



CoverPage Designer User's Guide

Introduction

The Imager Cover Page Designer is a set of tools that you use at design-time and run-time to produce fax cover pages from within your application. By combining three simple inputs:

- a Base Image,
- a Template Descriptor and User Supplied Data,
- an infinite variety of fax cover pages

can be generated. Cover Pages are created by layering User Supplied Data on top of a Base Image following the rules defined in a Template Descriptor. Cover Pages are created as CCITT Group 3 TIFF images, DCX fax images, or PCX images.

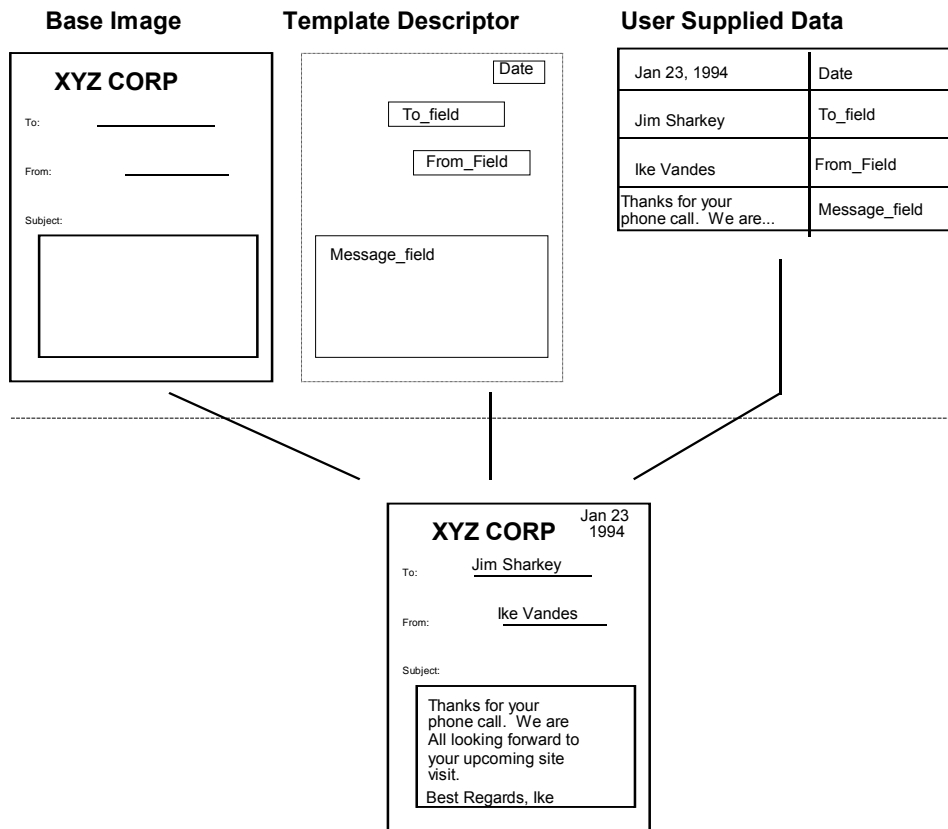
The Base Image (a TIF or PCX file) can be any previously generated fax image, or fax print capture file from a word processor or graphics design program. The base image defines the dimensions of the cover page (short, normal, or long), and the constant data information. Base images can be scanned in from an existing 'paper based' cover page. Alternately, any number of sophisticated word processing or graphics programs can be used to generate a professionally designed cover page by capturing the output through the Print Capture Driver (available at WWW.META-SOFT.COM).

The Template Descriptor defines the placement and formatting of Text, Draw, and Bitmap fields onto the Base Image. Draw objects include Arrows, Circles, Lines, and Squares. Text formatting information includes Size, and Font. Advanced functionality is built in to support word wrap within the field boundaries. Bitmap support provides the capability to add company logos, product logos, and signatures to the cover page. Bitmap field information includes: Scale to Fit, and Crop. The content of Text and Bitmap fields can be replaced at run-time with User Supplied Data. Template Descriptors are *.CVR files and are created either from text input files using the BUILD_CVR.EXE program (included as part of the CoverPage Designer product) or are created interactively using the Imager CoverPage Designer product (licensed separately from the CoverPage Designer as part of the Annotation upgrade to the Fax Viewer product).

User Supplied Data is the variable data used to replace the content of selected template defined fields on the cover page. Supported data types include text, and graphics bitmaps. This data is supplied to the CoverPage Designer by YOUR program at run-time as a variable length list using the programmatic API to the CoverPage Designer DLL (described fully in "Imager CoverPage API.doc").

Specifying a cover page as the product of three inputs ensures that Application Integrators as well as end users can easily customize the Cover Page Designer to provide a wide range of flexible solutions to meet everyone's needs.

Cover Page Formatting can occur on the client machine, or be left to be done at the server. Formatting is done independent of how the fax transmission is created. This ensures cover page information can be modified up till the time of transmission (and can be made available as binary data for BFT transmissions).



CoverPage Designer Components

The CoverPage Designer product consists of the following components.

Documentation

Imager CoverPage User Guide.DOC
Imager CoverPage API.DOC

User's Guide (this document).
CoverPage Designer API.

Runtime DLLs

COVER32.DLL CoverPage Designer
IMG32VI.DLL Support DLL.

Design-time Executables

BUILD_CVR.EXE Template Descriptor Compiler
DUMP_CVR.EXE Template Descriptor Listing Program

Design Time: Creating a Base Image

A Base Image supplied to the CoverPage Designer DLL at run-time can be any previously generated fax image, or fax print capture file (TIF or PCX file). Alternately, if you pass a NULL parameter value in place of the Base Image file name at run-time, the CoverPage Designer DLL will create a blank page with the following default characteristics:

Width : 1728 pels
Height : 2200 pels
HorzDPI : 204
VertDPI : 196

You can easily create a CoverPage Base Image of your own using any word processing, graphics, or publishing program and then capture the output using the ImagerSDK Fax Print Driver (available at WWW.META-SOFT.COM). CoverPage Base Images can include one or more pages.

Design Time: Creating a Template Descriptor

The Template Descriptor defines the placement and formatting of Text, Draw, and Bitmap fields onto the Base Image. Draw objects include Arrows, Circles, Lines, and Squares. Text formatting information includes Size, and Font. Advanced functionality is built in to support word wrap within text field boundaries. Bitmap support provides the capability to add company logos, product logos, and signatures to the cover page. Bitmap field information includes: Scale to Fit, and Crop. The content of Text and Bitmap fields can be replaced at run-time with User Supplied Data.

Template Descriptors are *.CVR files and are created either from text input files using the BUILD_CVR.EXE program (included as part of the CoverPage Designer product) or are created interactively using the Imager CoverPage Designer product (licensed separately from the CoverPage Designer as part of the Annotation upgrade to the Fax Viewer product).

NOTE: Template Descriptor draw objects (arrows, circles, lines, and squares) can only be created using the CoverPage Designer product.

In this section we will describe how to create a CVR file using the BUILD_CVR.EXE application. BUILD_CVR.EXE is a console application that is run from a DOS command line using the following syntax:

```
BUILD_CVR InputFile [OutputFile]
```

Where *InputFile* is the name of the text input file and *OutputFile* is the name of the CVR file to be created. If *OutputFile* is not specified BUILD_CVR will produce an output file called *Template.CVR* in the current directory. To view a compiled CVR file use DUMP_CVR.EXE. DUMP_CVR is a console application that is run from a DOS command line using the following syntax:

```
DUMP_CVR InputFile
```

Output Format:

```
Name:FieldName (FieldNumber:FieldType) Page: PageNumber  
Col:Column Row:Row Width:Width Height:Height  
FieldType (Text or Bitmap)  
Default:DefaultString
```

NOTE: DUMP_CVR.EXE ignores draw objects created by CoverPage Designer.

Input File Format

The *InputFile* to BUILD_CVR is a text file you create using a text editor such as NotePad. This text file contains three types of lines; *CommentLines*, *PageLines* and *FieldLines*. *CommentLines* can appear anywhere in the file but must appear on a line of their own.

The format of a *CommentLine* is:
Text

Where,
Text is anything.

PageLines specify which page of the Base Image that following *FieldLines* will apply to. *PageLines* are optional (if they are omitted all *FieldLines* will apply to the first page of the Base Image). If, at run-time, a template descriptor specifies more pages than exist in the base image the last page in the base image will be repeated as necessary.

The format of a *PageLine* is:
FieldNumber, *PageNumber*

Where,

FieldNumber is always -1 (identifies this line as a *PageLine*).

PageNumber is an integer. *PageNumbers* start at 1 (one) and must appear in the text input file in ascending order.

FieldLines specify type, format, position, and default content of template fields. Each template field is described by one *FieldLine*.

The format of a *FieldLine* is:

FieldNumber, FieldName, FieldType, Column, Row, Width, Height, VariableData

Where,

FieldNumber is an integer that identifies the template field. It is used during run-time by your application to identify to the CoverPage Designer DLL which fields you want to provide user-supplied data for. *FieldNumber* need not be unique however ever template field that shares a *FieldNumber* must be the same *FieldType*.

FieldName is an ASCII string and is for reference only. It may be anything you want.

FieldType is an ASCII string. It may be either "text" or "bitmap". *FieldType* determines the type of the template field and the format and interpretation of the *VariableData*.

Column is an integer pixel position. It specifies the starting (left) column of the area on the Base Image that will contain this template field. Zero is left edge of page.

Row is an integer pixel position. It specifies the starting (top) row of the area on the Base Image that will contain this template field. Zero is top of page.

Width is an integer pixel value. It specifies the width (in pixels) of the area on the Base Image that will contain this template field.

Height is an integer pixel value. It specifies the height (in pixels) of the area on the Base Image that will contain this template field. For *Text* fields a value of 0 (zero) for height specifies that this template field will be formatted as a single line of text. In this case the *Row* value given is the base line of this line of text (lower case letters hang below this base line).

VariableData depends on the value of *FieldType*. For *Text* fields *VariableData* has the following format:

FontName, FontHeight, DefaultString

Where,

FontName is an ASCII string containing the name of the font to use in this template field. Any Windows supported font can be named here. For best results use a True Type font.

FontHeight is an integer specifying the font height in points. Point sizes are relative to the output page size. A standard visible size is around 40 to 50 points. Don't use 12 point as it will appear much too small.

DefaultString is an ASCII string containing the default value for this template field. If, during run-time, there is no User Supplied Data for this field then *DefaultString* will be used.

For *Bitmap* fields *VariableData* has the following format:

Orientation, DefaultBitmapFile

Where,

Orientation is an ASCII string containing either "center", "left", or "right" which determines how the bitmap image will be oriented within the template field.

To display the bitmap with no scaling specify either "right" or "left".

To display the bitmap scaled to fit width specify "center" and specify the *Height* of the template field (see above) as zero (0).

To display the bitmap scaled to fit the template field box (proportions stay the same) specify "center" with *Height* and *Width* (see above) specified.

DefaultBitmapFile is an ASCII string containing the name of a bitmap file to be used as the default value for this template field. If, during run-time, there is no User Supplied Data for this field then the bitmap contained in the file specified by *DefaultBitmapFile* will be used.

Example Text Input File

The following is an example Text Input File.

```
# following says these template fields are on page 1
-1, 1
4, "Company Logo", "bitmap", 10, 10, 1700, 0, "center", "logoxyz.pcx"
6, "To Field", "text", 710, 477, 0, 0, "Arial Bold", 50, "<To>"
8, "To Company Name", "text", 710, 526, 0, 0, "Arial Bold", 38, "<To Company>"
7, "To Fax Number", "text", 710, 618, 0, 0, "Arial Bold", 38, "<To Fax #>"
1, "From Field", "text", 710, 723, 0, 0, "Arial Bold", 50, "<From>"
3, "From Company Name", "text", 710, 773, 0, 0, "Arial Bold", 38, "<From Company>"
2, "From Fax Number", "text", 710, 865, 0, 0, "Arial Bold", 38, "<From Fax #>"
52, "From Tel Number", "text", 710, 818, 0, 0, "Arial Bold", 38, "<From Tel #>"
9, "Number of Pages", "text", 710, 1051, 0, 0, "Arial Bold", 38, "<# Pages>"
10, "Date Field", "text", 710, 956, 0, 0, "Arial Bold", 38, "<Date>"
11, "Time Field", "text", 810, 956, 0, 0, "Arial Bold", 38, "<Time>"
5, "Signature", "bitmap", 337, 1900, 800, 70, "left", "sigfred.pcx"
# following says these template fields are on page 2
# note that the fields "Company Logo", "To Field", and "Signature"
# appear on both page one and page two.
-1, 2
4, "Company Logo", "bitmap", 10, 10, 1700, 0, "center", "logoxyz.pcx"
6, "To Field", "text", 710, 477, 0, 0, "Arial Bold", 50, "<To>"
13, "Text Memo", "text", 337, 1259, 900, 500, "Arial Bold", 38, "<Text Memo>"
5, "Signature", "bitmap", 337, 1900, 800, 70, "left", "sigfred.pcx"
```

Determining the Position of Template Fields

When creating the template text input file you need to specify the coordinates of the bounding rectangle for the template field. One way to do this is to load into Windows Paint (as a PCX file) the CoverPage Base Image that the Template Descriptor will apply to. Under the 'View' menu select 'Cursor Position'. Use the cursor to determine the coordinates of the template fields. Fax images are big (a typical fax image is 1728 pixels across and 2200 pixels down) so don't worry about the size of the base image on the screen.

Run Time: Specifying User Supplied Data

At run-time your application can replace the default value of any template field with data supplied by the user. This data is always a text string. In the case of bitmap fields the string is the name of a bitmap file. In the case of text fields the string is the new text to be displayed. Under Windows NT user supplied data for text template fields can be UNICODE strings. Under Windows NT and Windows 95 user supplied data for text template fields that have a *Height* value that is non-zero (i.e. a text box) can be a Rich Text ANSI string. The interface functions...

CpCreateStringArray () - create an internal string array
CpAddStringElement () - add a string to the allocated array

are used to create and add to the list of user supplied data to replace the default values of template fields. Each element in this list contains a *FieldNumber* which identifies the template field it applies to. See "ImagerSDK CoverPage API.doc" for a complete description of the CoverPage Designer API.

Run Time: Creating the CoverPage

After the list of user supplied data has been provided to the CoverPage Designer by your application you can complete creation of the CoverPage by calling the interface function:

CpCreateCoverPage() - create the coverpage

When calling this function you specify the *BaseImageFile* to be used (or NULL to use a blank page), the *TemplateDescriptorFile* to use (i.e. the name of the CVR file you created using BUILD_CVR.EXE), and the name of the *OutputFile* where the completed coverpage will be stored.

Kit Options

The CoverPage Designer Toolkit comes in two configurations: The OEM Kit, and the End User Kit.

The OEM Kit provides a set of tools for the OEM to create pre-defined cover page templates. The OEM is responsible for building a separate user interface to prompt for user defined data (Addressing information, Memo fields, bitmap fields), and to properly format that data to a Parameter list before making a call to the Cover Page Designer. Unless provided for by the OEM, there are no tools available for the end user to customize Template Descriptors.

The End User Kit provides a set of tools to allow the end user to interactively define the CoverPage Template Descriptor using a visual interface. The CoverPage Designer graphics interface is provided as an enhancement to the ImagerSDK Fax Viewer (part of the Imager product line). Users can view a base image while placing template fields on top of the image. As the image scales or rotates, so do the template fields. This provides a flexible means for the user to see the results at different scaling values, where they can view the whole page, or a small blown up portion thereof.

The Imager CoverPage Designer is offered as part of the Annotation upgrade to the Fax Viewer product.

Copyright © 1997-2004 Metasoft Systems, Inc.