
</InstructionItem>
</Question>
<Question GUID="{BD29EA8A-4BD2-34B2-E034-080020CD8C15}" AllowResponse="Multiple">
<QuestionNumber SyncFlag="C"> <B>1.</B> </QuestionNumber>
<QuestionText SyncFlag="C"><B>What is your full name?</B> <I>Print name
below.</I></QuestionText>
<Response GUID="{BD29EA8A-4C35-34B2-E034-080020CD8C15}" ResponseOrg="Vertical"
SyncFlag="C">
<CodeNumber/>
<FieldName>P1_NAME_LAST </FieldName>
<DataType>Any </DataType>
<ResponseType Type="CombSpacedBox" CharCount="15" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> <B>Last Name</B> </FieldBoxLabel>
</Response>
<Response GUID="{BD29EA8A-4C34-34B2-E034-080020CD8C15}" ResponseOrg="Vertical"
SyncFlag="C">
<CodeNumber/>
<FieldName>P1_NAME_FIRST </FieldName>
<DataType>Any </DataType>
<ResponseType Type="CombSpacedBox" CharCount="13" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> <B>First Name</B> </FieldBoxLabel>
</Response>
```

```
<Response GUID="{BD29EA8A-4C36-34B2-E034-080020CD8C15}" ResponseOrg="Horizontal"
SyncFlag="C">
<CodeNumber/>
<FieldName>P1_NAME_MI </FieldName>
<DataType>Any </DataType>
<ResponseType Type="CombSpacedBox" CharCount="1" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> <B>MI</B> </FieldBoxLabel>
</Response>
</Question>
<Question GUID="{BD29EA8A-4BD5-34B2-E034-080020CD8C15}" AllowResponse="Multiple">
<QuestionNumber> <B>2.</B> </QuestionNumber>
<QuestionText><B>What is your mailing address?</B> <I>Print address
below.</I></QuestionText>
<Response GUID="{BD29EA8A-4D35-34B2-E034-080020CD8C15}" ResponseOrg="Vertical">
<CodeNumber/>
<FieldName>P1_ADDRESS </FieldName>
<DataType>Any </DataType>
<ResponseType Type="CombSpacedBox" CharCount="15" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> <B>Address</B> </FieldBoxLabel>
</Response>
<Response GUID="{BD29EA8A-4D34-34B2-E034-080020CD8C15}" ResponseOrg="Vertical">
<CodeNumber/>
<FieldName>P1_CITY </FieldName>
<DataType>Any </DataType>
<ResponseType Type="CombSpacedBox" CharCount="15" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> <B>City</B> </FieldBoxLabel>
</Response>
<Response GUID="{BD29EA8A-4D36-34B2-E034-080020CD8C15}" ResponseOrg="Horizontal">
<CodeNumber/>
<FieldName>P1_STATE </FieldName>
<DataType>Any </DataType>
<ResponseType Type="CombSpacedBox" CharCount="2" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> <B>State</B> </FieldBoxLabel>
</Response>
<Response GUID="{BD29EA8A-4D37-34B2-E034-080020CD8C15}" ResponseOrg="Horizontal">
<CodeNumber/>
<FieldName>P1_ZIPCODE </FieldName>
<DataType>Any </DataType>
<ResponseType Type="CombSpacedBox" CharCount="5" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> <B>Zip Code</B> </FieldBoxLabel>
</Response>
</Question>
<Question GUID="{BD7D5C43-496D-3D7B-E034-080020CD8C15}" AllowResponse="Multiple">
<QuestionNumber> <B>3.</B> </QuestionNumber>
<QuestionText SyncFlag="C"> <B>What is your area code and telephone number?</B>
</QuestionText>
<InstructionItem GUID="{BD7D5C43-496E-3D7B-E034-080020CD8C15}"
InstructionOrg="Vertical">
<InstructionItemText>Area Code + Number </InstructionItemText>
</InstructionItem>
<Response GUID="{BD7D5C43-4970-3D7B-E034-080020CD8C15}" ResponseOrg="Vertical">
<CodeNumber/>
<FieldName>HH_PHONE_AREA </FieldName>
<DataType>NumericInteger </DataType>
<ResponseType Type="CombSpacedBox" CharCount="3" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> </FieldBoxLabel>
</Response>
```

Forcing Synchronization of XML Content

```
<Response GUID="{BD7D5C43-4972-3D7B-E034-080020CD8C15}" ResponseOrg="Horizontal">
<CodeNumber/>
<FieldName>HH_PHONE_PREFIX </FieldName>
<DataType>NumericInteger </DataType>
<ResponseType Type="CombSpacedBox" CharCount="3" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> </FieldBoxLabel>
</Response>
<Response GUID="{BD7D5C43-4974-3D7B-E034-080020CD8C15}" ResponseOrg="Horizontal">
<CodeNumber/>
<FieldName>HH_PHONE_SUFFIX </FieldName>
<DataType>NumericInteger </DataType>
<ResponseType Type="CombSpacedBox" CharCount="4" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> </FieldBoxLabel>
</Response>
</Question>
<Question GUID="{BD29EA8A-4BD3-34B2-E034-080020CD8C15}" AllowResponse="Multiple">
<QuestionNumber> <B>4.</B> </QuestionNumber>
<QuestionText SyncFlag="C"><B>What is your sex?</B> <I> Mark &XMark; ONE
box.</I></QuestionText>
<Response GUID="{BD29EA8A-4C00-34B2-E034-080020CD8C15}" ResponseOrg="Vertical">
<CodeNumber/>
<FieldName>P1_SEX_MALE </FieldName>
<DataType>YesNo </DataType>
<ResponseType Type="CheckBox" CharCount="1" BoxHsize=".125i" BoxVsize=".125i"/>
<FieldBoxLabel>Male </FieldBoxLabel>
</Response>
<Response GUID="{BD29EA8A-4C01-34B2-E034-080020CD8C15}" ResponseOrg="Horizontal">
<CodeNumber/>
<FieldName>P1_SEX_FEMALE </FieldName>
<DataType>YesNo </DataType>
<ResponseType Type="CheckBox" CharCount="1" BoxHsize=".125i" BoxVsize=".125i"/>
<FieldBoxLabel>Female </FieldBoxLabel>
</Response>
</Question>
<Question GUID="{BD29EA8A-4BD4-34B2-E034-080020CD8C15}" AllowResponse="Multiple">
<QuestionNumber SyncFlag="B"> <B>5.</B> </QuestionNumber>
<QuestionText SyncFlag="B"> <B>What is your age and date of birth?</B> </QuestionText>
<InstructionItem GUID="{C06324A8-5B53-64EC-E034-080020CD8C15}"
InstructionOrg="Horizontal">
<InstructionItemText> <I>Print numbers in boxes.</I> </InstructionItemText>
</InstructionItem>
<Response GUID="{C06324A8-5B51-64EC-E034-080020CD8C15}" ResponseOrg="Vertical">
<CodeNumber/>
<FieldName>P1_AGE </FieldName>
<DataType>NumericInteger </DataType>
<ResponseType Type="CombSpacedBox" CharCount="3" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> <B>Age</B> </FieldBoxLabel>
</Response>
<Response GUID="{C06324A8-5B55-64EC-E034-080020CD8C15}" ResponseOrg="Vertical">
<CodeNumber/>
<FieldName>P1_DOB_MONTH </FieldName>
<DataType>NumericInteger </DataType>
<ResponseType Type="CombSpacedBox" CharCount="2" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> <B>Month</B> </FieldBoxLabel>
</Response>
```



```
<Response GUID="{C06324A8-5B57-64EC-E034-080020CD8C15}" ResponseOrg="Vertical">
<CodeNumber/>
<FieldName>P1_DOB_DAY </FieldName>
<DataType>NumericInteger </DataType>
<ResponseType Type="CombSpacedBox" CharCount="2" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> <B>Day</B> </FieldBoxLabel>
</Response>
<Response GUID="{C06324A8-5B59-64EC-E034-080020CD8C15}" ResponseOrg="Vertical">
<CodeNumber/>
<FieldName>P1_DOB_YEAR </FieldName>
<DataType>NumericInteger </DataType>
<ResponseType Type="CombSpacedBox" CharCount="4" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> <B>Year of Birth</B> </FieldBoxLabel>
</Response>
</Question>
<Question GUID="{BD298A8A-4BD2-54B2-E034-080020CD8C15}" AllowResponse="Multiple">
<QuestionNumber SyncFlag="B"> <B>6.</B> </QuestionNumber>
<QuestionText SyncFlag="B"><B>How many people in your household are on a diet?</B>
</QuestionText>
<Response GUID="{BD89EA8A-8C35-3482-E034-080020CD8C15}" ResponseOrg="Vertical"
SyncFlag="B">
<CodeNumber/>
<FieldName>P1_HOWMANY </FieldName>
<DataType>Any </DataType>
<ResponseType Type="CombSpacedBox" CharCount="2" BoxHsize=".2i" BoxVsize=".22i"/>
<FieldBoxLabel> <B>Number of people</B> </FieldBoxLabel>
</Response>
</Question>
etc...
```

Example of Synchronization Output

PREVIOUS	CURRENT
<p style="text-align: center;"><i>Please provide subscription information:</i></p> <p>1. What is your name? <i>Print name below.</i> Last Name <input type="text"/> First Name MI <input type="text"/> <input type="text"/></p> <p>2. What is your mailing address? <i>Print address below.</i> Address <input type="text"/> City <input type="text"/> State Zip Code <input type="text"/> <input type="text"/></p> <p>3. What is your telephone number? Area Code + Number <input type="text"/> <input type="text"/> <input type="text"/></p> <p>4. What is your sex? <i>Mark <input type="checkbox"/> ONE box.</i> <input type="checkbox"/> Male <input type="checkbox"/> Female</p> <p>5. What is your age and date of birth? <i>Print numbers in boxes.</i> Age Month Day <input type="text"/> <input type="text"/> <input type="text"/> Year of Birth <input type="text"/></p> <p>6. How many people in your household are on a diet? Number of people <input type="text"/></p>	<p style="text-align: center;"><i>Please provide subscription information:</i></p> <p>1. What is your full name? <i>Print name below.</i> Last Name <input type="text"/> First Name MI <input type="text"/> <input type="text"/></p> <p>2. What is your mailing address? <i>Print address below.</i> Address <input type="text"/> City <input type="text"/> State Zip Code <input type="text"/> <input type="text"/></p> <p>3. What is your area code and telephone number? Area Code + Number <input type="text"/> <input type="text"/> <input type="text"/></p> <p>4. What is your sex? <i>Mark <input type="checkbox"/> ONE box.</i> <input type="checkbox"/> Male <input type="checkbox"/> Female</p> <p>5. What is your age and date of birth? <i>Print numbers in boxes.</i> Age Month Day <input type="text"/> <input type="text"/> <input type="text"/> Year of Birth <input type="text"/></p> <p>6. How many people in your household are on a diet? Number of people <input type="text"/></p>

Figure 7-9: Previous and Current Output

INDEX

- Absolute 0/0, Move Component 27**
- [Add Column to Base Page] 44**
- Adding New Column 35, 48**
- Applicable DTD Tags 6, 7**
- Arrange Responses, Using 19**
- Automated Composition Forms 5**
- Barcode Tab 41**
- Base Drawing Template, Create 25**
- Block**
 - Information 36
 - Modify 34
 - Tab 32
 - Working with 33
- Boundary, Page Coordinate 0/0 27**
- Box Field, Change Comb 51**
- Change**
 - Comb Box Field 51
 - New Stylesheet, Verify 53
 - P11 Text Style 49
 - Page Boundary 0/0 Coordinate 27
- Coding, XML 6**
 - Question Group Five 11
 - Question Group Four 11
 - Question Group One 8
 - Question Group Six 13
 - Question Group Three 10
 - Question Group Two 9
- Column**
 - Add New 35, 48
 - Modify 34, 47
 - Tab 32
 - Using Lock Model 71
 - Working with 33
- Comb Box Field, Change 51**
- Component**
 - Identify GUID's 3
 - Move to Absolute 0/0 27
 - Stylesheet 29
 - Viewing GUID 63
- Compose Options, XML Import 17**
- Composition Model**
 - Option 67
 - Samples 68
 - Using 69
- Content Open/Processing Options 16**
- Coordinate**
 - Change Page Boundary 0/0 27
 - Getting Stylesheets 27
- Create New Template 25, 26**
- DTD (Document Type Definition) 3, 4**
 - Applicable Tags 6, 7
 - Types
 - OFDPBarcode.dtd 3
 - OFDPBOCSurvey.dtd 3
 - OFDPBOCSurveyEdits.dts 3
 - OFDPBOCSurveyPages.dtd 3
 - OFDPDocument.dtd 3
 - OFDPForm.dtd 3
- Description/Title Elements, Form 6**
- Document Type Definitions 3, 4**
- Drawing Template, Create Base 25**
- During Import, [Save Current] 22**
- Editor, Interactive Stylesheet 29**
- Elements**
 - Form Title/Description 6
 - Question Group 7
- Example XSL File 55**
- (Export), XML Save Option 22**
- Field**
 - Change Comb Box 51
 - No Fillable View 65
 - Response Tab 39
- File**
 - Example XSL 55
 - Opening Stylesheet 30
 - Saving Graphic 22
 - Use JAR 3
 - Working with Graphic 63
- Fillable Fields, No View 65**
- form2columns.xml 55**
- Form Title/Description Elements 6**
- Forms, Automated Composition 5**
- GUID (Globally Unique Identifier) 4**
 - Identify Each Object/Component 3
 - Moving Items 66
 - Viewing Components 63
- GUIDDelete 74**
- GUIDInsertAfter 74**
- GUIDInsertFile 74**
- GUIDReplace 74**
- Getting Coordinates for Stylesheets 27**
- (Globally Unique Identifier) GUID 4**
- Glossary Terms 4**
- Graphic File**
 - Saving 22

Working with 63

Group Elements, Questions 7

How to Make a New Stylesheet 43

ISE (Interactive Stylesheet Editor) 4

Identify Object/Component, GUID's 3

Import

- Remainder 21
- [Save Current] 22
- Synchronizing Objects 73
- Using Options 18
- Using XML Interactive 15
- XML Compose Options 17

Information about Blocks 36

Inserting New Questions/Instructions 76

Interactive

- Stylesheet Editor 4, 29
- Using XML Import 15

Items, Moving GUID 66

JAR (Java Archive File Format) 4

- Save as 24
- Use Files 3

Lock Model to Column, Using 71

Make New Stylesheet 43

Miscellaneous Notes 70

Model

- Composition Option 67, 69
- Lock To Column 71

Modifying Columns 34, 47

Move

- Component to Absolute 0/0 27
- GUID Items 66

New

- Add Column 35, 48
- Create Template 26
- Inserting Question/Instructions 76
- Make Stylesheet 43
- Save Stylesheet 52
- Verify Change in Stylesheet 53

No Fillable Fields View 65

Notes, Miscellaneous 70

OFDP (OneForm Designer Plus) 4

OFDPBarcode.dtd 3

OFDPBOCSurvey.dtd 3

OFDPBOCSurveyEdits.dts 3

OFDPBOCSurveyPages.dtd 3

OFDPDocument.dtd 3

OFDPForm.dtd 3

Object

- Automated Composition XML 5
- Identify GUID's 3
- Synchronizing Imported 73

Open/Processing Options 16

Opening Stylesheet File 30

Option

- Composition Model 67
- Content Open/Processing 16
- Import Compose 17, 18
- Save (Export) 22

Overview XML Workflow 1

P11, Change Text Style 49

PDF, Save as 23

Page Boundary, Change Coordinate 27

Processing/Open Options 16

QRM (Question/Response Metadata) 4

Question Group

- Elements 7
- Inserting New 76
- XML Coding
 - One 8
 - Two 9
 - Three 10
 - Four 11
 - Five 11
 - Six 13

Remainder, Import 21

Requirements, XML Options 2

Response Fields Tab 39

Responses, Using Arrange 19

Sample A 5

Samples, Composition Models 68

[Save Current] During Import 22

Save

- As JAR 24
- As PDF 23
- (Export) Option, XML 22
- Graphic File 22
- New Stylesheet 52

Steps for Creating New Template 26

Style

- Change P11 Text 49
- Text Tab 37

Stylesheet

- Components 29
- Getting Coordinates 27
- Interactive Editor 29
- Make New 43
- Opening File 30
- Save New 52
- Verify Change 53

Synchronizing Imported Objects 73

System Requirements XML Options 2

-
- Tab**
 - Barcode 41
 - Column/Blocks 32
 - Response Field 39
 - Templates 42
 - Text Styles 37
 - Tags, Applicable DTD 6, 7**
 - Template, Create Base Drawing 25, 26**
 - Templates Tab 42**
 - Terms, Glossary 4**
 - Text Style, Change P11 49**
 - Text Styles Tab 37**
 - Title/Description Elements, Form 6**
 - Type, Document Definitions 3**
 - Use**
 - Arrange Responses 19
 - Automated Composition XML Objects 5
 - Composition Modeling 69
 - Import Options 18
 - JAR Files 3
 - Lock Model to Column 71
 - XML Interactive Import 15
 - Verify Changes in New Stylesheet 53**
 - View, No Fillable Fields 65**
 - Viewing GUID Components 63**
 - Workflow, XML Overview 1**
 - Working With**
 - Columns and Blocks 33
 - Graphics File 63
 - XML (eXtensible Markup Language) 4**
 - Coding 6
 - Question Group One 8
 - Question Group Two 9
 - Question Group Three 10
 - Question Group Four 11
 - Question Group Five 11
 - Question Group Six 13
 - Content Open/Processing Options 16
 - Document Type Definitions 3
 - Import Compose Options 17
 - Objects, Automated Composition 5
 - Options, System Requirements 2
 - Save (Export) Option 22
 - Workflow Overview 1
 - Using Interactive Import 15
 - XSD (eXtensible Schema Document) 4**
 - XSL (eXtensible Stylesheet Language File) 4**
 - Example 55
 - 0/0**
 - Change Page Boundary Coordinate 27
 - Move Component to Absolute Zero 27